

2024421

by Matthew Perry

FILE	16803_MATTHEW_PERRY_2024421_179196_1063304924.PDF (2.14M)		
TIME SUBMITTED	24-MAY-2021 11:54PM (UTC+0100)	WORD COUNT	7224
SUBMISSION ID	154685589	CHARACTER COUNT	36327

Course: HNC Electrical and Electronic Engineering

Module: PROGRAMMABLE LOGIC CONTROLLERS

Assignment Title: Level and Flow Control System

Name: Matthew Perry

University number: 2024421

Date Submitted: 24/05/21

Word Count: 2722

Table of Contents

Good Structure

1.	Research.....	3
	1.1 Ladder Diagram.....	3
	1.2 Function Block Diagram	3
	1.3 Sequential Function Chart	4
	1.4 Structured Text	4
	1.5 Instruction List	4
2	Hardware Design.....	5
	2.1 Input Output Tag Table	6
	2.2 PLC Drawings.....	6
3	Program Design and Explanation.....	7
4	System Testing	8
5	Meeting the Customer Needs and Team Working	9
6	Conclusions and Recommendations.....	10
7	Citation List.....	11
8.	Appendix.....	12

1. Research

IEC 61131-3 outlines the standard on programming languages and are now used by most of the major PLC manufacturers. The 5 programming languages used are:

- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Sequential Function Chart (SFC)
- Structured Text (ST) also known as Structured Controlled Language (SCL) in Siemens TIA Portal.
- Instruction List (IL) also known Structured Text List (STL) in Siemens TIA Portal.

(Hanssen, 2015) Correctly cited

1.1 Ladder Diagram

According to Bolton (2006), Ladder diagrams are a commonly used method. Writing a program is then the equivalent to drawing a switching circuit. The diagram consists of two vertical lines representing the power rails and the circuits are joined with horizontal lines which are the rungs of the ladder. Along the rungs are contacts and coils. The coils behave as outputs such as a light or a motor. The contacts act as inputs and often represent switches/pushbuttons which can be normally open or normally closed (St.Michael, 2019). Correctly cited Stacking the rungs can allow circuits with particular path ways and N/O and N/C contacts to create logic gates. In relation to **Error! Not a valid bookmark** Improper **self-reference**. in the appendix it can be shown that the input is an normally open contact and the output is a coil. This type of programming latches also includes timers, counters, latching and advanced logic instructions which are a box symbol. An application that this type of language is used, is traffic light systems. This is because they use timers and interlocks.

1.2 Function Block Diagram

Function block diagrams which are usually abbreviated to FBD is a graphical programming language which is represented by blocks. The inputs are entering from the left and the outputs exit from the right. The function blocks have standard **functions such** Correctly cited as counters, timers, logic gates and can also be defined by the designer as required (Bolton, 2006). A block which has been defined by a designer could be a when a particular set of input values then outputs one primary result, these blocks do not include any internal storage refer to Figure 2 for an illustration. One of the benefits while using function block diagrams include data preservation, encapsulation, and safe keeping of data (Walter Correctly cited & Veeramani, 2007). An application that would be preferred for a function block could be a drink

filling machine, this is because there is the use of logic gates. This means it would be easy to configure for inputs to be selected for the desired output.

1.3 Sequential Function Chart

Sequential Function Charts is a graphical type of programming language. The main concept of this programming languages consists of transitions and steps. A step is a partial function within the overall system and a transition is a change from one step to another step (Budimir, 2017). In this language steps are represented as rectangular boxes in a vertical line with the transitions in between (Devasia, 2020). A great advantage of an SFC is that large complex programs can be easily understood in the way that the process is carried out. Other programming languages are converted into SFC for this reason. Figure 3 illustrates an example of an SFC where the numbers represent the steps and the comments in between are the transitions. It should be noted that SFCs are technically not a language but rather a graphical representation of the code and the machine state. (Collins, 2019). Sequential function charts should be used in every application, this would help troubleshooting and help engineers/consumers understand the function of the machine.

1.4 Structured Text

Structured text is a textual programming language, it allows designers to create complex routines and flows that may not be as efficiently carried out in other programs such as ladder logic, an example of this would be creating loops. This programming language is sometimes compared to traditional electronic programming such as C, C++ or java (Romanov, 2020). However, Structured text or SCL known by Siemens is superior to the traditional languages due to extensive applications in data management, process optimization, recipe management and mathematical/statistical operations (Siemens, 1998). The lines are made up of instructions, built-in functions and comments which make it easy for debugging (Anton, 2019). Figure 4; Structured Text/SCL Example in the appendix shows an example of an ST/SCL program that runs a control loop.

1.5 Instruction List

Instruction list, also known as structured text list in Siemens, is a textual based language. This language consists of a list of instructions which allows the designer to create a program using mnemonic commands (ISD, 2017). IL is considered low level logic and was one of the first PLC languages created. Each instruction or command is placed on a new line and is known as 'stacking'. An advantage of this language is that the programs are fast and does not require large amount of memory storage. This programming language is typically used in applications which use control loops (Collins, 2019). Figure 5 illustrates an example of the program.

2 Hardware Design

The hardware components for this design are Digital and Analog and will be described in this section. Some of the hardware components exist within Factory I/O so they cannot be specifically identified.

Pilot lights are used to indicate start, stop and reset; these are digital outputs. Pushbuttons are used for start, stop, and reset the program, which work as a normally open contact, when pushed inputs a signal to the PLC.

A potentiometer is used as an input for the setpoint. This works by varying the position of a sliding contact across resistance, this then gives a variable output voltage (Electrical4u, 2020). This will send a signal to the plc which sets the setpoint.

Analogue actuator valves are used for the filling and discharge process. This works by receiving a signal from the control system. In on/off control the voltage supplied half opens or closes the valves, and when in PID mode the control system will send a voltage between 0-10V to slightly open or slightly close maintaining the incoming/outgoing water (Spirax Sarco, 2021).

An analogue flow meter is used for the flow rate of the discharge, this works by the water pushing the paddle wheel which turns it, thus sending a signal back to the PLC to calculate the flow rate (RS Components, 2020).

An analogue level meter is used to indicate how full the tank is, this could be a conductive or resistance meter. These work by using two metallic probes of different lengths which are inserted into the tank. The long probe transmits a low voltage, the second shorter probe is cut so the tip is at the switching point (SST, 2019).

A Siemens 1214C PLC is used for inputs from input devices such as push buttons and then compares those inputs with program logic data. The PLC is then able to generate outputs to control peripheral output devices such as the valves (Kamel & Kamel, 2013). Analog input and output modules have also been attached to the main PLC so that the analogue hardware can be used.

A 12-inch HMI is used to allow the user to control what mode/controls to select and displays data about the tank such as graphs.

2.1 Input Output Tag Table

A tag table was created to distinguish inputs and outputs including whether they are analogue or digital.

this is a tag list, some of this information is not needed in an IO list

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

Table 1; Input/Output List

2.2 PLC Drawings

PLC drawings of each module was created and displays the connections for the input and outputs for each hardware component. This can be viewed in the appendix *Figure 17; Wiring diagram PLC*.

3 Program Design and Explanation

The program was designed using TIA Portal in Ladder. Ladder was selected because the program itself does not follow a lengthy or in-depth sequence and consists of arithmetic and move function bits. By using Ladder, the networks can be easily understood, and values can be altered quickly to specified tags such as PID control bits. SCL was used to create the auto sequence, this is because it is a short sequence that used tags built within the ladder blocks.

Manual controls are available using buttons and notify the user what state they are in by pilot lights. The fill valve can be opened and each time it is pressed it opens 20%, however each time it is pressed closed it will close by 20%. This method is also used for the manual drain valve controls.

Semi Auto Controls consist of PID or ON-OFF modes. By using the potentiometer on the panel, the user can determine a setpoint, then by choosing PID it will fill to the required setpoint. Output flows can be selected by defining it on the HMI. The PID mode will then calculate the desired actuator output by calculating the proportional, integral, and derivative responses and summing the three components to open the fill valve to the correct value. However, On-Off control will wait for it to reach the dead band, which can be specified, and then half open the valve until the setpoint is reached again. In both modes, the setpoint can be selected either by the potentiometer, or on the HMI.

Auto mode is available, when it needs to be turned auto mode will already be active which is triggered by the HMI screen. Auto mode run will start the sequence which sets the level setpoint to 175 and the flow control to 25. If auto mode stop is pressed, the process will stop and the tank will drain completely. The screen displays the level setpoint and flow setpoint graphs and values. Lastly, each mode is colour co-ordinated for easy identification.

There are 3 HMI screens which are 'Drain Valve Flow Control', 'Fill Valve Level Control' and 'Operator'. The operator screen is the root screen, this is because a user with basic understanding can start the filling and flow with 2 buttons. It allows the customer to not be concerned with any user setting values that may not be used regularly. Two types of graphs are used which are:

- Fill Valve Chart – Tracks the level setpoint and the level process variable.
- Drain Valve Chart – Tracks the flow setpoint and flow process variable.

Each network for all function blocks has appropriate comments and can be located after the PLC drawing in the appendix.

4 System Testing

Description of test	Expected Result	Actual Result	Pass or Fail	Evidence – (Located in the Appendix)
Setpoint Display	Potentiometer on turned to 150, program converts and displays correctly on Factory I/O.	Potentiometer set to 150, then multiplied by 30, then converter from a real to double integer which displayed 150 on the panel meter.	Pass	Figure 6; Test - Setpoint Converter and Display
Process Variable Display	Fill the tank to 150, check PV display on factory I/O and compare with measurement.	Filled to 150, confirmed against measurement.	Pass	Figure 7; Test - Fill to 150, Check Panel PV display
Using manual mode fill to 300.	Once manual mode is selected and open fill valve it should fill to a maximum of 300, then manually close the fill valve.	Manual mode selected, then pressed open fill valve. Filled to 300 then closed the valve.	Pass	Figure 8; Test - Manually fill to 300
Using manual mode drain to 0.	At 300, then manually open the drain valve until 0 is reached.	Once 300 had filled, manually opened the drain valve which emptied the tank completely.	Pass	Figure 9; Test - From 300 manually drain to 0.
Level Control	Set the setpoint to 150 then press level control. Fill valve should open and stop when 150 is reached.	Setpoint is set to 150, pressed level control and semi auto. Filled to 150 then stopped.	Pass	Figure 10; Test - Setpoint to 150
Maintain a level setpoint of 150 with flow out of 20. (PID MODE)	Set the setpoint to 150, press PID control. Once it has filled to 150, then press semi auto and flow control on the drain valve screen. It should drain at 20 and hold the level setpoint with a constant stream in and out of water.	Set the setpoint to 150, pressed semi auto and PID. The tank then filled to 150. Then set the flow control setpoint to 20. Looking at the graph the initial level did drop but recovered and then stabilised to 150.	Pass	Figure 11; Test - Maintain L.SP of 150 with F.SP of 20.
Maintain a level setpoint of 150 with a flow out of 20. (ON-OFF MODE)	Set the setpoint to 150, press semi auto and on off control. The tank should fill to 150, then on the drain HMI select flow control with a setpoint of 20. It should then stay at 150 by turning the fill valve on intermittently.	Setpoint was set to 150, semi auto and on off control was pressed and then the tank filled to 155. Once flow control was selected with a setpoint of 20 the level dropped. The fill valve opened every time the level went under 153 however was late responding and overshoot +- 10%. Instead of it stabilising the level oscillated above and under the setpoint.	Pass	Figure 12; Test - Maintain L.SP of 150 and F.SP of 20 On-Off Control

