**PRACTICAL MANUAL 1**

**EEE4118F**

**INTERFACING THE DELTA DVP-PLC**

**WITH AN HMI SCREEN**

**AUTHORS:**

**Dino Kingsley-Rizzo**

**John Mkwebo**

**Katerina Itopoulos**

**Nicolen Pillay**

**Dominic de Maar**

**Contents:**

1. Pre Reading 3-6

1. Pre-Practical 7-18

# 3. Practical 19-26

3.1 Setting up the PLC 19-25

3.2 Creating a Usable HMI 26

3.3 Creating a digital controller in Visual Studio 26

4 . Table of Devices 27-30

1. Pre Reading

Get to Know the Delta DVP Series

## **Delta DVP-PLC**

The Programmable Logic Controller (PLC) is a control system that makes use of electronic operations for automation. Delta’s DVP series of PLC’s offers high-speed, stable and reliable applications for various industrial automation machines. The DVP-PLC supports many different kinds of communication protocols as well as being able to connect to Delta’s AC motor drive. The device also has a human machine interface (HMI) and temperature controller.

The following hardware is present in the DVP-PLC:

* [DVP-12SE](bookmark://_DVP-12SE/)
* [DVP-06XA](bookmark://_DVP-06XA/)
* [DVP-PS01](bookmark://_DVP-PS01/)
* [DOP-B07E415](bookmark://_DOP-B07E415/)

The following software is used for programming the DVP-PLC:

* [ISPSoft](bookmark://_ISPSoft/)
* [DOPSoft](bookmark://_DOPSoft/)
* [DCISoft](bookmark://_DCISoft/)

## **Hardware**

### DVP-12SE

The DVP-12SE is the most complete network type PLC in the industry. This device can be applied to remote device monitoring, production line monitoring, building automation, container fillers (recipes) and more.

**Features:**

* Eight digital inputs
* Four digital outputs
* Program memory of 16,000 steps • Supports DVP-S series left-side and rightside modules
* Has a mini USB port, Ethernet port and two RS-485 ports
* No battery required for maintenance of DVP-06XA



### DVP-PS01

The DVP-PS01 is a Delta DIN rail power supply. A DIN power

supply is a switched mode power supply that

converts unstabilised input voltage to regulated output

voltage.

**Features:**

* Nominal output voltage of 24V

* Temperature range from -20°C to +75°C

* Minimum holdup time: 20ms
* Overvoltage, overload and thermal

protection

* 24W
* Input voltage range:
  + 85 – 264 VAC (1 phase)
  + 320 – 575 VAC (3 phase)

### DOP-B07E14/DOP-107EV

The DOP-B07E14 and DOP-107EV are touch screen HMIs.

**Features:**

* Three COM ports that support RS-

232, RS-422 and RS-485

* The COM ports support Modbus

ASCII/RTU communication o For RTU communication, data length should be 8 bytes • Data can be transmitted or

downloaded through RS-232 (9-pin

male D sub connector), USB or Ethernet cable

* Supports SD cards, Ethernet and

Audio output (MP3 and WAV files)

* Resistant to water (IP65) • Supports horizontal and vertical displays

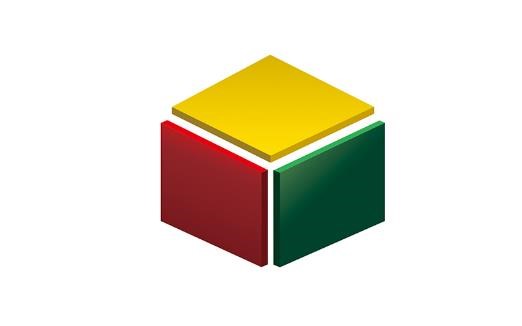
**Note:** Cannot use USB communication port as master

**Communication Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **COM1** | **COM2** | **COM3** |
| Function as master or slave? | Yes | Yes | Yes |
| Can be used to change number of data bits? | Yes | Yes | Yes |
| Can be used to change a parity bit? | Yes | Yes | Yes |
| Can be used to change number of stop bits? | Yes | Yes | Yes |
| Data Rates | 115200 bps | 921 kbps | 921 kbps |

## **Software**

### ISPSoft



ISPSoft is a software development tool used for PLC. Supported programming languages:

* LD: Ladder Diagrams
* SFC: Sequential Functions
* FBD: Functional Block Diagrams
* IL: Instruction Lists
* ST: Structured Text

**Note:** It is possible to use more than one programming language in a project.

**Built-in Configurations**

1. HWCONFIG: used to configure hardware for a system
2. NWCONFIG: used to configure network for a PLC system
3. Card Utility:used to backup and restore a system through a management wizard and memory card.

**Project Framework**

1. Single Project:
   * Only one model developed
   * File extension for single project:
     + .isp
2. Group Project
   * If several devices are connected to a network
   * An unlimited number of projects can be created in a group • Configure network using NWCONFIG • File extension for group project:
     + .pri
   * File extension for projects inside group:
     + .isp

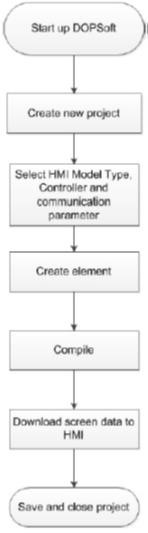
### DOPSoft

DOPSoft is a software interface that can be used to create and edit HMI screens easily.

### DCISoft

DCISoft is integrated configuration software for Delta network modules. Its is used for the integration of other plc modules and handles the communication between them.

**Flowchart for creating a project** **on DOPSoft**



**Figure 2 Flowchart for creating project in DOPSoft**

# Setup of Software Tools

Each hardware module has an associated software tool used for programming and project creation.

The Delta **PLC** (DVP12SE) is programmed using *ISPSoft* application.

The Delta **HMI** is programmed using the *DOPSoft* application.

# 2. Pre-Practical

**Starting a New Project for the PLC**

## **Setting up the Workspace**

All the necessary software will be downloaded to the PCs in the lab. Make sure all the hardware is properly connected (removed any cables and re-insert just to make sure. Just remember where each cable goes). It is also very important to take note of the model name and number you are using. For example, some HMI’s are different (some are newer models while others are older and may be outdated or slightly more difficult to use). Apart from this, the HMI should automatically update its firmware but it is also equally important to check that it is indeed updated. These are important as this will determine what version of software you must use. This will be explained when creating your HMI.

Start up ISPSOFT which is the software used for compiling and downloading onto the PLC. The version that was used in documenting this pre-practical is ISPSoft 2.05.

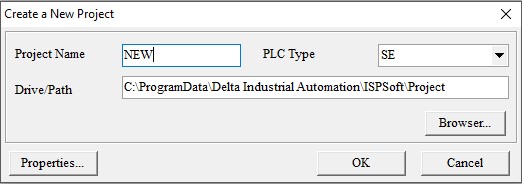
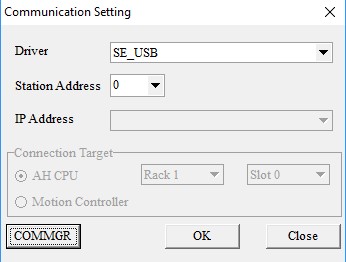
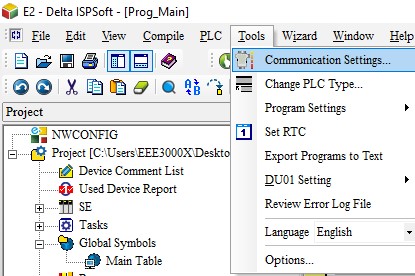
Before starting a new project ensure that the communication settings have been configured.

