EEE4118F – Process Control & Instrumentation

Lab Report – Part C: Introduction to HMI Project

Names	Student Number	Date of Practical	Station Number
Shaun Labuschagne	LBSSHA005	10/05/2022	6
Sarah Tallack	TLLSAR002		

Contents

Table of Figures

Introduction

This report details the programming of a Programmable Logic Controller (PLC) and design of a Human Machine Interface (HMI) for the control of input and reading of output from a servo motor system plant. The controller used will be the same as the one implemented and tested in practical 2. The details of how this controller works and the results obtained for testing are detailed in that report.

PLCs as controllers are popular in industry, as they are generally more compact, cheaper, and reliable than other digital control configurations, such as control via less dedicated processors or older relay/contactor systems. PLCs are highly configurable and can be programmed in a variety of ways. For this practical, ladder logic was used.

It is often of interest for a plant operator to be able to view certain plant parameters and adjust the controller's operation. For this reason, HMIs are used. They provide concise, visual output and allow adjustments to be made via input controls. They are also highly configurable. In this practical, a simple HMI was designed with a slider for input and a meter for output. Ideally, the angle of the output dial will also be displayed on the HMI screen, and not just the digital value range.

1. Laboratory Design Work

1.1. PLC and HMI hardware used

The PLC used is a Delta DVP-PLC unit, consisting of:

- DVP-12SE – the PLC, featuring 8 digital inputs and 4 digital outputs, alongside mini-USB, ethernet and RS485 ports.
- DVP-PSO1 – a switch-mode power supply and voltage regulator.
- DOP-B07E415 – a touch-screen HMI.

Since the PLC has its input and output limited to high and low logic states, an extension module is attached to the side of the PLC. The module is the DVP06XA Analogue Input/Output Converter. This ensures that the PLC can receive and output a voltage range of -10V to +10V.

1.2. Wiring between elements

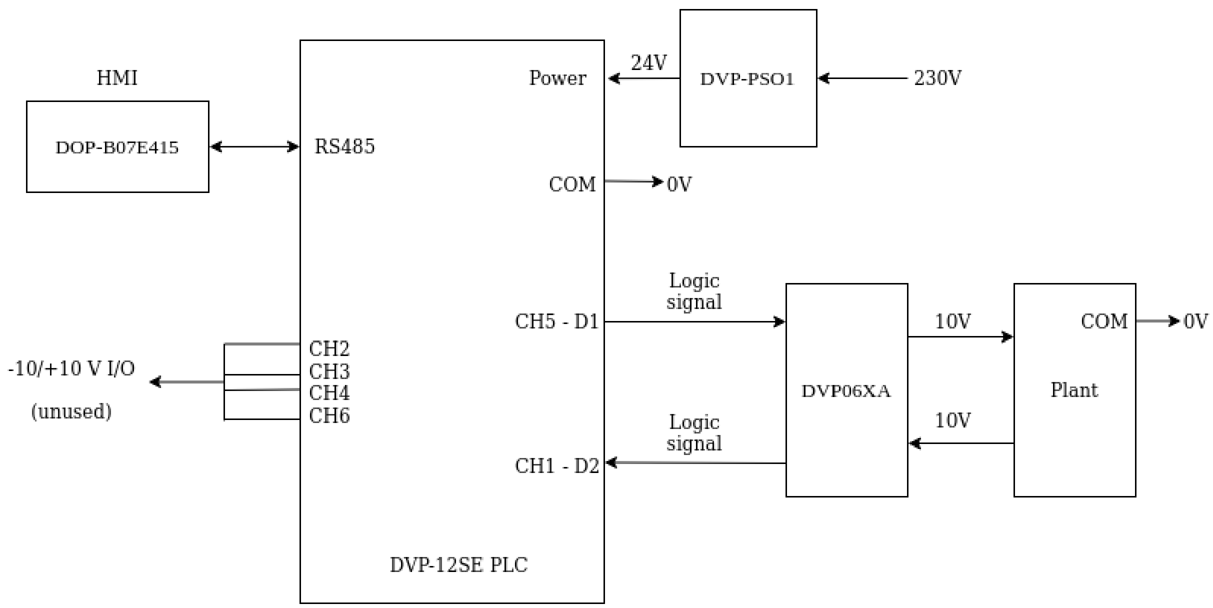

Figure 1: Wiring between elements of system (IEEE Standard)

1.3. Steps used in HMI design and programming

1.3.1. PLC programming

PLC programming was done through ISPSoft software. To begin programming the PLC, the extension module must have its Control Registers configured. Firstly, the input/output mode settings need to be configured for different channels. This sets whether the channel is an input or output, and also set the voltage limits of the channel. For the input channels (CH1 to CH4) the inputs are limited to -10V to +10V. For the output channels (CH5 and CH6) the outputs are limited to 0V to +10V.