Maintain a level setpoint of 150 with flow out of 50. (PID MODE)	Set the setpoint to 150, press PID control. Once it has filled to 150, then press semi auto and flow control on the drain valve screen. It should drain at 50 and hold the level setpoint with a constant stream in and out of water. 10 % tolerance of 150 = acceptable.	Set the setpoint to 150, press PID control. Once it has filled to 150, then selected semi auto and flow control on the drain valve screen. It drained at a flow control setpoint of 50. Initial the level dropped considerably but then steadily rose and stabilised at 141.	Pass	Figure 13; Test - Maintain L.SP of 150 and F.SP of 50 PID
Maintain a level setpoint of 150 with a flow out of 50. (ON-OFF MODE)	Set the setpoint to 150, press semi auto and on off control. The tank should fill to 150, then on the drain HMI select flow control with a setpoint of 50. It should then stay at 150 by turning the fill valve on intermittently.	Setpoint was set to 150, semi auto and on off control was pressed and then the tank filled to 154. Once flow control was selected with a setpoint of 50 the level dropped dramatically. It eventually stabilised at 50.	Fail	Figure 14; Test - Maintain L.SP of 150 and F.SP of 50 On-Off Control
Auto Mode	Auto mode should be on once on the operator screen, press auto mode run and the level setpoint should be 175 and flow setpoint should be 25. The tank should fill to 175 while maintaining a flow output of 20. Once auto mode stop is pressed the tank should drain completely.	Auto mode is set while on the page, auto mode run is pressed and the setpoints are set and the tank filled to 175. However the flow out did not maintain 25 until the level stabilised. Once stabilised the level and flow was maintained.	Pass	Figure 15; Test - Auto Start up &

5 Meeting the Customer Needs and Team Working

In the group, it was decided that each program was to be presented to check that all modes were working correctly. During this, faults were diagnosed due to bits being tagged incorrectly such as 'flow process variable' instead of 'level process variable'. It was decided that everyone should include pilot lights to alert the user to what state they have selected but also, to help build evidence while testing each mode. Lastly, while analysing the graphs for On-Off control they showed that it was overshooting by more than 10 percent of the required setpoint. To try combat this when the process variable is lower than the setpoint instead of moving 10 to the fill valve it was changed to 5. This slightly improved the overshoot and brought the specification close to the customer's needs.

Maintenance and reliability of the program and hardware was discussed. Due to the liquid being water, there was no reason for the reliability of the hardware to be questioned unless it was due to age. However, when talking about maintenance, the potentiometer and flow meter was focused on. This is because it will require calibration. The possibility of removing the setpoint dial on the panel meter could be an idea for the future. All hardware equipment should be checked yearly and calibrated if required. With the use of displays, graphs, and practical measurements it will allow any engineer to perform checks accurately. If alterations need to be made, each network has been named appropriately, which also includes comments.

6 Conclusions and Recommendations

In conclusion, the PLC system design was successful. The design offers 3 modes and 2 control methods. It is possible to specify setpoints and meet them using on-off control and PID control. Both setpoints can be compared against the process variable with graphs and displays. The setpoints and dead band can be altered by the customer via the potentiometer on the panel or the HMI screen. Auto mode can be altered so that setpoints are predetermined to fill and flow, including the tank to drain after auto mode is turned off.

During this design, analogue sensors and displays have been used. By using this, knowledge has been gained on how to scale the input onto a display appropriately. Learning how each part of the PID design by arithmetic functions has been easy to understand due to the function boxes being titled appropriately.

Lastly, there are some features that could be added or altered. In regard to on-off control, if the customer needs a higher flow setpoint then instead of 5V being sent to the fill valve, 10V should be sent. This will combat the level from dropping too much, however it will cause a larger overshoot. This means it would be appropriate for the customer to specify how large of a flow output would be regularly used. The potentiometer on the panel could be removed to improve maintenance times and reliability, because if it breaks down it can cause a faulty setpoint. The flow setpoint and process variable could be implemented onto the panel and an emergency stop button separate from the HMI could be added for an additional safety feature.

7 Citation List

A.Rehg, J., 2001. *Structured PLC Programming With Sequential Function Charts*. Pennsylvania, American Society for Engineering Educational Conference & Exposition.

Anton, K., 2019. *PCCWEB*. [Online]

Available at: <https://www.pccweb.com/wp-content/uploads/2019/11/S17-SCL.pdf>
[Accessed 29th April 2021].

Bolton, W., 2006. *Programmable Logic Controllers*, Oxford: Newnes.

Budimir, M., 2017. *What are sequential function charts (SFCs) for PLCs?*. [Online]

Available at: <https://www.motioncontroltips.com/sequential-function-charts-sfcs-plcs/>
[Accessed 22nd April 2021].

Collins, D., 2019. *Why is the instruction list (IL) language for PLCs falling out of favor?*. [Online]

Available at: <https://www.motioncontroltips.com/why-is-the-instruction-list-il-language-for-plcs-falling-out-of-favor/>
[Accessed 10th May 2021].

Devasia, A., 2020. *An Overview of Sequential Function Chart (SFC) Programming*. [Online]

Available at: <https://control.com/technical-articles/an-overview-of-sequential-function-chart-sfc-programming/>
[Accessed 22nd April 2021].

Electrical4u, 2020. *Potentiometer: Definition, Types, And Working Principle*. [Online]

Available at: <https://www.electrical4u.com/potentiometer/>
[Accessed 12th 5 2021].

Hanssen, D. H., 2015. *Programmable Logic Controllers : A Practical Approach to IEC 61131-3 Using CoDeSys : A Practical Approach to IEC 61131-3 Using CoDeSys*. 1st ed. Chichester: John Wiley & Sons, Incorporated.

ISD, 2017. *The basics of Siemens PLC's and programming in Simatic Step7*. [Online]

Available at: https://isd-soft.com/tech_blog/basics-siemens-plcs-programming-simatic-step7/
[Accessed 10th May 2021].

Kamel, K. & Kamel, E., 2013. *Programmable Logic Controllers: Industrial Control*. s.l.:McGraw-Hill.

Romanov, V., 2020. *Introduction to Structured Text Programming in RSLogix and Studio 5000 Allen Bradley*. [Online]

Available at: <https://www.solisplc.com/tutorials/introduction-to-structured-text-programming-in-rslogix-and-studio-5000-allen-bradley>
[Accessed 29th April 2021].

RS Components, 2020. *Flow Sensors*. [Online]

Available at: <https://uk.rs-online.com/web/c/automation-control-gear/sensors/flow-sensors/>
[Accessed 12th 05 2021].

Siemens, 1998. *Structured Control Language (SCL) for S7-300/S7-400 Programming*. [Online]

Available at: https://cache.industry.siemens.com/dl/files/188/1137188/att_27471/v1/SCLV4_e.pdf
[Accessed 29th April 2021].

Spirax Sarco, 2021. *CONTROL VALVE ACTUATORS AND POSITIONERS*. [Online]
Available at: <https://www.spiraxsarco.com/learn-about-steam/control-hardware-electric-pneumatic-actuation/control-valve-actuators-and-positioners>
[Accessed 12th 05 2021].

SST, 2019. *7 Main Types of Level Sensing Methods - How do they differ?*. [Online]
Available at: <https://sstensing.com/7-main-types-of-level-sensors/#:~:text=Conductive%20sensors%20are%20used%20for,is%20at%20the%20switching%20point.>
[Accessed 12th 05 2021].

St.Michael, S., 2019. *Ladder Logic in Programmable Logic Controllers (PLCs)*. [Online]
Available at: [https://control.com/technical-articles/ladder-logic-in-programmable-logic-controllers-plcs/#:~:text=Ladder%20diagram%2C%20better%20known%20as,PLCs%20\(programmable%20logic%20controllers\).&text=Outputs%20don't%20have%20to,the%20code%20as%20another%20input.](https://control.com/technical-articles/ladder-logic-in-programmable-logic-controllers-plcs/#:~:text=Ladder%20diagram%2C%20better%20known%20as,PLCs%20(programmable%20logic%20controllers).&text=Outputs%20don't%20have%20to,the%20code%20as%20another%20input.)
[Accessed 22nd April 2021].

Walter, T. & Veeramani, A., 2007. *Function block diagrams*. [Online]
Available at: <https://www.controleng.com/articles/function-block-diagrams/>
[Accessed 22nd April 2021].

8. Appendix

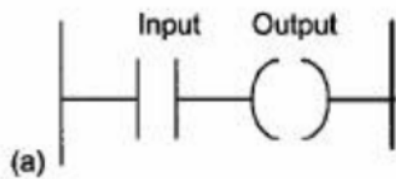


Figure 1; Ladder Diagram Example; (Bolton, 2006)

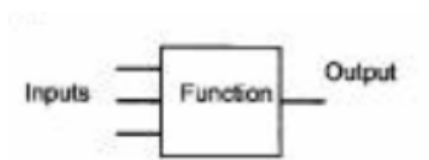


Figure 2; Function Block Example (Bolton, 2006)

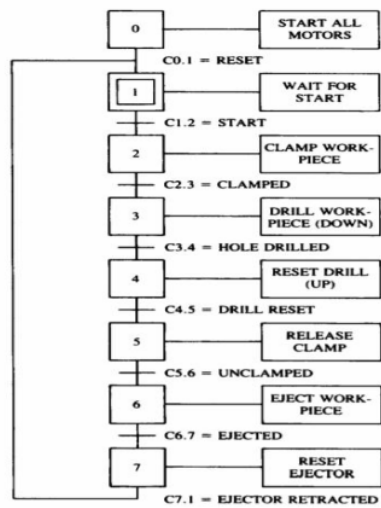


Figure 3; Sequential Function Chart Example (A.Rehg, 2001)

```

1  // Program control with loops
2
3  "MyInt_1" := 0;
4
5  FOR #i := 1 TO 5 DO
6      "MyInt_1" := "MyInt_1" + 1;
7  END_FOR;
8
9  WHILE "MyInt_1" > 0 DO
10     "MyInt_1" := "MyInt_1" - 1;
11 END_WHILE;
12
13 REPEAT
14     "MyInt_1" := "MyInt_1" + 2;
15 UNTIL "MyInt_1" = 10 END_REPEAT;

```

Figure 4; Structured Text/SCL Example (Anton, 2019)

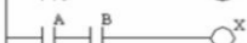
Ladder	Instruction List (IL)	
	LD A	ST X
	LDN A	ST X
	LD A	LD A
	LD B	AND B
	ANB	ST X
	ST X	
	LD A	LD A
	LDN B	ANDN B
	ANB	ST X
	ST X	
	LD A	LD A
	LD B	OR B
	ORB	AND C
	LD C	ST X
	ANB	
	ST X	
	LD A	LD A
	LD B	LD B
	LD C	OR C
	ORB	ANB
	ANB	ST X
	ST X	
	LD A	LD A
	LD B	OR B
	ORB	LD C
	LD C	OR D
	LD D	ANB
	ORB	ST X
	ANB	
	ST X	

Figure 5; Instruction List Example (Collins, 2019)