Click on **Tools > Communication Settings**

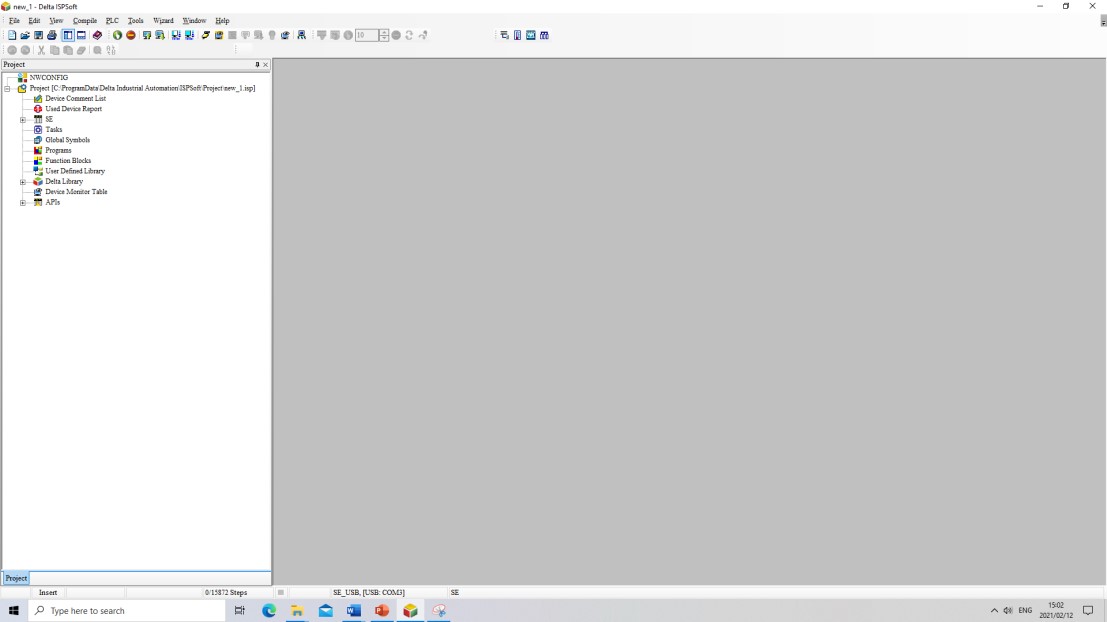
In the **Communication Settings** window, in the Driver list select **SE\_USB** from the drop-down list and click **OK**.

Click on

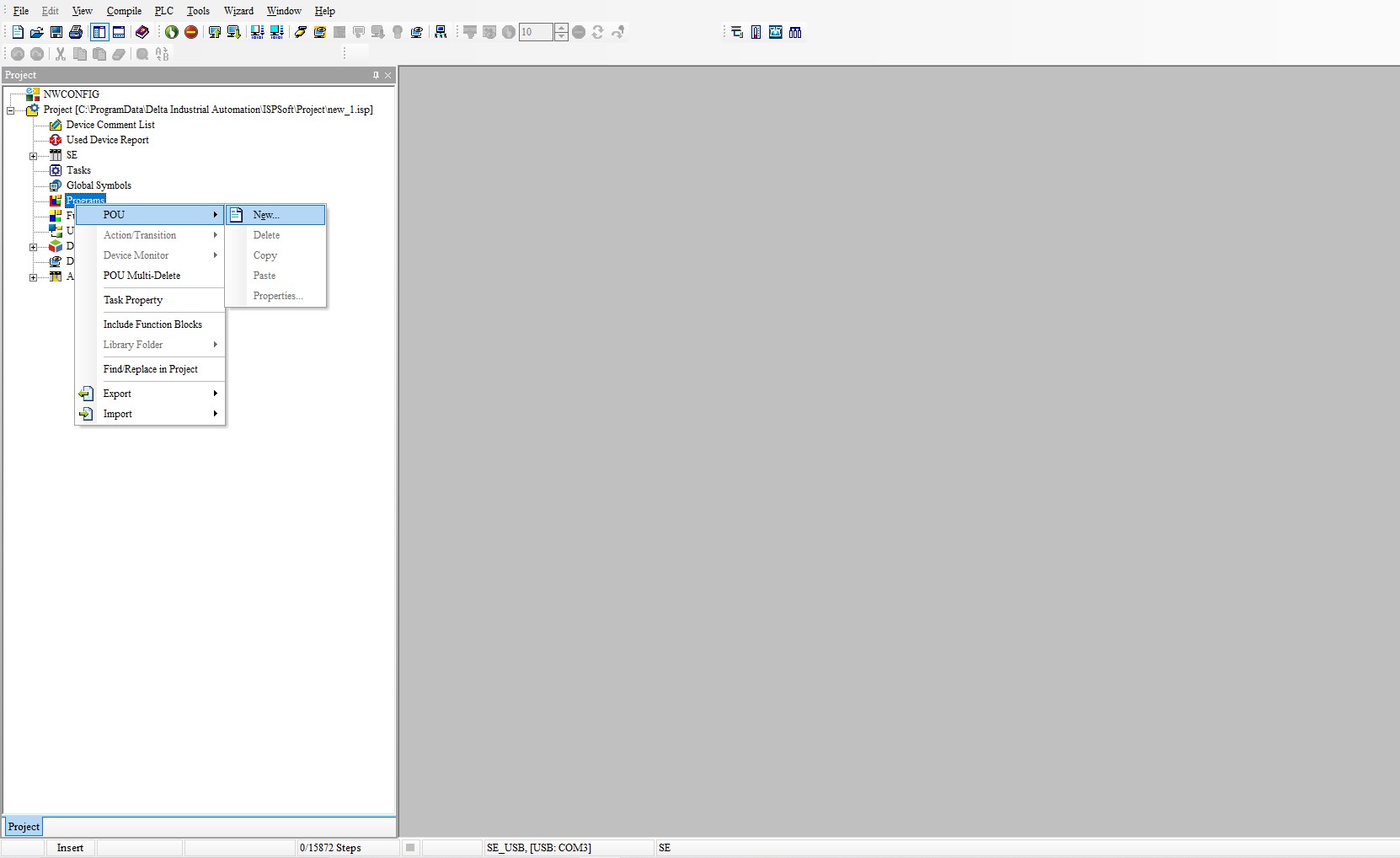
in the toolbar to add a new project. The following window will appear:



Rename your project and select SE in the PLC Type drop down list. Once ok is clicked a project environment will be displayed.



To create a main program, go to the project management area (on the left) and right click on Programs. From here you can select POU and click New.