Channel 5 is then configured as a write register, and D1 is set as the address where these values will be written to. The values outputted by this channel are used to set the setpoint of the system.

Channel 1 is configured as a read register, and is D2 is set as the address where it will read from. The values inputted to this channel correspond to the output position of the system.

Once these configurations have been made and applied, ladder networks are automatically generated by ISPSoft. The ladder logic diagrams are shown in Figure 2. The program is then compiled, to ensure that there are no errors. Once the program has been compiled it is downloaded to the PLC.

1.3.2. HMI design

HMI design was done through DOPSoft 4 – this software allows users to design realistic interfaces and simulate them or download them to an actual connected PLC's screen.

A new project was created, then the controller and communication parameters were configured. Next, HMI elements are chosen. For input, a slider as well as a numerical input block is added. To display the output, a meter is used.

The slider and input block are configured to write to D1, with the data type set to word, and the data format set to unsigned decimal. The limits are set to 0 (minimum) and 4000 (maximum). The meter is configured to read from D2, with the data type set to word, and the data format set to signed decimal. The limits are set to -2000 (minimum) and 2000 (maximum).

Once all of the elements had been placed and configured, customization to the HMI was done. The colours of each element were changed to the group's preference, and labels for the input and output of the system were placed.

The project was then compiled to check for errors. Once this was done, the HMI was simulated on the computer. The reason the HMI screen could not be downloaded to the PLC screen is because of an authentication barrier at station 6.

1.4. Ladder Logic Diagram

The Ladder Logic Diagram can be found in Figure 2 below.

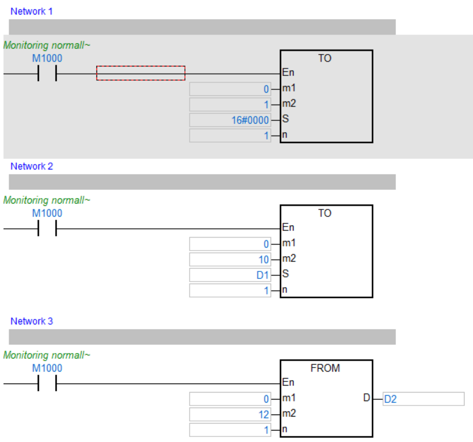

Figure 2: Ladder Logic Diagram

Network 1 represents a write register, namely #1 I/O Mode Setting. This sets the input and output voltage modes of the channels on the DAC.

Network 2 represents a write register, namely #10 CH5 Output Value. This is used to set D1 as the address which will set the output voltage of channel 5 (CH5). This can take values from 0 to 4000, and serves as a digital to analog converter.

Network 3 represent a read register, namely #12 CH1 Input Present Value. This is used to set D2 as the address where the voltage of channel 1 (CH1) is stored. This voltage is converter from analog to digital, and corresponds to the output position of the servo motor. It can store values between -2000 and 2000.

In all of the networks, M1000 ensures that the API blocks are constantly enabled when the PLC is running, as it always has a state of 1.

1.5. PLC Communication Protocol

The Delta DVP-PLC unit used in this practical makes use of the RS-485 communication protocol. This is a serial communication protocol which is an improved version of the older RS-232 protocol. RS-485 is used in control networks with multiple devices as a communication bus to connect them, as well as data transmission up and down the line. It is robust in that cabling can extend up to 1200 meters and is not highly susceptible to noise.

In the practical, the lab PC is connected to the PLC via RS-485 protocol with a DB9 connector. Data is transmitted in binary. After programming the ladder logic for the PLC's operation in ISPSoft and compiling, it is downloaded to the PLC via the cable, serially (one bit at a time). Similarly, after HMI design is completed in DOPSoft, it is also downloaded to the PLC's touch screen via the cable, serially.

2. Final HMI and Operation Results

The final HMI design is shown in Figure 3 below.