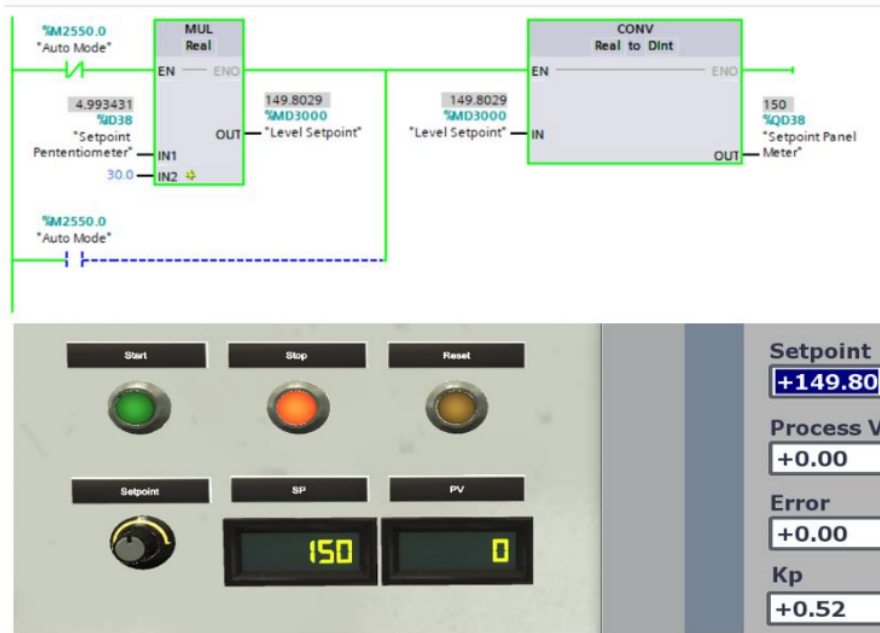


Figure 6; Test - Setpoint Converter and Display



Figure 7; Test - Fill to 150, Check Panel PV display

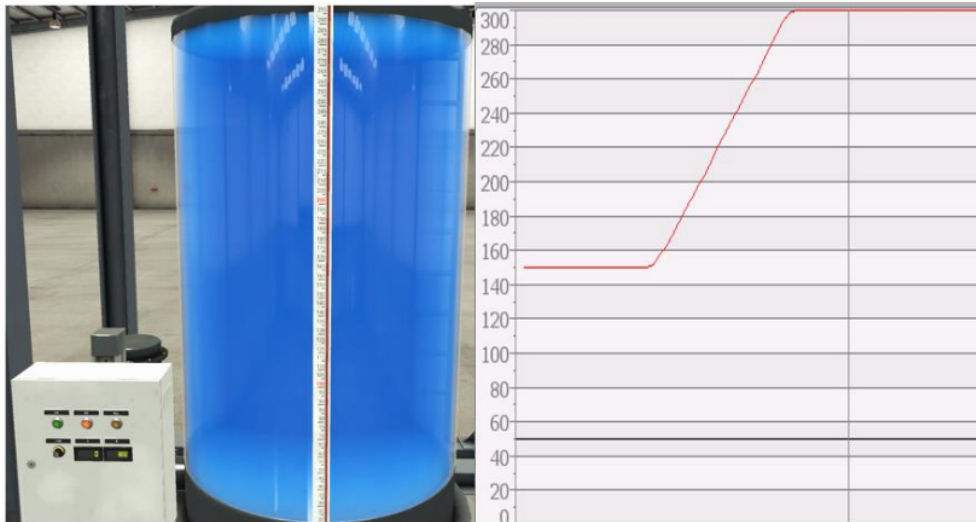


Figure 8; Test - Manually fill to 300

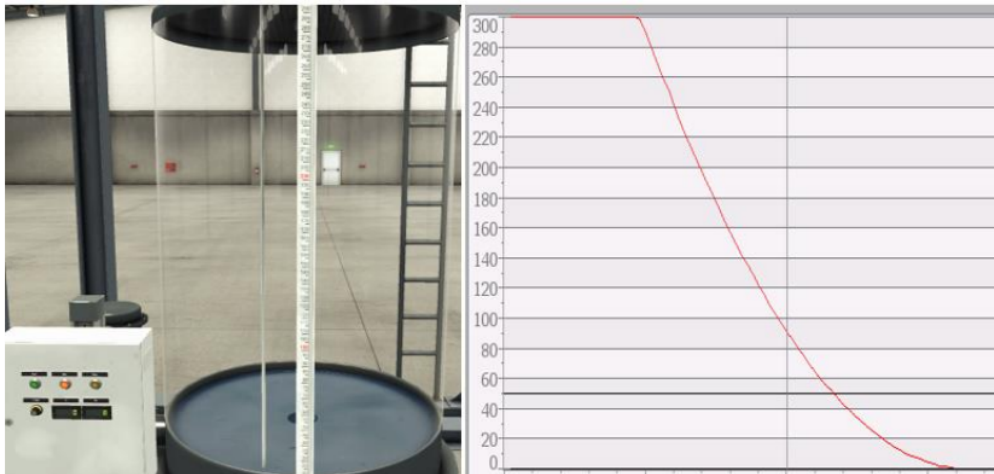


Figure 9; Test - From 300 manually drain to 0.

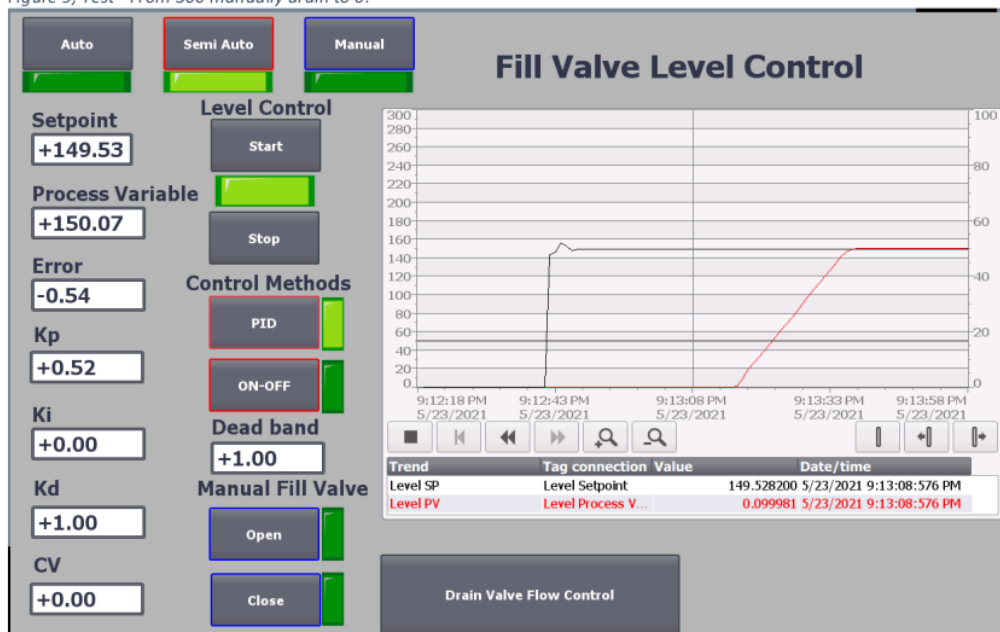


Figure 10; Test - Setpoint to 150

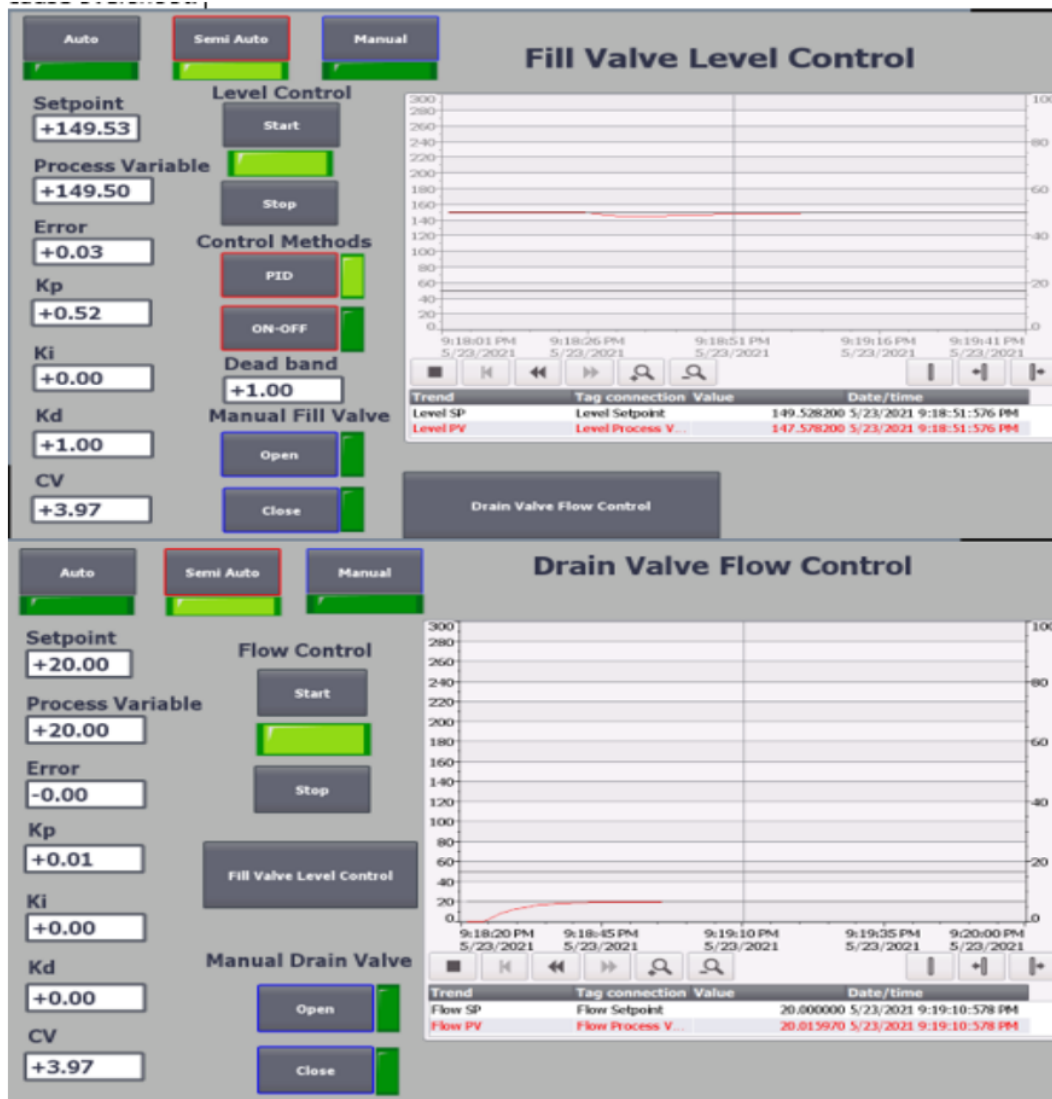


Figure 11; Test - Maintain L.SP of 150 with F.SP of 20.