1

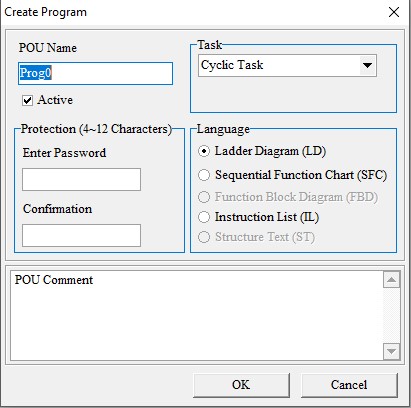


2

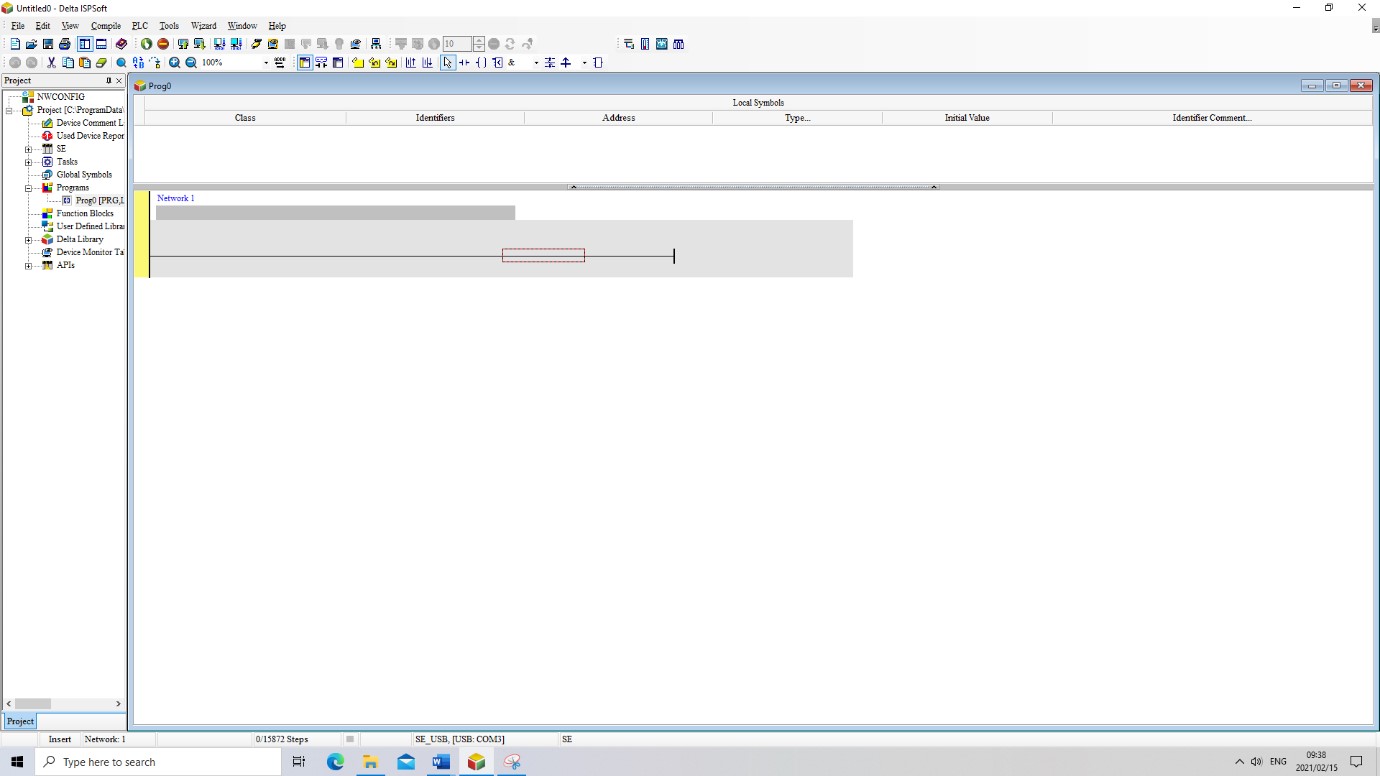


3

A create program pop-up window will appear. Here the POU name can be edited. Select Cyclic task and Ladder Diagram (LD), then click ok.



The user can now see their new main program appear on the main screen. Here they are started off with Network 1.



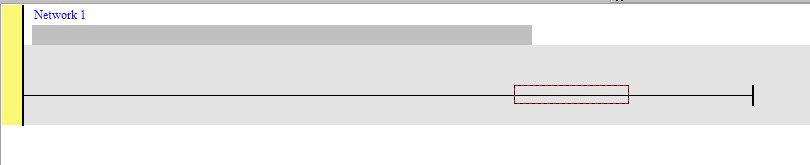
We can now begin to edit the program code for the PLC. The programme language used here is a Ladder Diagram.

To create a new network you can click either 

A blank network looks as so,

To get started a contact must be placed. Click on the contact icon on the toolbar (

)



and then drag your mouse over the red box after which the cursor icon will change. Left-click to place the contact.

Once done the red box will shift to the left and a contact with ??? above it will appear.



(Click the select icon in the toolbar  to avoid any accidental or unwanted clicks).

Click on the question marks to assign device address to it. You can choose any address you’d like, but for now assign the external input pin X0 to it like so.

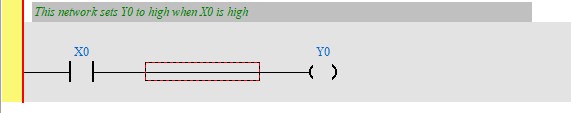


Now that we have an input we can connect it to an output coil (  ).

As you did with the contact, select the coil icon from the toolbar, move your cursor to the red box and then left click



We will also assign this a device address, bu this time we will use the external output pin Y0



The final network should look as above. The gray bars just below the network names can be used to make comments on the ladder diagram. It is recommended you make use of them so that you can keep track of what each network does.

This network will pull the external output pin Y0 high when X0 goes high.

We still need to compile the program and upload it to the PLC. Before a program can be loaded,

however, the PLC must be stopped. This can be done two ways, using  on ISPsoft workspace or by using the physical switch which can be found on the PLC itself\*. If this has worked properly the Run light (located under the power light of the PLC) will be off.

\*sometimes ISPsoft has a communication error with the PLC and can’t stop the PLC via the virtual button. In this scenario the physical switch will still work and should be used (located under the Run indicator light).

When the PLC has been stopped from running the program can be checked and compiled using respectively.

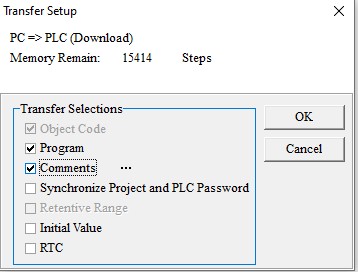


It is recommended you first check  and then compile .

If no warnings are thrown up on the **Compile Message** (window found at the bottom of your workspace) you can proceed with downloading your program to the PLC. This is done by clicking



A window will appear. Check the **Program** box and **Comment** Box, and click **OK.**



A warning message may appear. If this happens, close the dialogue box, check and compile the program again (as per the instructions above). Download the program again, but this time, uncheck the **Program** box and **Comment** Box, and then check them again. The program should download thereafter. Repeat this process if the program does not download.