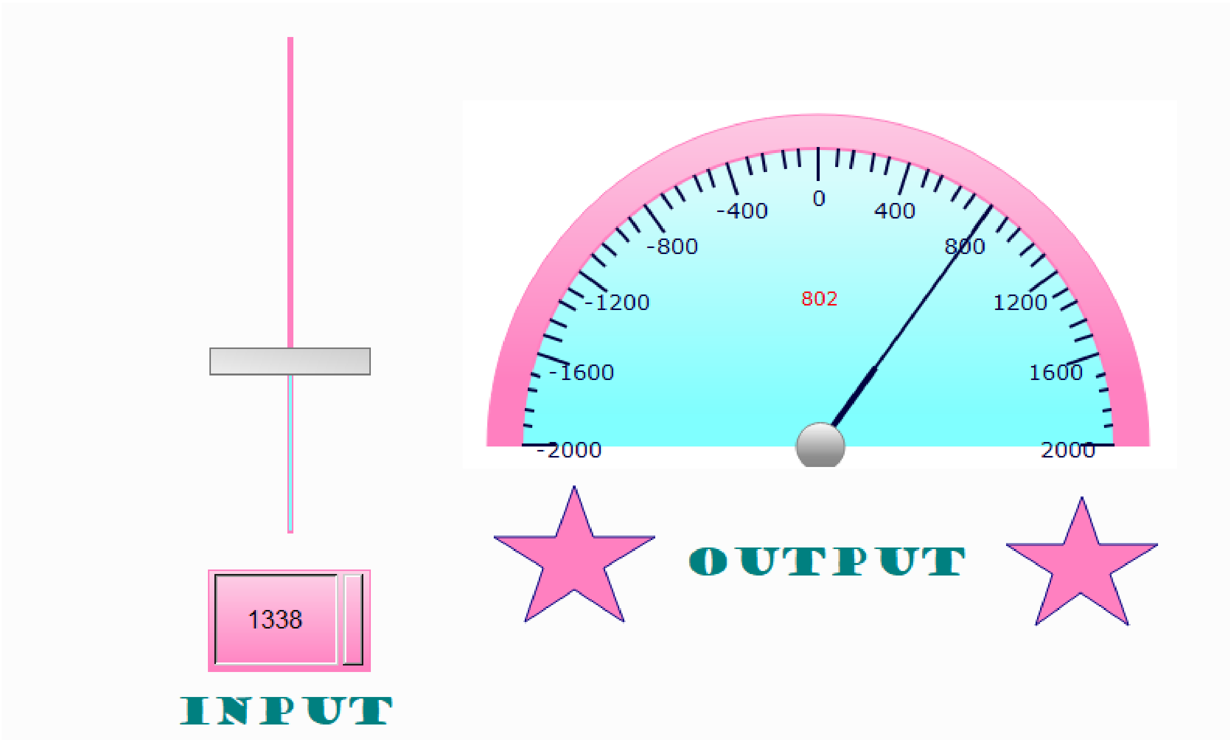

Figure 3: Final HMI Design

To operate the system with the PLC, the C# controller program on the lab PC is started. Once the controller is configured, the HMI is simulated and tested. As expected, when the slider or input block are changed, the output position of the system changed accordingly. One thing that didn't work was the range of outputs. It was found that the output never went below 0, and so the position had a limited output.

This is due to the way that the control register was configured for the external module. After doing the design and demo, it was noted that the output channels had their voltages limited to 0V to +10V. This means that a negative voltage input to the DAC cannot be achieved with the current register configuration.

Despite this issue, the HMI screen still operated as expected. The system was reacted when the slider was moved, or when a number was entered into the input block. The output meter changed when the output position changed.

Conclusion

Overall, the main aims of this practical were achieved. A usable HMI screen was designed, and the PLC was adequately implemented. Input commands transmitted from the HMI to the plant, via the PLC, resulted in a change of setpoint and a corresponding change in output position. This change in output position was displayed on the HMI.

Due to time constraints, a meter to display the angle of the output position could not be calibrated. This meant that the meter displaying output did not truly represent position, but rather the digital value that corresponded to a particular output voltage.

The PLC worked as expected, for the most part. The main issue arose in the fact that for all input values, no negative voltage values were being outputted. This was found to be due to the configuration settings of the control register for the channel modes. Due to time constraints, this could not be confirmed in the laboratory, however it seems likely that this is the issue. Should the range of voltages for the output channels be changed, it is likely that the full range of output positions could be achieved.