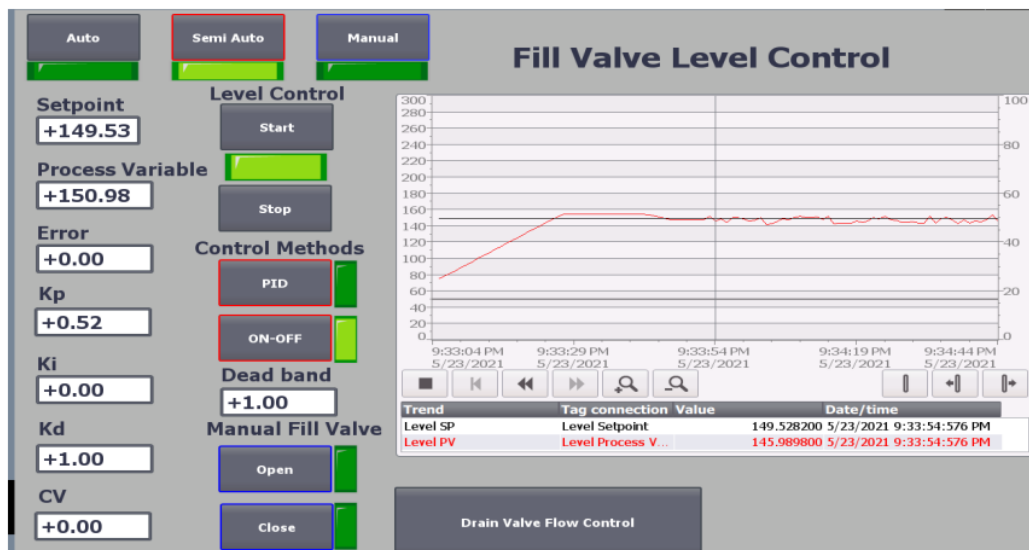


Figure 12; Test - Maintain L.SP of 150 and F.SP of 20 On-Off Control

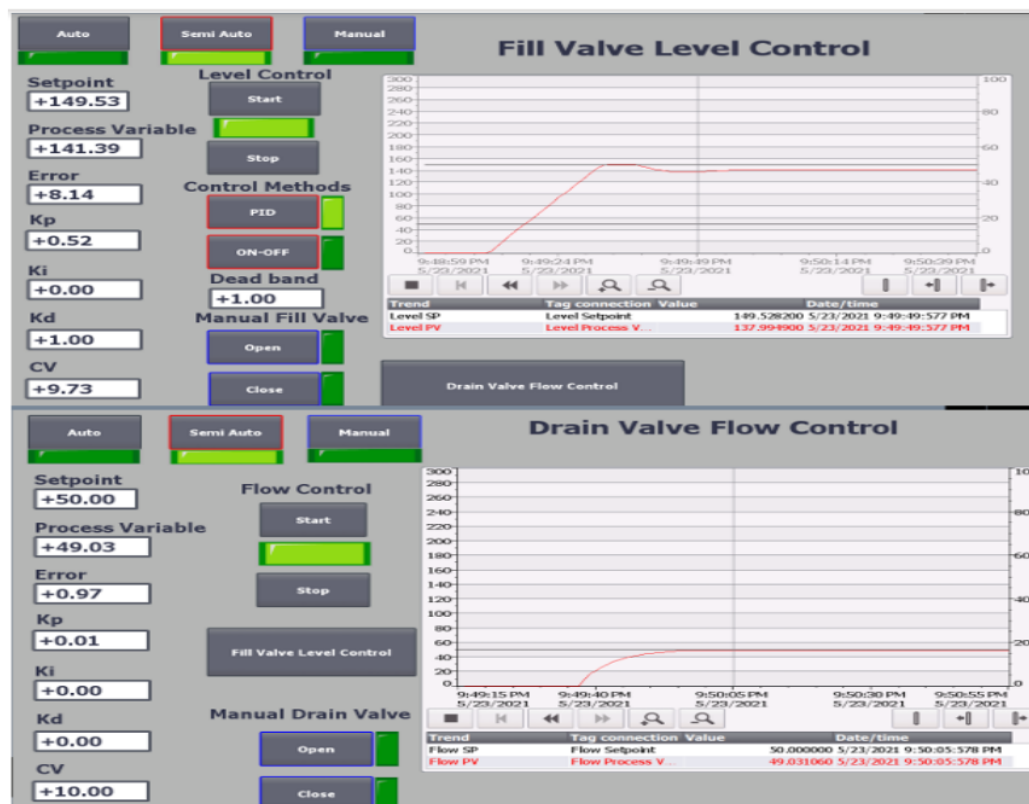


Figure 13; Test - Maintain L.SP of 150 and F.SP of 50 PID

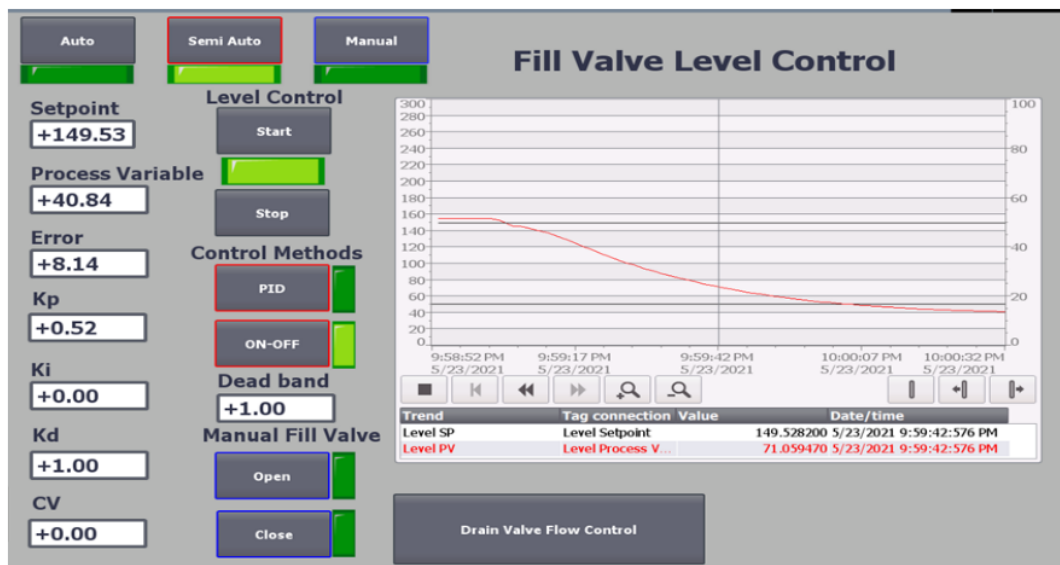


Figure 14; Test - Maintain L.SP of 150 and F.SP of 50 On-Off Control

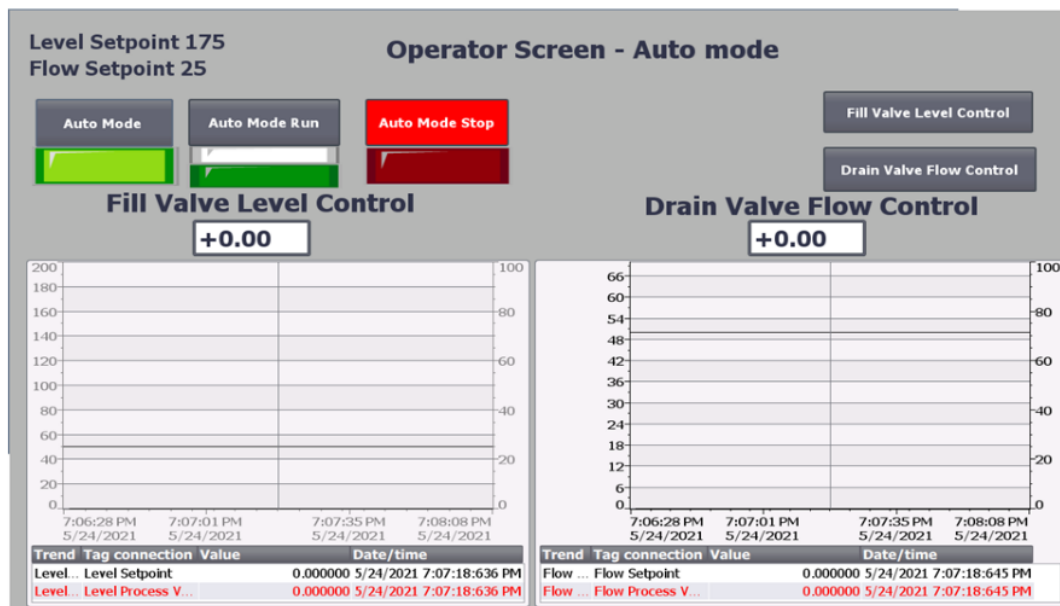


Figure 15; Test - Auto Start up

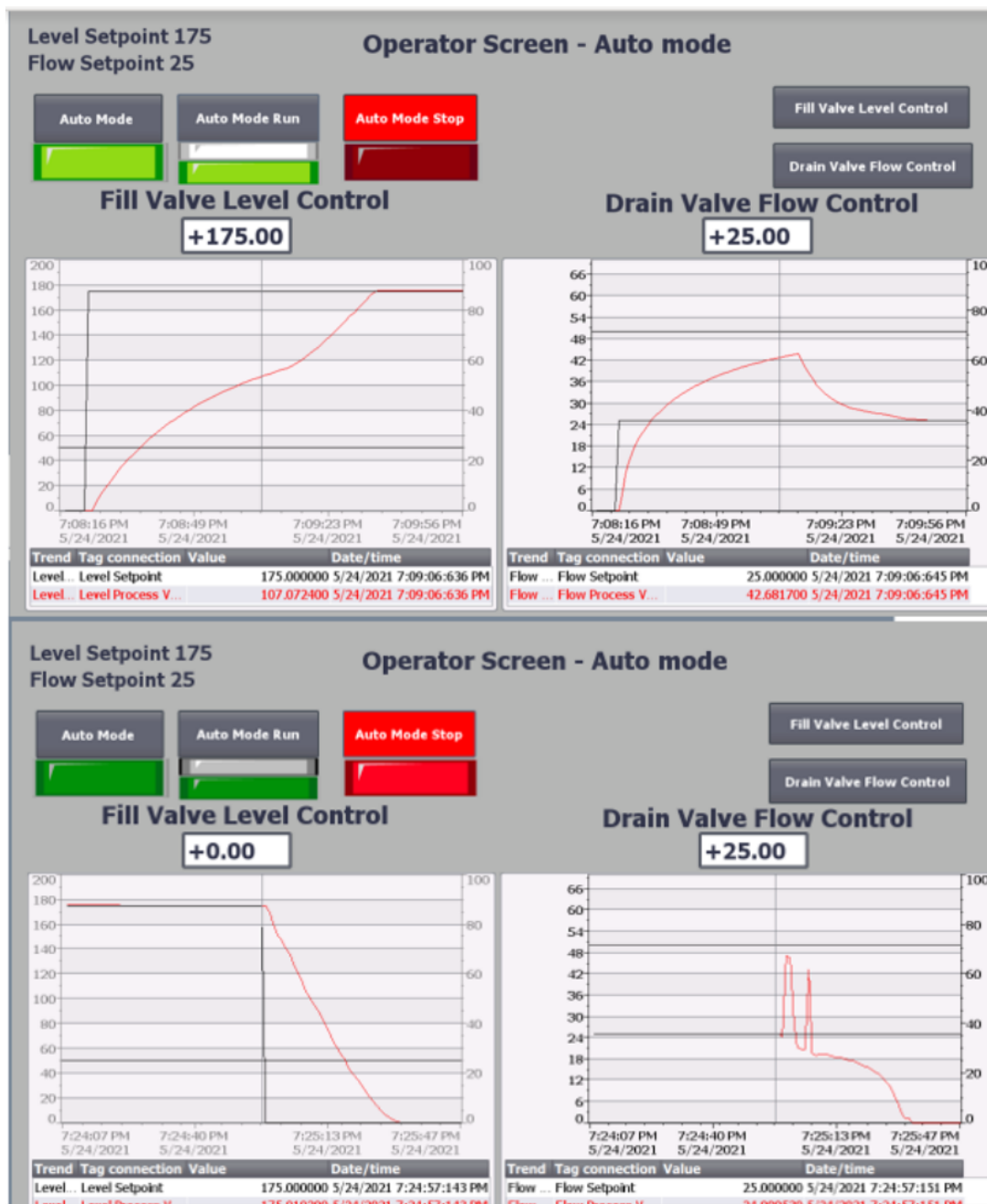


Figure 16; Test - Auto mode Setpoints 175 and 20 with dr

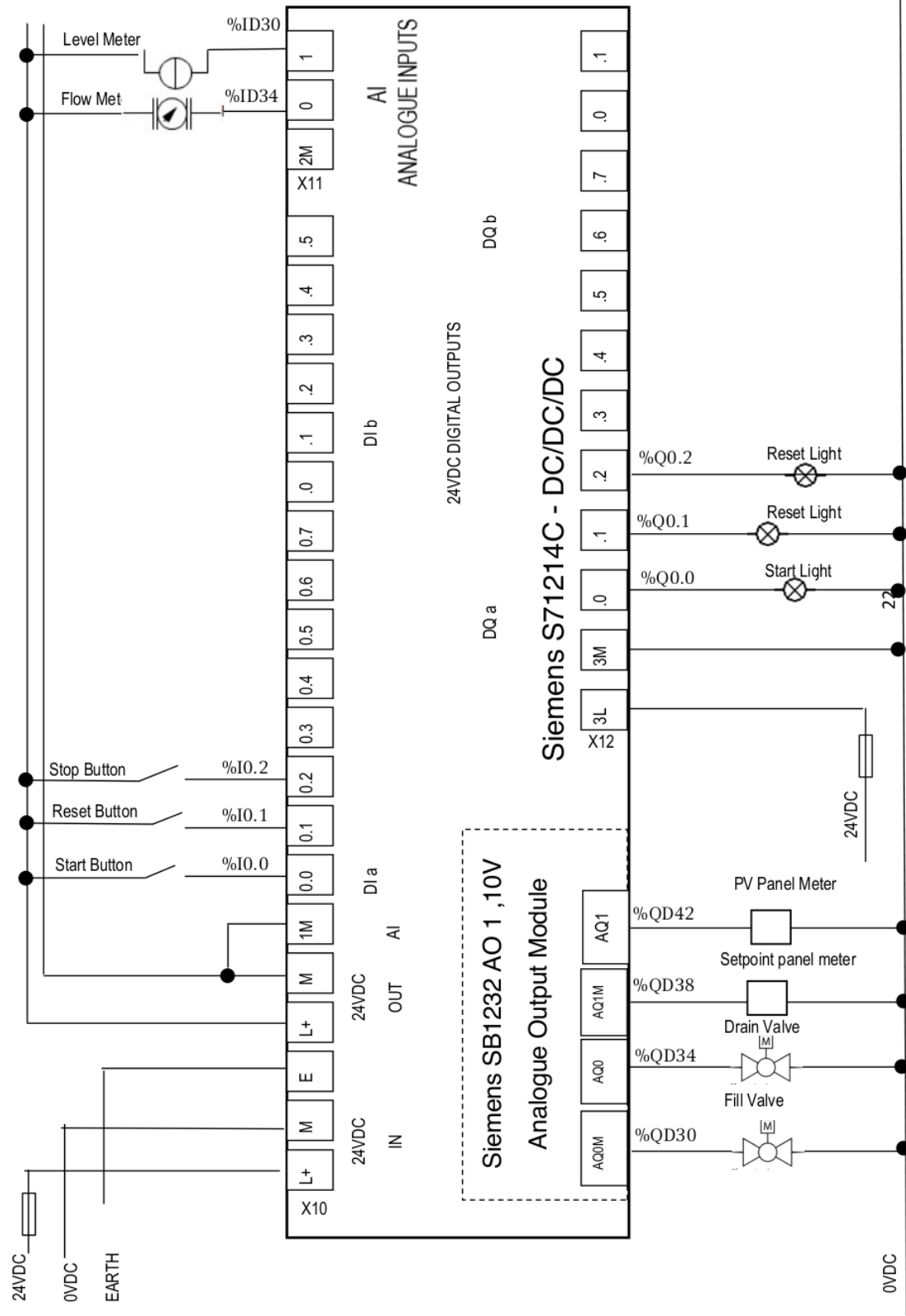


Figure 17: Wiring diagram PLC board, you should have added an extra module on the rail

PLC tags									
	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engi- neering	Supervision	Comment
	Start Button	Bool	%I0.0	False	True	True	True		
	Reset Button	Bool	%I0.1	False	True	True	True		
	Stop Button	Bool	%I0.2	False	True	True	True		
	Start Light	Bool	%Q0.0	False	True	True	True		
	Reset Light	Bool	%Q0.1	False	True	True	True		
	Stop Light	Bool	%Q0.2	False	True	True	True		
	Factory IO Running DI	Bool	%I0.3	False	True	True	True		
	Level Meter AI Real	Real	%ID30	False	True	True	True		
	Flow Meter AI Real	Real	%ID34	False	True	True	True		
	Fill Valve AQ Real	Real	%QD30	False	True	True	True		
	Drain Valve AQ Real	Real	%QD34	False	True	True	True		
	Setpoint Panel Meter	DInt	%QD38	False	True	True	True		
	Process Variable Panel Meter	DInt	%QD42	False	True	True	True		
	Setpoint Pententiometer	Real	%ID38	False	True	True	True		
	Level Control RUN	Bool	%M2500.0	False	True	True	True		
	Level PID Control	Bool	%M2500.1	False	True	True	True		
	Level Setpoint	Real	%MD3000	False	True	True	True		
	Level Process Variable	Real	%MD3004	False	True	True	True		
	Level Error	Real	%MD3008	False	True	True	True		
	Level Kp	Real	%MD3012	False	True	True	True		
	Level CV	Real	%MD3016	False	True	True	True		
	Level ON-OFF Control	Bool	%M2500.2	False	True	True	True		
	HMI Start Level Control	Bool	%M2500.3	False	True	True	True		
	HMI Stop Level Control	Bool	%M2500.4	False	True	True	True		
	HMI Manual Open Fill Valve	Bool	%M2500.5	False	True	True	True		
	HMI Manual Close Fill Valve	Bool	%M2500.6	False	True	True	True		
	HMI Manual Open Fill Valve Pulse	Bool	%M2501.0	False	True	True	True		
	HMI Manual Close Fill Valve Pulse	Bool	%M2501.1	False	True	True	True		
	Auto Mode	Bool	%M2550.0	False	True	True	True		
	Semi Auto Mode	Bool	%M2550.1	False	True	True	True		
	Manual Mode	Bool	%M2550.2	False	True	True	True		
	Flow Control RUN	Bool	%M2600.0	False	True	True	True		
	HMI Start Flow Control	Bool	%M2600.2	False	True	True	True		
	HMI Stop Flow Control	Bool	%M2600.3	False	True	True	True		
	HMI Manual Open Drain Valve	Bool	%M2600.4	False	True	True	True		
	HMI Manual Close Drain Valve	Bool	%M2600.5	False	True	True	True		
	HMI Manual Open Drain Valve Pulse	Bool	%M2600.6	False	True	True	True		
	HMI Manual Close Drain valve Pulse	Bool	%M2600.7	False	True	True	True		
	Level P Part	Real	%MD3020	False	True	True	True		
	Level Ki	Real	%MD3024	False	True	True	True		
	Level Integral Accumulator	Real	%MD3028	False	True	True	True		
	Level I Part	Real	%MD3032	False	True	True	True		
	Level Old Error	Real	%MD3036	False	True	True	True		
	Level KD	Real	%MD3040	False	True	True	True		
	Level Difference	Real	%MD3044	False	True	True	True		
	Level D Part	Real	%MD3048	False	True	True	True		