If no warnings are thrown up then the download was succesful. If you haven’t stopped the PLC before downloading, a warning window will appear to remind you to stop it first.

Now that the program is on the PLC memory you can Run it again using  or the physical switch on the machine\*. The Run indicator light will be on

\*sometimes ISPsoft has a communication error with the PLC and can’t run the PLC via the virtual button. In this scenario the physical switch will still work and should be used (located under the Run indicator light).

The network we have just created makes use of the green button which can be found under the HMI screen labelled X0. When the button is pressed Y0 will be activated.

When your program is running you can use ISPsoft to observe online what the circuitry of the PLC is doing in real-time by using .

### A little aside

The PLC makes use of different device types, each with their own range of addresses. For example X0-X7 are the addreses of the external inputs of the PLC. X0 and X1 correspond to the red and green buttons, while Y0-Y3 are the adresesses of the external output pins. All the other device addresses can only be accessed through the programs loaded into the PLC or HMI. M and D will be the devices you will likely be using the most of. M is a bit device and each address can only store a bit value. D is a word device. A more in depth table can be found in the appendix at the end of this document.

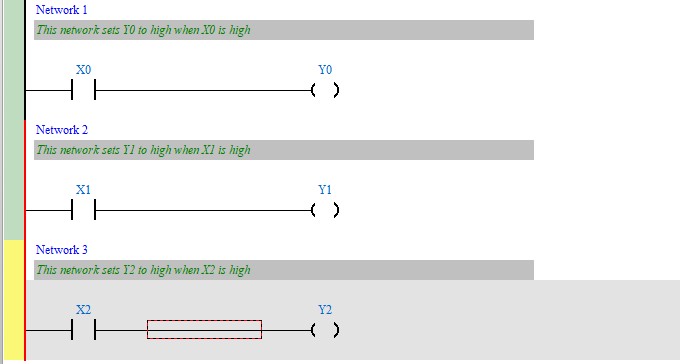
Each device address can be read from and written to by both the PLC and HMI. This is the main way to communicate information between them.

The final example we will be going through will show how the PLC interacts with the HMI. Using the same ISPsoft program you have created, make two more networks using the new network button

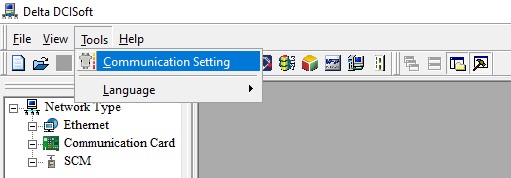


Add the necessary components into network 2 and





Once you’ve saved, compiled and downloaded everything to the PLC, you have to establish a connection between the PLC and HMI. To do this, open DCISoft (version 1.12). Once DCISoft is open, you’ll see the following screen.

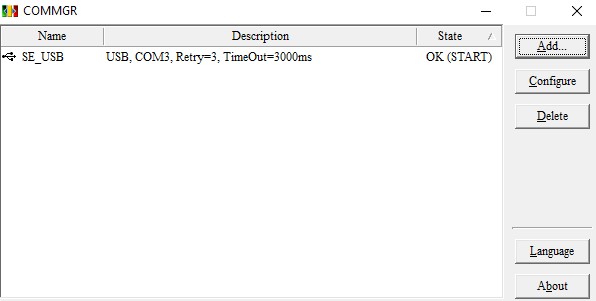
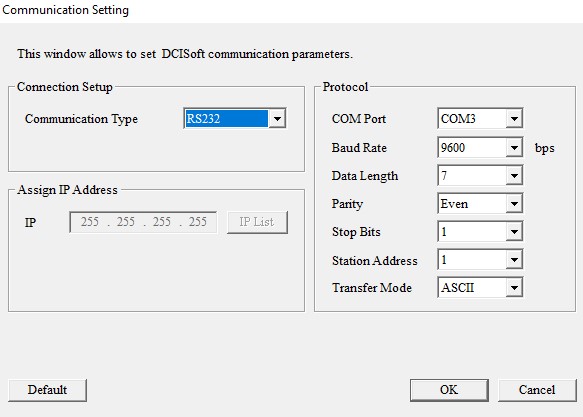


Navigate to the Tools menu as shown and click Communication Setting. Make sure the

Communication Type is set to RS232. Go to the Search Bar (at the bottom left of the screen). Type in

“COMMGR” and open. Note no window will open, but an icon will appear in the bottom right of the ). Double click the icon to bring up a menu.