	Level Deadband	Real	%MD3052	False	True	True	True		
	Level Maximum Deadband	Real	%MD3056	False	True	True	True		
	Level Minimum Deadband	Real	%MD3060	False	True	True	True		
	Fill Valve Position	Real	%MD3064	False	True	True	True		
	Flow Setpoint	Real	%MD3200	False	True	True	True		
	Flow Process Variable	Real	%MD3204	False	True	True	True		
	Flow Error	Real	%MD3208	False	True	True	True		
	Flow KP	Real	%MD3212	False	True	True	True		
	Flow Ki	Real	%MD3216	False	True	True	True		
	Flow Kd	Real	%MD3220	False	True	True	True		
	Flow Old Error	Real	%MD3224	False	True	True	True		
	Flow Integral Accumulator	Real	%MD3228	False	True	True	True		
	Flow Difference	Real	%MD3232	False	True	True	True		
	Flow P Part	Real	%MD3236	False	True	True	True		
	Flow I Part	Real	%MD3240	False	True	True	True		
	Flow D Part	Real	%MD3244	False	True	True	True		
	Flow CV	Real	%MD3248	False	True	True	True		
	Drain Valve Position	Real	%MD3252	False	True	True	True		
	Auto Mode Run	Bool	%M2500.7	False	True	True	True		
	auto mode stop	Bool	%M2501.2	False	True	True	True		

PLC tags									
	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engi- neering	Supervision	Comment
	System_Byte	Byte	%MB2000	False	True	True	True		
	FirstScan	Bool	%M2000.0	False	True	True	True		
	DiagStatusUpdate	Bool	%M2000.1	False	True	True	True		
	AlwaysTRUE	Bool	%M2000.2	False	True	True	True		
	AlwaysFALSE	Bool	%M2000.3	False	True	True	True		
	Clock_Byte	Byte	%MB2001	False	True	True	True		
	Clock_10Hz	Bool	%M2001.0	False	True	True	True		
	Clock_5Hz	Bool	%M2001.1	False	True	True	True		
	Clock_2.5Hz	Bool	%M2001.2	False	True	True	True		
	Clock_2Hz	Bool	%M2001.3	False	True	True	True		
	Clock_1.25Hz	Bool	%M2001.4	False	True	True	True		
	Clock_1Hz	Bool	%M2001.5	False	True	True	True		
	Clock_0.625Hz	Bool	%M2001.6	False	True	True	True		
	Clock_0.5Hz	Bool	%M2001.7	False	True	True	True		

Auto_mode [FC1]

*Good attempt at auto
mode*

Auto_mode Properties

General

Name	Auto_mode	Number	1	Type	FC	Language	SCL
------	-----------	--------	---	------	----	----------	-----

Numbering

Automatic

Information

Title	Author	Comment	Family
-------	--------	---------	--------

Version	0.1	User-defined ID
---------	-----	-----------------

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
Auto_mode	Void		

```

0001 IF "Auto Mode" = 1 AND "Auto Mode Run" = 1 THEN // If the two buttons are pressed it will set the setpoints.
0002   "Level Setpoint" := 175; // Sets level setpoint to 175
0003   "Flow Setpoint" := 25; // Sets flow setpoint to 25
0004 END_IF;
0005 IF "Process Variable Panel Meter" < "Setpoint Panel Meter" THEN // If the process variable is less than the setpoint
0006   "Level Control RUN" := 1; // turn on level control run
0007   "Level PID Control" := 1; // turn on PID control
0008 END_IF;
0009 IF "Flow Process Variable" < "Flow Setpoint" THEN // If flow process variable is less than flow setpoint then
0010   "Flow Control RUN" := 1; // turn on flow control
0011 END_IF;
0012 IF "Auto Mode" = 0 AND "auto mode stop" = 1 THEN // if auto mode stop is pressed it resets auto mode, then :
0013   "Fill Valve AQ Real" := 0.0; // closes fill valve
0014   "Drain Valve AQ Real" := 10.0; // opens drain valve fully
0015 END_IF;
0016 // The tank will be empty, and can be restarted by auto mode run or other modes can be selected.