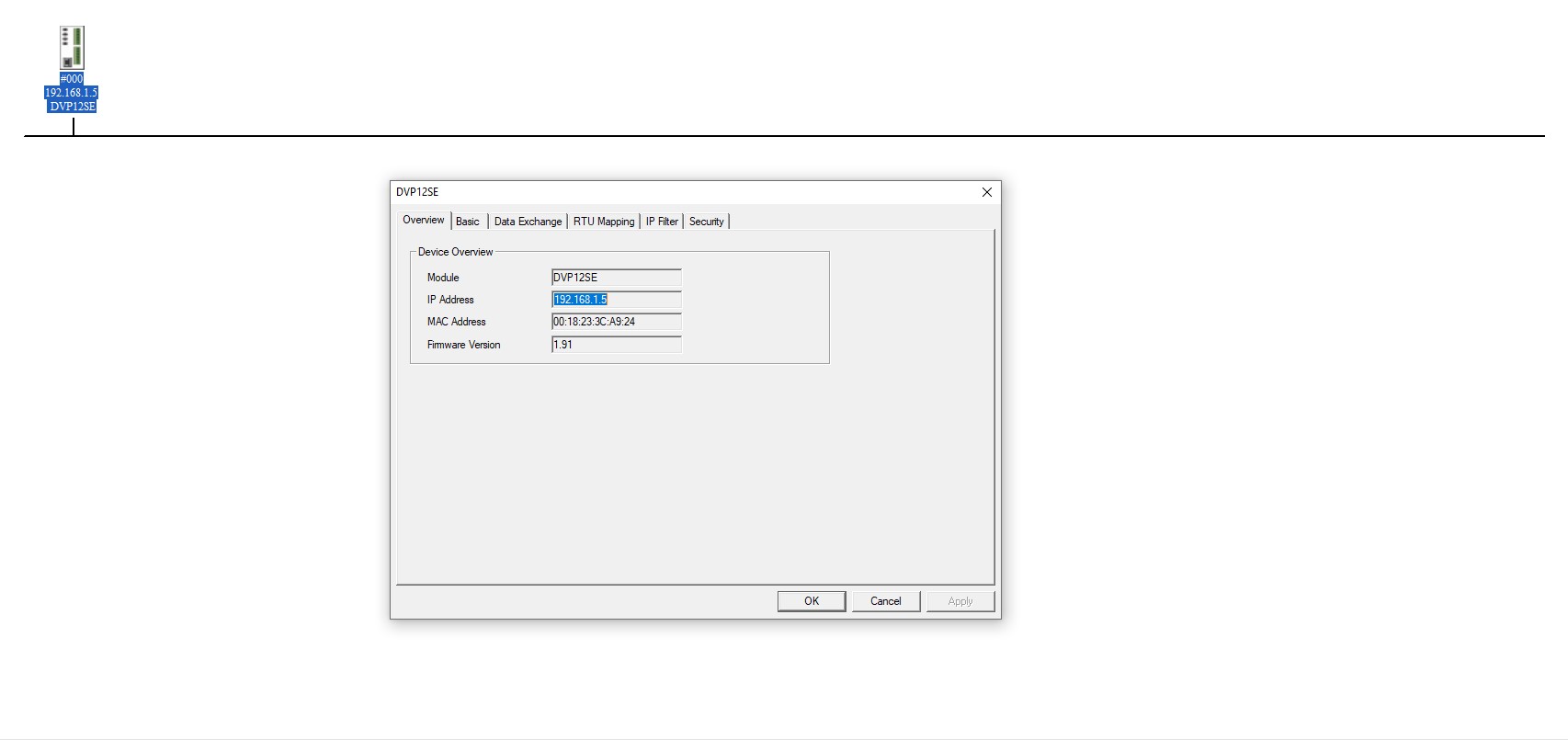
screen (



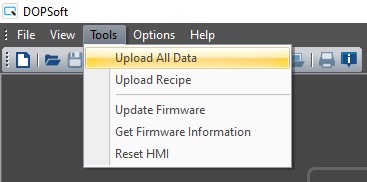
Make sure the COM Port in the Communication Setting window is the same as the COM Port set for SE\_USB in the COMMGR window. After that, click OK. Next click the Search button (  ). Thereafter you should see something like the image below.



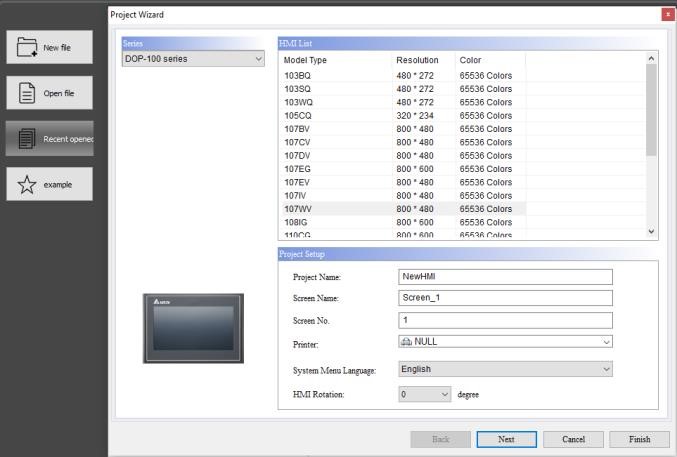
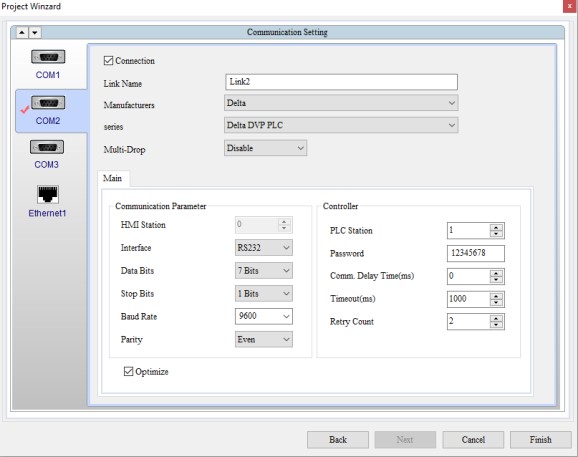
Double click and a window will appear. Make sure the IP address reads “192.168.1.5”. Then click the Basic tab. Make sure Module Name is set to “DVP12SE” and that the IP address is the same as the above, and Gateway must read “192.168.1.1”.



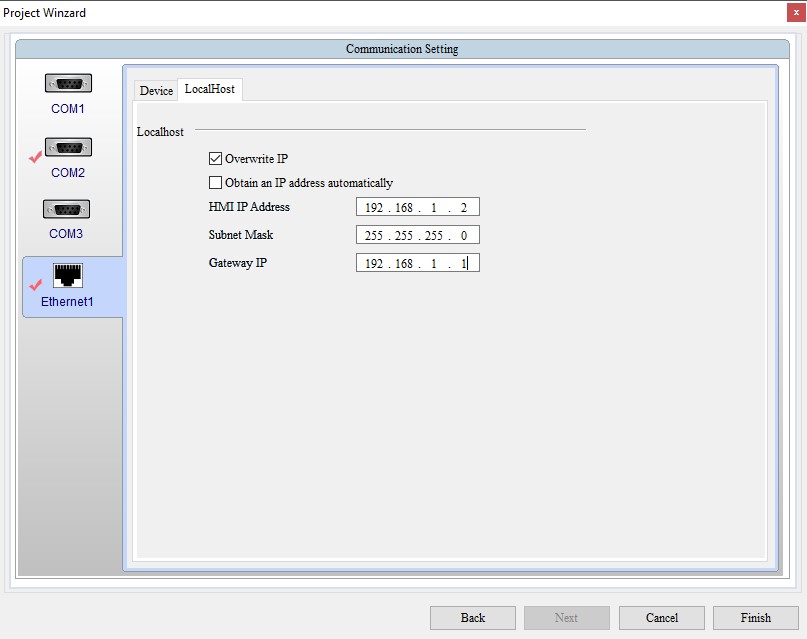
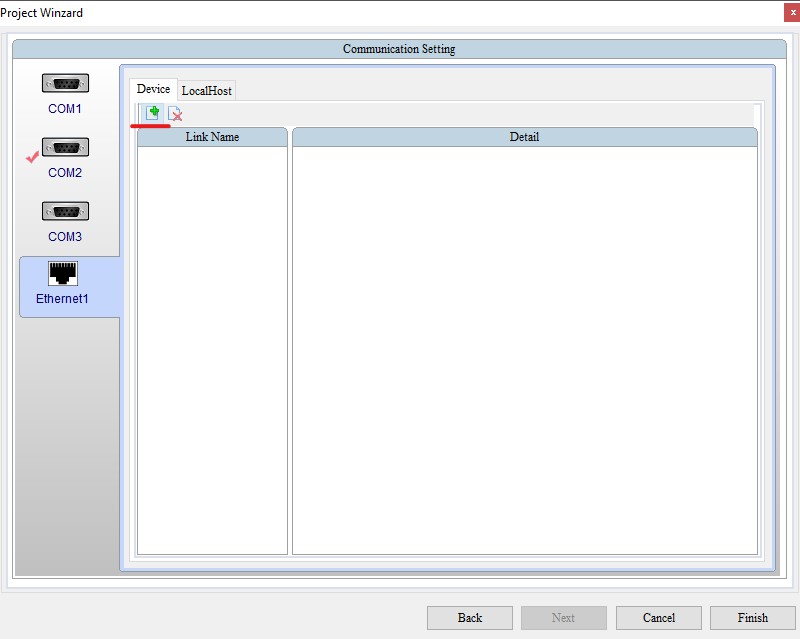
Once the connection has been established, open up DOPSoft (you’ll either have version 2.00.40 or version 4.00.10 depending on the HMI model you are using). As mentioned before, it is important to make sure the HMI is properly connected to the PC, as well as taking note of the model you are using and updating the firmware. Once you open up DOPSoft, go to the tools in the top left corner, and there will be 2 options in the drop-down menu that say “Reset HMI” and “Update Firmware”. First, select Reset HMI. This will factory reset the screen and change it to its default state (this screen will have buttons which will allow you to check the system settings and therefore the model name). The HMI’s in the lab will be Model Type: DOP-B07E415 or DOP-107EV (but you are advised to go and confirm this for yourself by navigating to system settings found on the HMI’s default screen). The DOP-B07E415 HMI will run version 2.00.40 of DOPSoft while the DOP-107EV HMI will run version 4.00.10, although both versions feature very minor differences, the main procedure for compiling programs should still be the same.