```

Symbol	Address	Type	Comment
"Auto Mode Run"	%M2500.7	Bool	
"auto mode stop"	%M2501.2	Bool	
"Auto Mode"	%M2550.0	Bool	
"Drain Valve AQ Real"	%QD34	Real	
"Fill Valve AQ Real"	%QD30	Real	
"Flow Control RUN"	%M2600.0	Bool	
"Flow Process Variable"	%MD3204	Real	
"Flow Setpoint"	%MD3200	Real	
"Level Control RUN"	%M2500.0	Bool	
"Level PID Control"	%M2500.1	Bool	
"Level Setpoint"	%MD3000	Real	
"Process Variable Panel Meter"	%QD42	DInt	
"Setpoint Panel Meter"	%QD38	DInt	

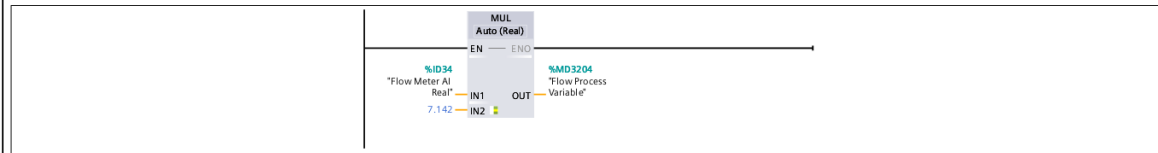
Flow Control Cyclic interrupt 1ms [OB30]

Flow Control Cyclic interrupt 1ms Properties

General							
Name	Flow Control Cyclic interrupt 1ms	Number	30	Type	OB	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					
Name		Data type	Default value		Comment		
▼ Input							
Initial_Call		Bool			Initial call of this OB		
Event_Count		Int			Events discarded		
Temp							
Constant							

Network 1: Read analogue input for flow

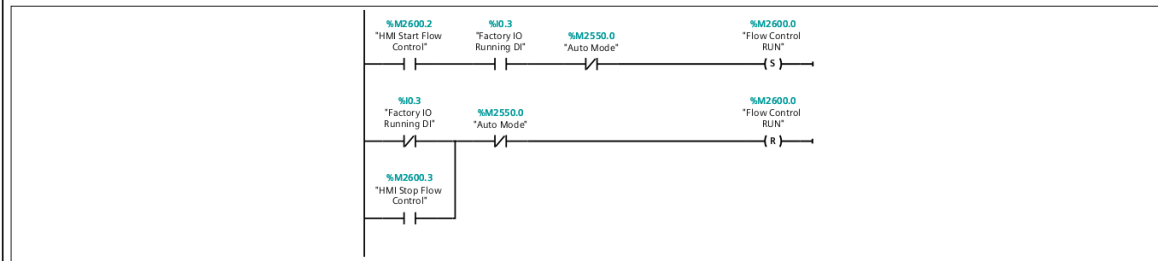
Inputs the signal from flow meter real and multiplies by 7.142, then displays on flow PV.



Network 2: Flow Control

Hmi start button pressed , factory io running, auto mode off starts flow control run.

If factory IO isnt running, auto mode is off then resets flow control run, in addition if the hmi stop flow control is pressed, then it will stop flow control.



Network 3: Manual Drain Valve Control

If flow control run is pressed, manual control pressed then manual open fill valve a pulse of 1v is sent to the drain valve to open slightly.

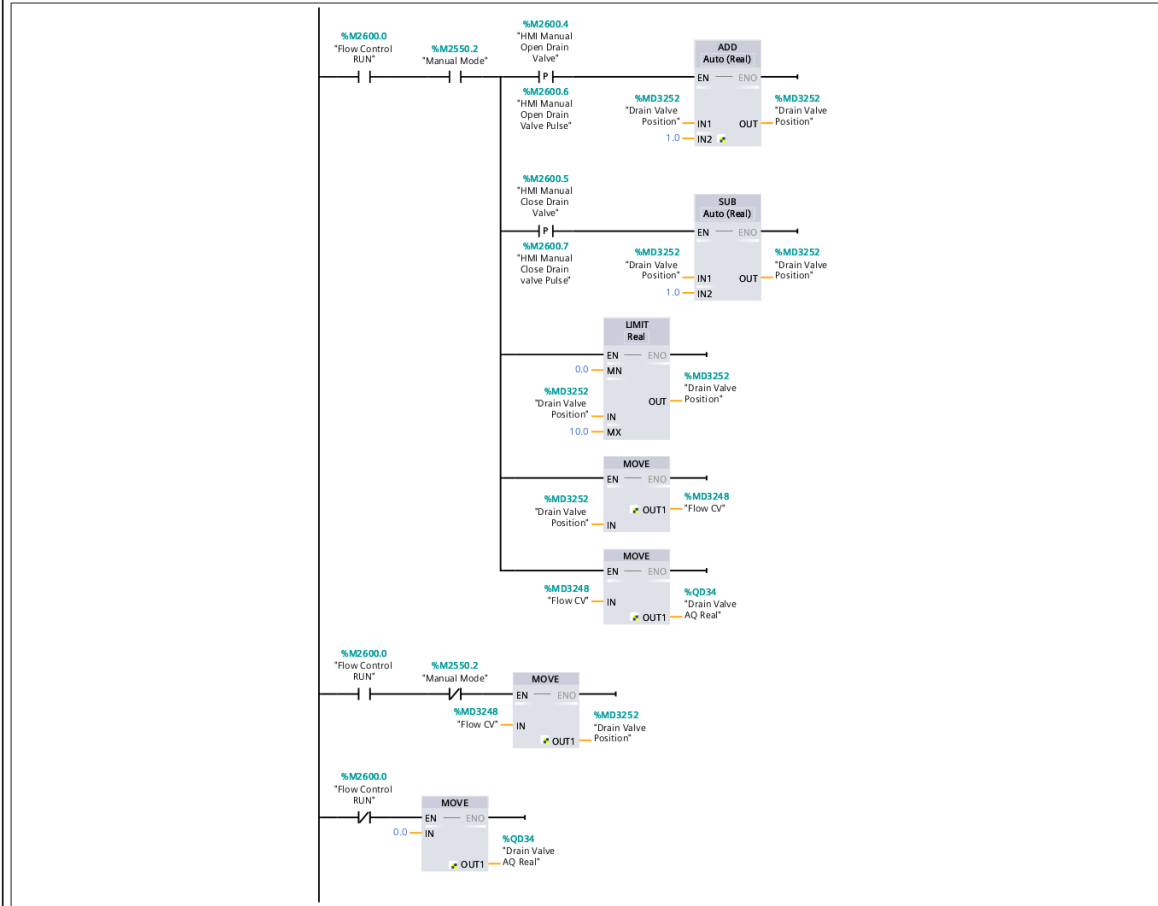
If flow control run is pressed, manual control pressed and hmi manual close then a pulse of 1v is sent to the drain valve to close slightly.

Limit is set 0-10V.

Drain valve position goes to CV and CV goes to the real fill valve position.

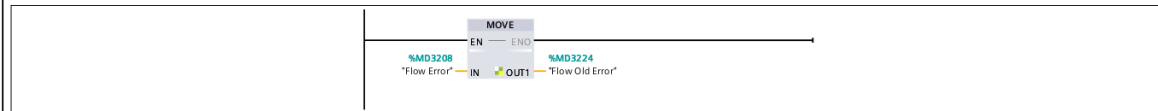
If flow control is on, but manual mode off then move flow CV to fill valve position.

If flow control run is off, then sent 0V to valve, meaning it will be closed.



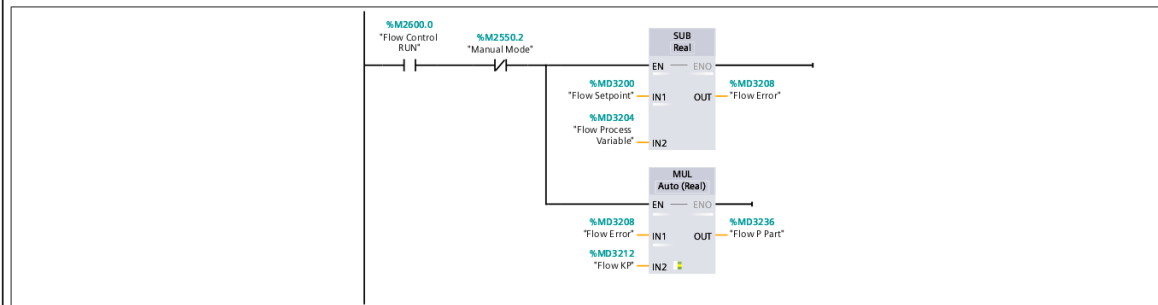
Network 4: Store the old error

This assists calculating the D part for PID control.



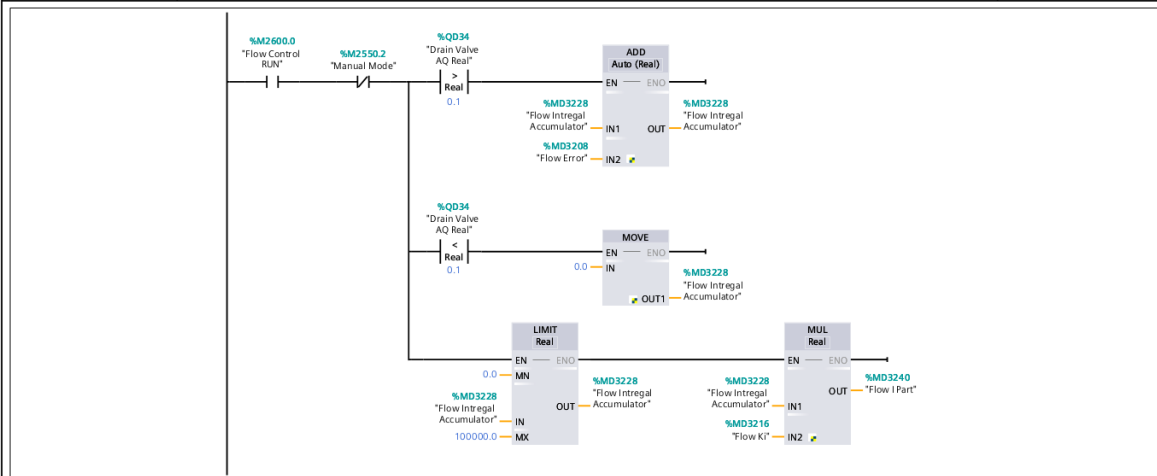
Network 5: Calculate the P Part

Flow control run is set, manual mode is off, then flow error is calculated by (flow setpoint) - (flow process variable) then the Flow P part can be calculated as (flow error) (flow KP)



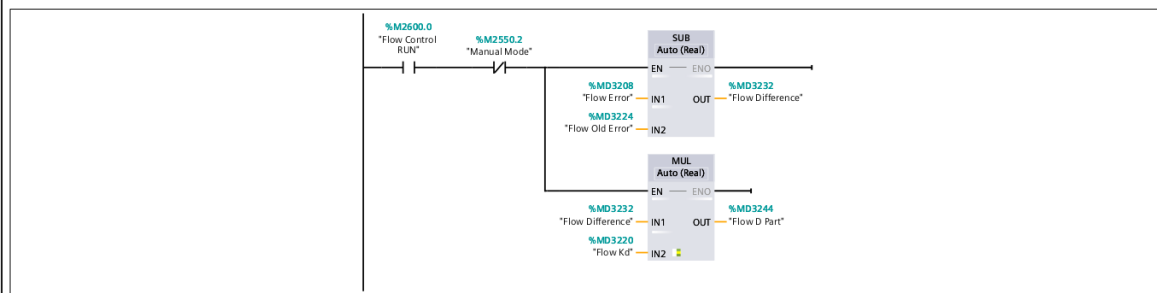
Network 6: Calculate the I Part

If flow control run is set, manual mode is off then - if the drain valve is less than 0.1 add flow integral accumulator and flow error to make new flow integral accumulator. If the drain valve is more than 0.1 send 0 to accumulator. Set the limit of the flow integral accumulator and create the flow I part by multiplying flow integral accumulator and flow Ki.



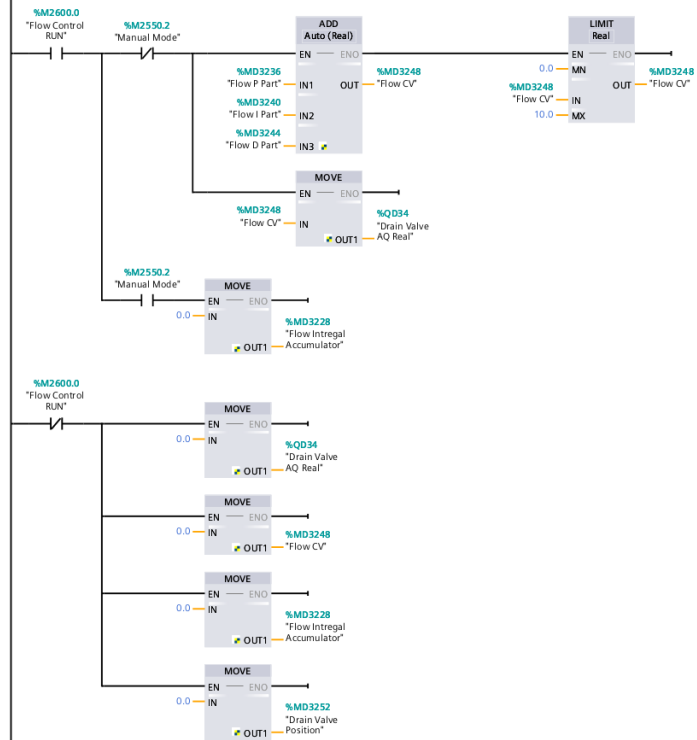
Network 7: Calculate the D Part

Flow difference is calculated when flow run is on and manual mode is off.
This is done by flow error - flow old error = flow difference.
The flow D part is then calculated by (Flow Difference)(Flow kd).



Network 8: Calculate the CV Part

Flow control sets, manual mode off, then adds the 3 parts together which creates flow CV. Set limits to 0-10V.
Flow cv is then moved to the drain valve position.
If flow control is turned off it resets all the bits and closes the valve.



Level Control Cyclic interrupt 1ms [OB31]

Level Control Cyclic interrupt 1ms Properties

General

Name	Level Control Cyclic interrupt 1ms	Number	31	Type	OB	Language	LAD
------	------------------------------------	--------	----	------	----	----------	-----

Numbering Automatic

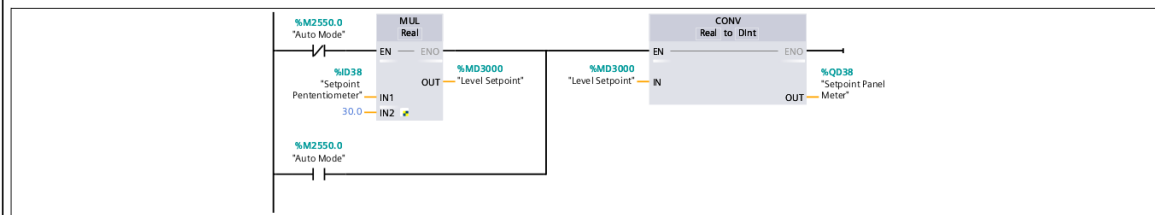
Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
Temp			
Constant			

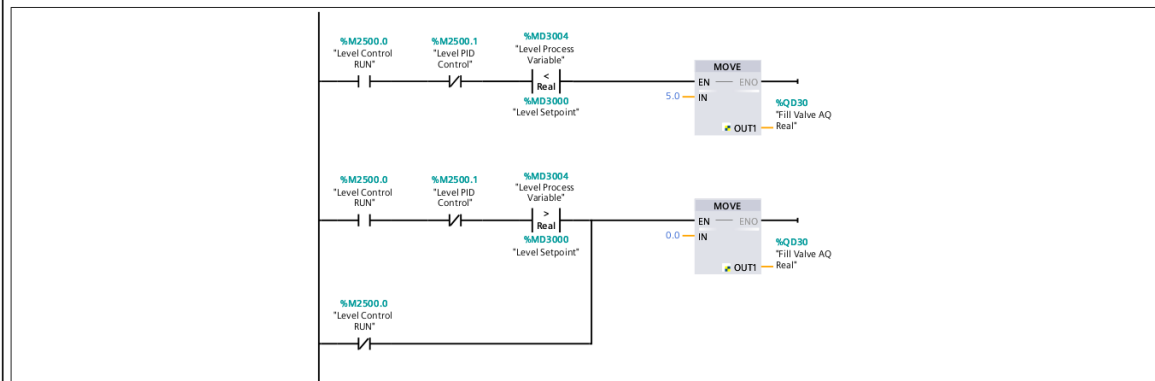
Network 1: Scale Level Setpoint and Display

This multiplies it by 30 and converts to be able to be displayed. This scales the analog input for the level setpoint from volts into millimeters by 10 so that 10V=300mm. If not in auto mode it will receive the setpoint from the potentiometer on the panel meter, if it is in auto mode then it will just display the setpoint.



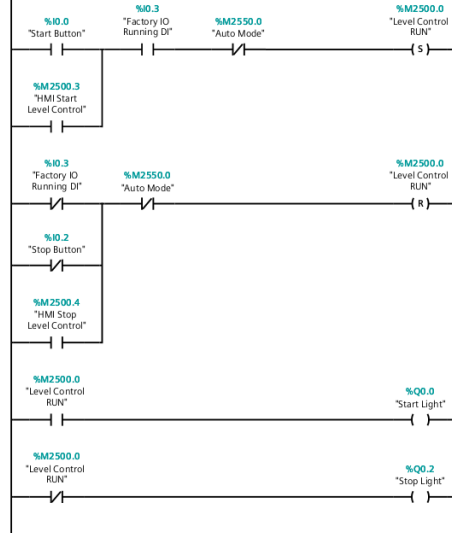
Network 2: Level ON-OFF Control

If level control run is on, PID is off then, if the process variable is lower than the required setpoint send a 10v signal to the fill valve to open all the way.
If level control run is on, PID is off the, if the level process variable is greater than level setpoint then 0V to the fill valve which closes it completely. In addition to this, if the level control is running, but if the user wants to turn it off then also send a signal of 0v to shut the valve.



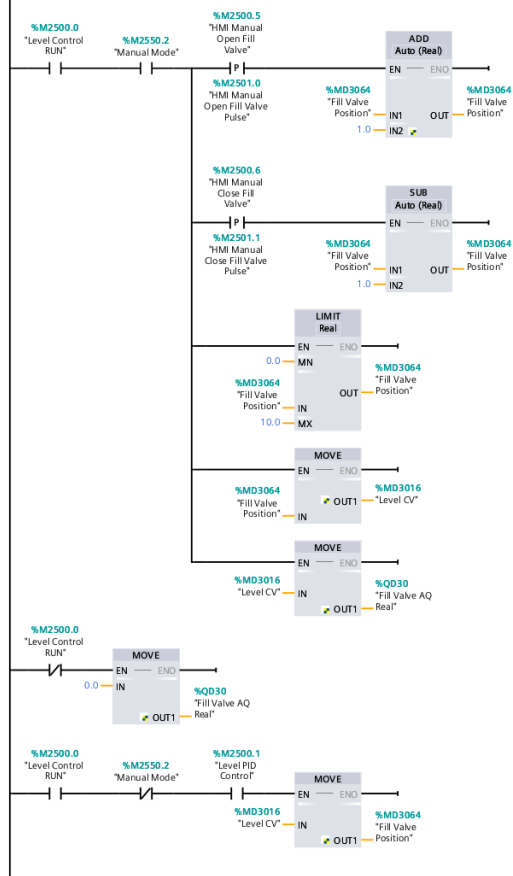
Network 3: Level ON-OFF Control Run

If factory IO is running, then it will start the level control run, if the stop button is pushed and Factory I/O is off it will reset the level control run.
The stop button needs to be N/C otherwise it will consistently turn it off.
This network also controls the lights, if level control run is on, then the start light will come on. If level control is off, then the stop light will come on.



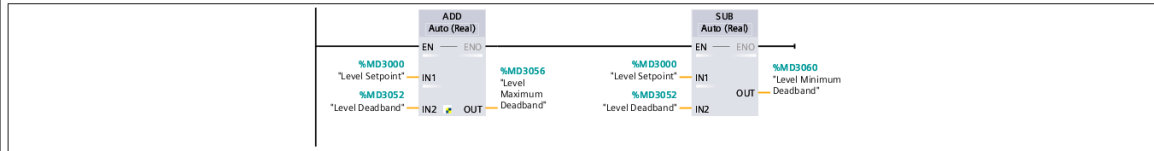
Network 4: Manual Fill Valve Control

When level control is set, manual mode is set, and open fill valve it will pulse 1v to the fill valve position.
If hmi close fill valve is set, manual mode is set, and open fill valve it will pulse 1v to the fill valve position.
A limit of 0-10V is set.
If level control is turned off it will move OV to the fill valve position for it to close completely.
If manual mode is off, level control is on and PID is on it will use the CV to control the position of the valve.



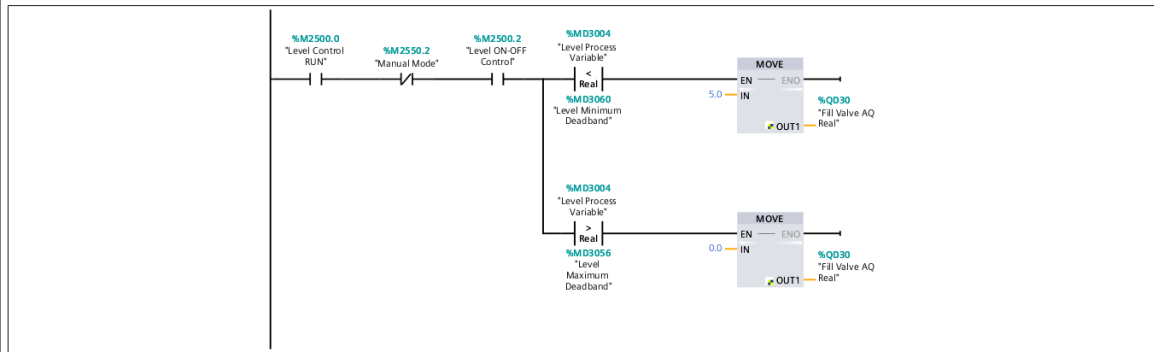
Network 5: ON - OFF Control Deadband

Sets maximum and minimum deadband.



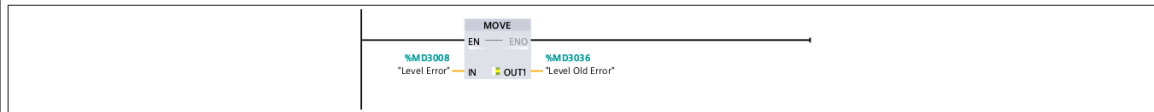
Network 6: Level ON - OFF Control With Deadband

If less than the minimum then adds 5 to the fill valve, if its more than the maximum then sends 0V to the fill valve to close it.



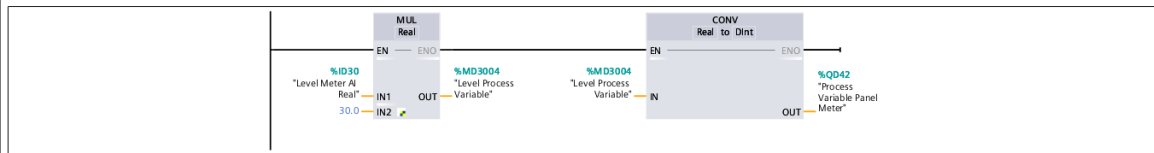
Network 7: Store The Old Error

This assists with calculating the D part.



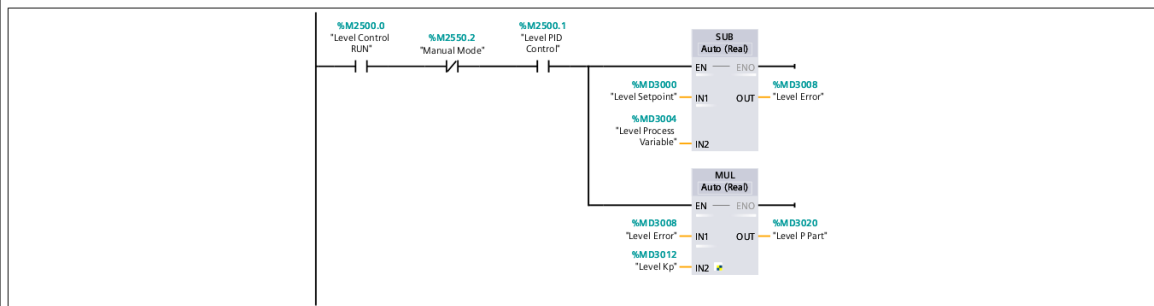
Network 8: Scale process variable and display

Displays how much water is in the tank reading from the analog sensor.