Next, click New File and select the model of HMI you are using (DOP-107EV in this case).

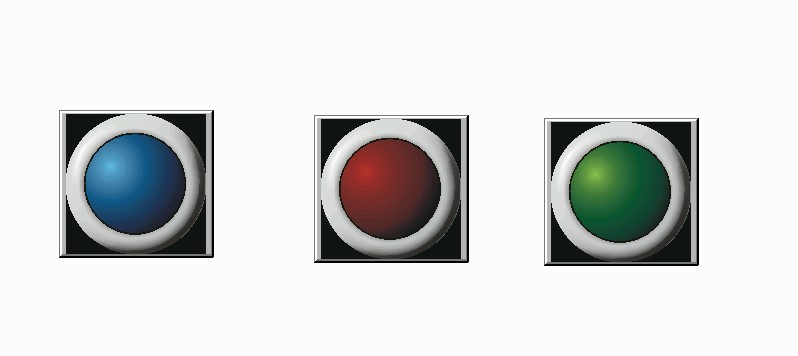


Click Next and then select Ethernet, and click Add Device. Change Link name to “DVP\_SE” and set Controller IP: Port to “192.168.1.5” (the IP address that was set in DCISoft). Next click Local Host. Check the “Overwrite IP” box and uncheck the “Obtain IP address automatically” box. Set the HMI IP address to “192.168.1.2” and Gateway IP to “192.168.1.1”. Leave the Subnet Mask as is.



Now click COM 2 and change the Link Name to “DVP12SE11R” and change the Interface to RS485. Leave everything else as is. Click the Tools button again in the toolbar, and click Update Firmware to ensure the firmware is compatible with DOPSoft. The update will begin shortly. You may encounter an error message that will read “No Response From HMI”, “

Assuming there are no errors and you were able to load up DOPSoft and interact with the HMI successfully, create an HMI layout similar to this one shown below,

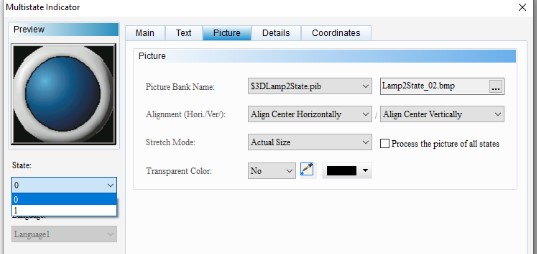


To create each button :

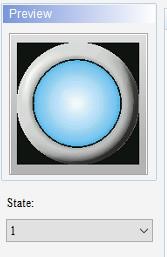
Right click in the blank space, select **Indicator** , **Multi State Indicator**

Each Button Will have two states, one where the button is not being pressed which is considered State 0 and the second one where the button is being pressed the light is on.

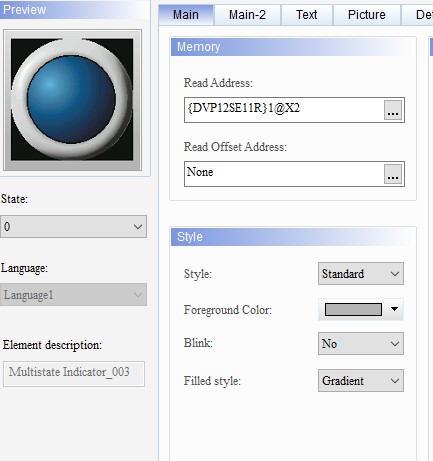
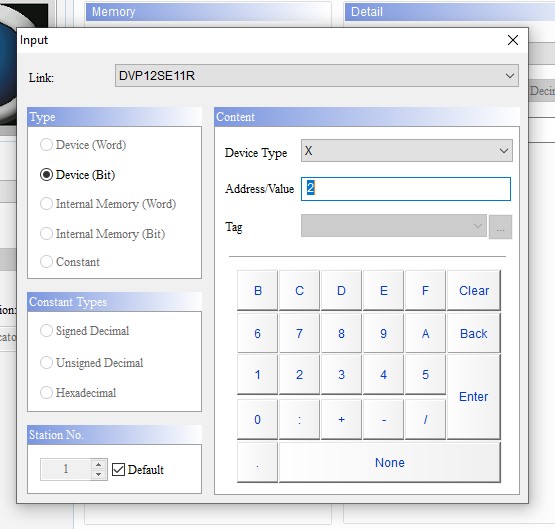
Go to the Picture tab on the multistate indicator settings. Make sure that the **State** selected is 0 and then go to the picture bank name drop list and select **$3DLamp2State.pib**



Change the State to 1, then to Indicate that the button has been pressed, a lit-up indicator must be selected. For each state there must be 1 image, 1 dark and 1 bright. The dark image represents state 0 while the bright image represents state 1. The images below show the three lit up (bright) indicators.



For each indicator we will assign a read address of X0, X1 and X2 corresponding to the labels on the HMI for each button respectively (for example Multistate Indicator\_001 corresponds to X0).

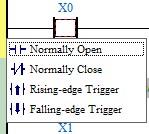
 

Compile the HMI program by clicking Tools in the toolbar and selecting Compile. Now download the program to the HMI by clicking Download Screen, again under the Tools menu.

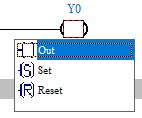
Now you can interact with the buttons and see how the HMI responds.

While the HMI DOPsoft tool is very user friendly and obvious to understand, the PLC’s ISPsoft has a lot more tools to offer that weren’t explored in the examples.

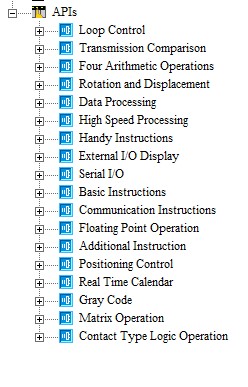
Each contact when double clicked has options for when to register a high signal

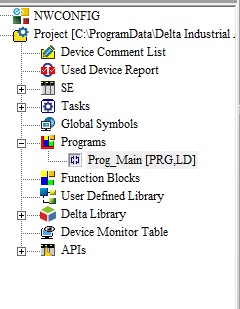


Similarly, each coil can have a predefined condition for what state to be in when activated.



Furthermore, in your workspace there are API’s that can be used for constructing networks with more complexity.





There is a large assortment of API’s to use and not enough time in this text to explain them. If you wish to learn what is available feel free to use the ISPsoft User Index (or press F1 key while an element is selected).

# 3. Practical

Main objectives of the practical:

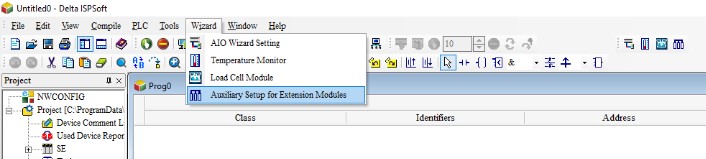
1. Program the PLC to output an analogue reference voltage to the DAC and accept input voltages.
2. Create an interface on the HMI to change said output voltage from any value between -10v to +10v.
3. Create a digital controller using Visual Studio 2019.

## 1. Setting up the PLC

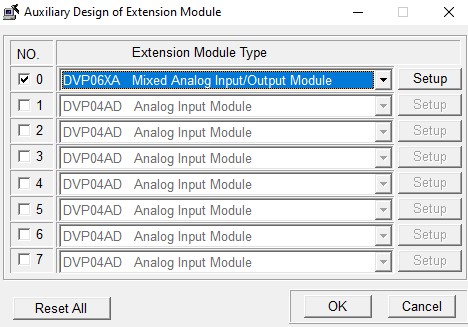
The PLC itself can only output, and accept, Logic states of 0 and 1. For this practical we wish to output a range of voltages from -10v to +10v. We would also like to accept a range of voltage from -10v to +10v. For this reason, an extension module has been attached to the side of the PLC. This is the DVP06XA Analogue Input/Output Converter.

Before we can begin to use it, we must configure the Control Registers (CR) of the Analogue I/O device.

Go to the top of the toolbar and select **Wizard >** **Auxiliary Setup for Extension Modules**.

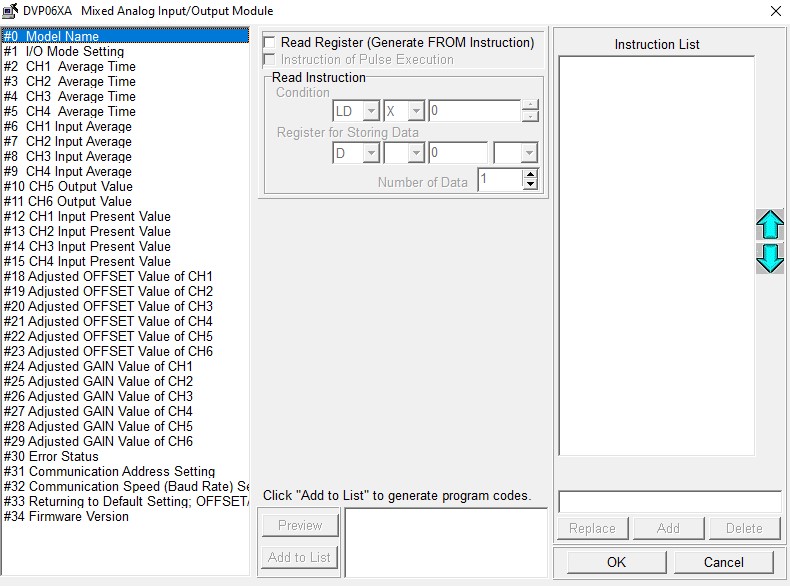


A window will pop-up where you can select the model of the extension module you would like to setup.



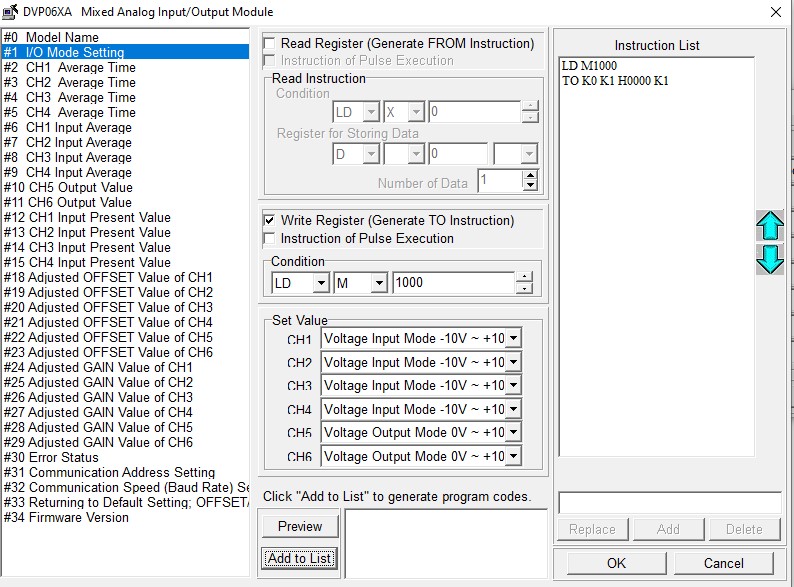
Click the No.0 box and from the drop-down list select **DVP06XA** and click **Setup**.

A New window will appear showing all 34 Control Registers (CR) which can be used to configure the settings of the module.



Go to the **#1 I/O Mode setting**. Check the Write Register box. Ensure that CH1-CH6 are set to the same modes that can be seen in the below image.

The condition section can be filled out using the values as shown in the image. **LD>M>1000**. This will be explained after setup is complete.

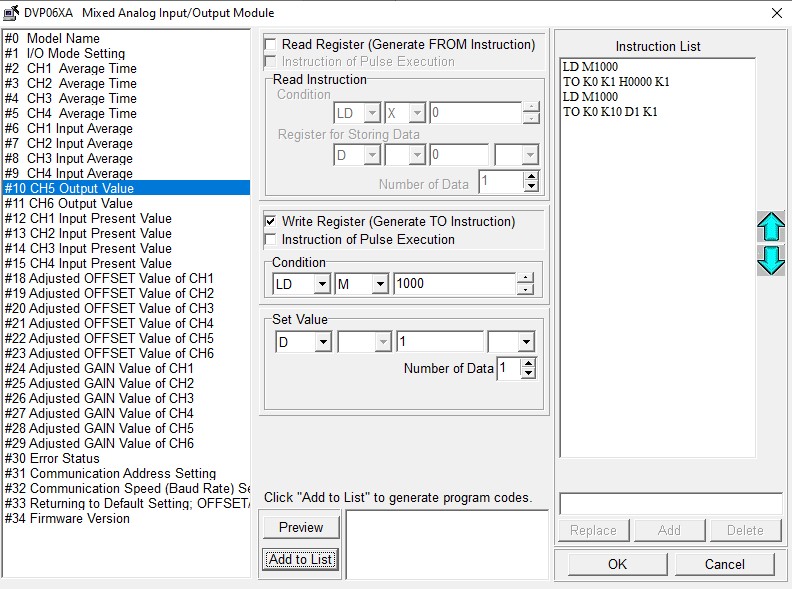


NB CH5 SHOULD BE ON -10 TO 10

Click **Add to List**. This list will generate the networks needed to edit the CR. Before we click OK two more CR needs to be configured.