Network 9: Level PID Control - calculate error and the P Part

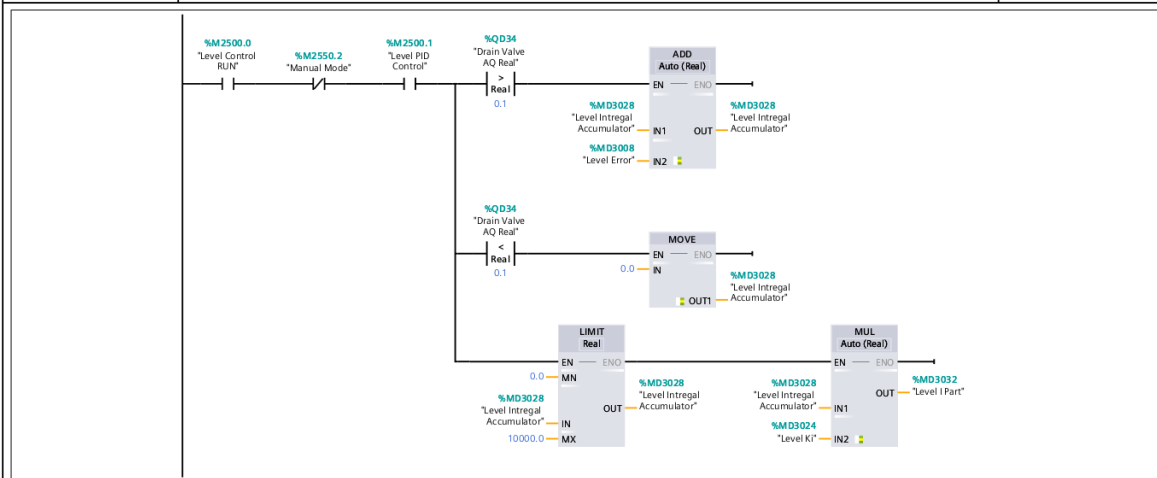
If level control run is set, manual off and level PID control on then:
Level setpoint subtracts level process variable to create level error.
Level error is then multiplied by level Kp to create the P part.



Network 10: Level PID Control - Calculate the I Part

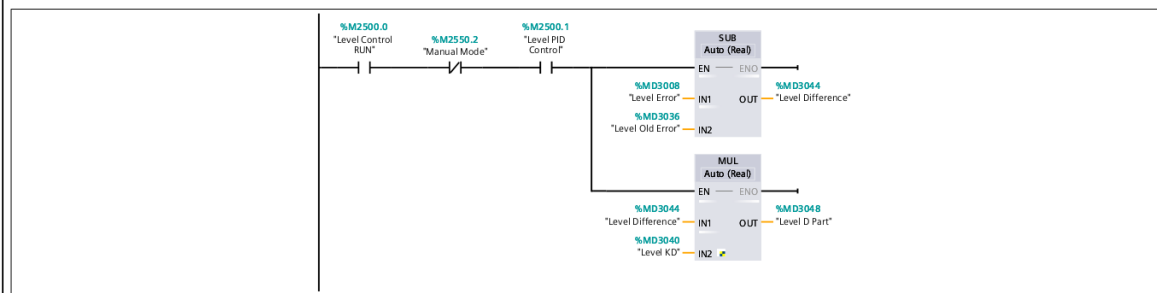
Switches I off if there is no flow out.
Limits accumulator to 10,000.





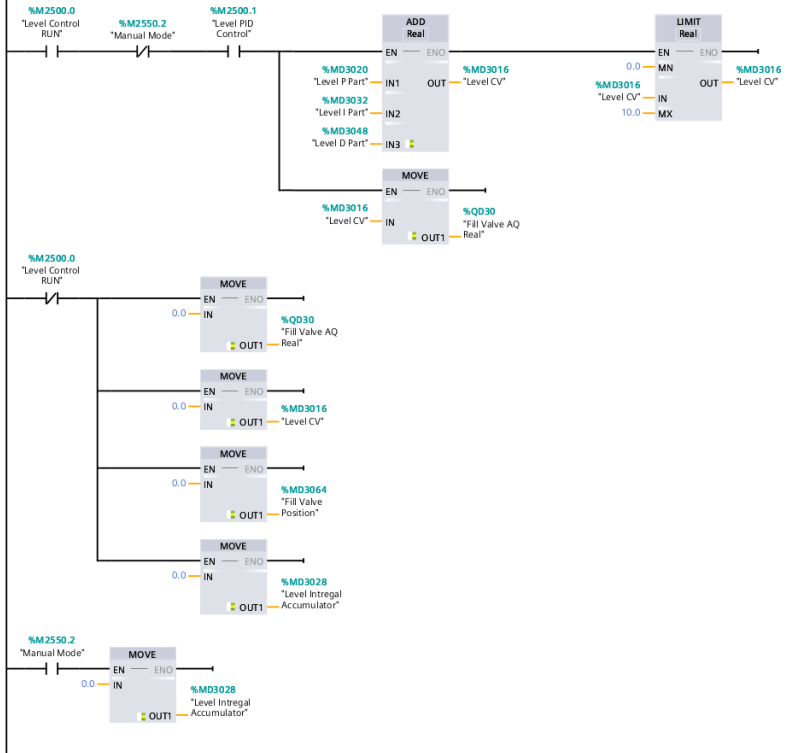
Network 11: Level PID Control - Calculate the D Part

If level run control is set, manual off and level PID control on then
Subtracts level error by level old error to create level difference.
Level difference is multiplied by level KD to create the D part.



Network 12: Level PID Control - Calculate CV

If level control is set, manual mode is off and level PID Control is on then P,I and D are added to create level CV. Limit set 0-10V for level CV.
Then the Level CV is moved to fill valve position to know how much or less to open/close.
If level control is off, then send 0V to fill valve to close.
If manual mode is selected, resets the level accumulator.
If level control run is off then resets the fill valve to 0, which closes it.



Totally Integrated Automation Portal		
--------------------------------------	--	--

Main [OB1]

Main Properties

General

Name	Main	Number	1	Type	OB	Language	LAD
Numbering	Automatic						

Information

Title	"Main Program Sweep (Cycle)"	Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
▼ Temp			
junk	Bool		
Constant			

Network 1 : Factory I/O

Automatic block created from factory I/O

%FC9000

"MHJ-PLC-Lab-Function-S71200"

EN

ENO

Network 2: Move Initial Values for Level PID

When the plc is turned on, then initial conditions are set which are:
Level setpoint is set for 200.
Level KP is set to 0.52
Level Ki is 0.00055
level Kd is 1.0
level deadband is 1.0.

%M2000.0

"FirstScan"

EN

ENO

200.0

IN

OUT1

%MD3000

"Level Setpoint"

0.52

IN

OUT1

%MD3012

"Level Kp"

0.00055

IN

OUT1

%MD3024

"Level Ki"

1.0

IN

OUT1

%MD3040

"Level Kd"

1.0

IN

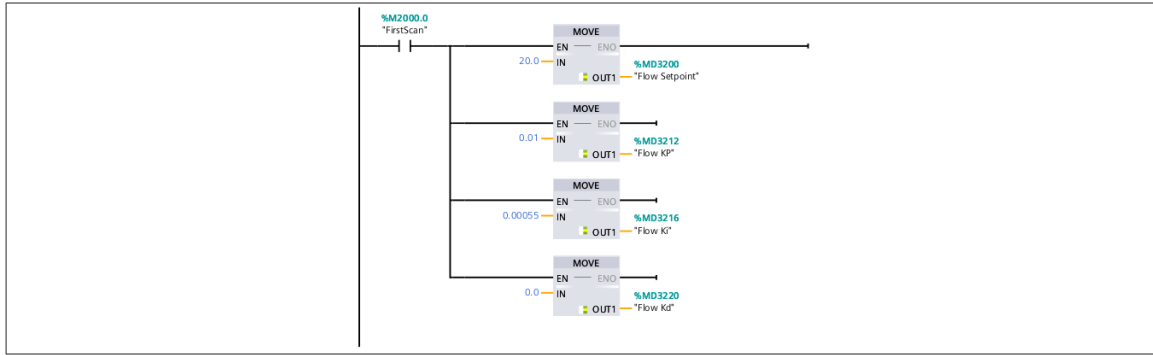
OUT1

%MD3052

"Level Deadband"

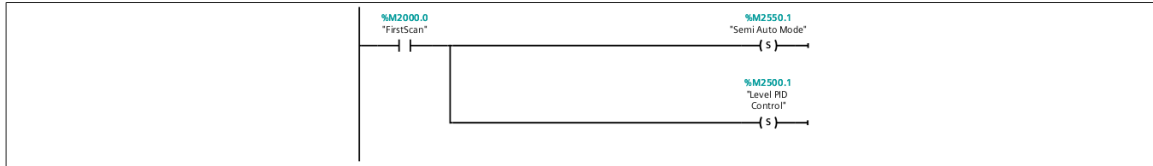
Network 3: Move Initial Values for Flow PID

When the plc is turned on, then initial conditions are set which are:
Level setpoint is set for 20
Level KP is set to 0.01
Level Ki is 0.00055
level Kd is 0.0

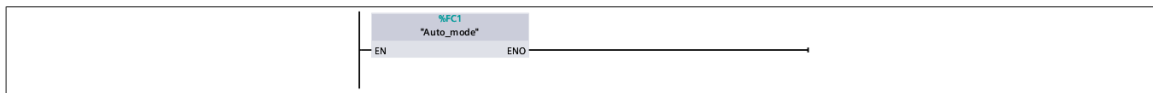


Network 4: Setting the modes on initial start up.

If first scan is closed then semi auto and level PID modes are set, meaning they are pre-selected.



Network 5: Calling auto mode from SCL function block.



FINAL GRADE

GENERAL COMMENTS

Instructor

78/100

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Improper Citation

Improper citation:
Improperly cited material. Please use the link below to find links to information regarding specific citation styles: http://www.plagiarism.org/plag_article_citation_styles.html

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

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.

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 5

PAGE 6

Text Comment. this is a tag list, some of this information is not needed in an IO list

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

Text Comment. There is only one analogue output on this signal board, you should have added an extra module on the rail.

PAGE 23

PAGE 24

PAGE 25

Text Comment. Good attempt at auto mode

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

GRADING FORM: NEW CSG-RUBRIC

MATTHEW PERRY

78

1. ENGAGEMENT WITH LITERATURE

- | | | |
|---|---|-----------|
|  | Engagement with an extensive range of relevant and credible literature. Consistently accurate application of referencing. (14/20) | 14 |
|---|---|-----------|

2. KNOWLEDGE AND UNDERSTANDING

- | | | |
|---|---|-----------|
|  | Excellent knowledge and understanding of the basic underlying concepts and principles of the subject. (15/20) | 15 |
|---|---|-----------|

3. COGNITIVE AND INTELLECTUAL

- | | | |
|---|---|-----------|
|  | Excellent evaluation of the underlying concepts and principles. Can interpret qualitative and quantitative data accurately and with some insight. Excellent ability to devise arguments using evidence and make appropriate and valid judgements. (16/20) | 16 |
|---|---|-----------|

4. PRACTICAL APPLICATION

- | | | |
|---|--|-----------|
|  | An advanced application of a range of taught, basic methods, materials, tools and/or techniques. The context of the application is well considered, and insightful. Excellent evidence of different approaches to problem-solving in particular contexts and proposes appropriate solutions. (17/20) | 17 |
|---|--|-----------|

5. TRANSFERABLE SKILLS



Work is coherent, very fluent and is presented professionally. Can work effectively independently and/or as part of a team, with an excellent contribution to group activities. Demonstrates excellent skills for employment requiring the exercise of some personal responsibility with an appetite for further development. (16/20)

16