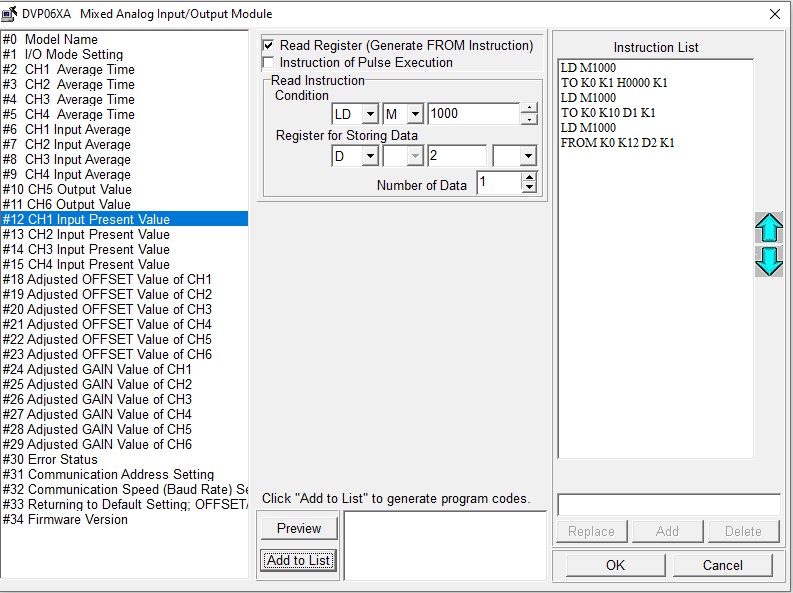
For this practical CH1 will be used for voltage input and CH5 for voltage output. The device addresses into which they read and write from need to be selected.

Go to the CR register **#10 CH5 Output Value**. Check the Write Register box and set the condition to **LD>M>1000** and Set value section to **D>1**.

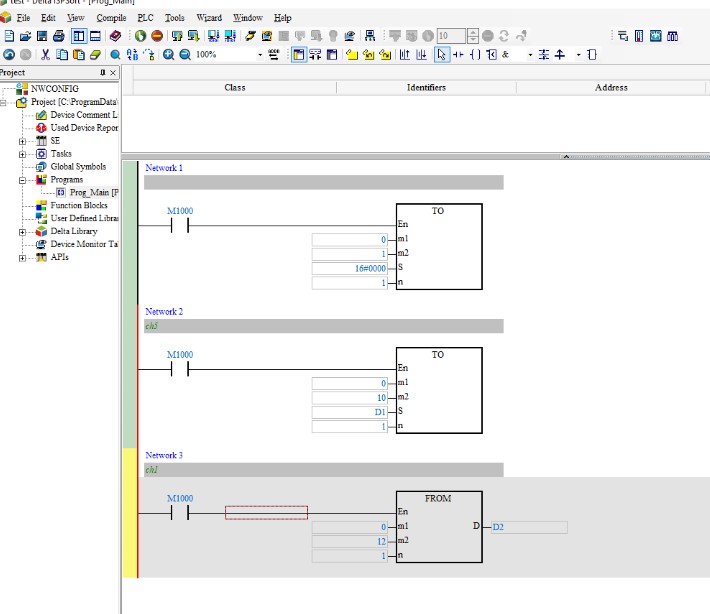


**D1** is now the address which sets the output voltage value of CH5. A D register was chosen becase it is a word register and we need it to hold any value between 0 - 4000. Click **Add to List.**

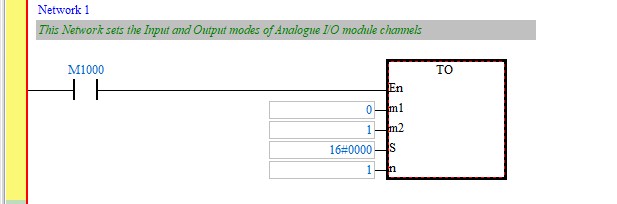
Next go to CR **#12 CH1 Input Present Value**. Check the Read Register box, set the condition to **LD>M>1000**, and the register for storing data to **D>2**. **Add to List**.



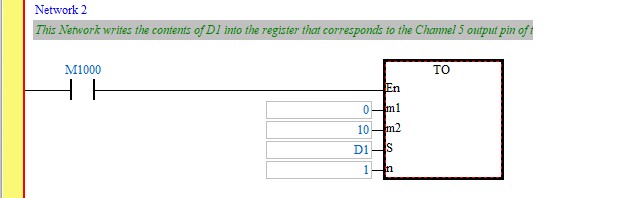
Once this is done click **OK**, and then **OK** again on the Wizard window. Your main program will now have 3 new networks. Each corresponding to the CR configurations that you made earlier.



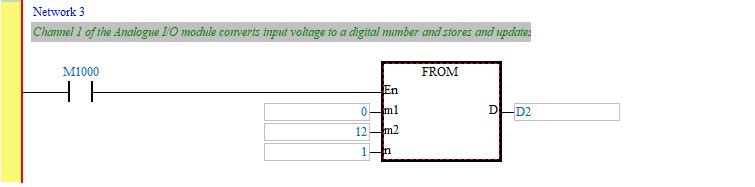
Network 1’s function is illustrated by its comment,



Network 2 writes the contents of **D1** into the register that corresponds to the Channel 5 output pin of the Analogue I/O module. This acts a digital to analogue converter which can take in values from between **0-4000**

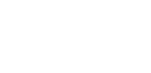


Network 3: Channel 1 of the Analogue I/O module converts input voltage to a digital number and stores and updates the value in **D2.**



In Each network **M1000** is used as it has a constant state of 1, ensuring that the API block’s (TO/FROM) are constantly enabled while the PLC is running.

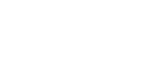
Now that we have set up the digital to anlogue conversion channel (CH5) and analogue to digital conversion channel (CH1) we can start outputting voltage values by changing the integer value held in **D1.** The illustrations below show the A/D and D/A conversion characteristic graph of CH1 and Ch5.



-

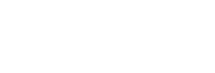
10

V

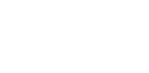


10

V



+2000

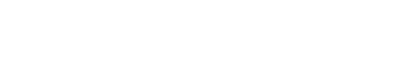


-

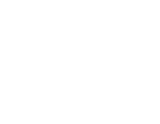
2000



0

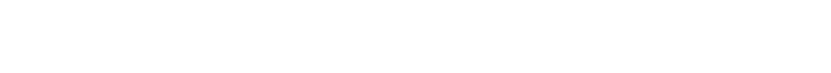


Digital Output

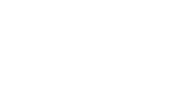


Voltage

Input



**A/D Conversion Characteristic Graph of CH1~CH4**



Digital

Input



Voltage Output



0



0



V

10



-

10

V



+2000



+4000

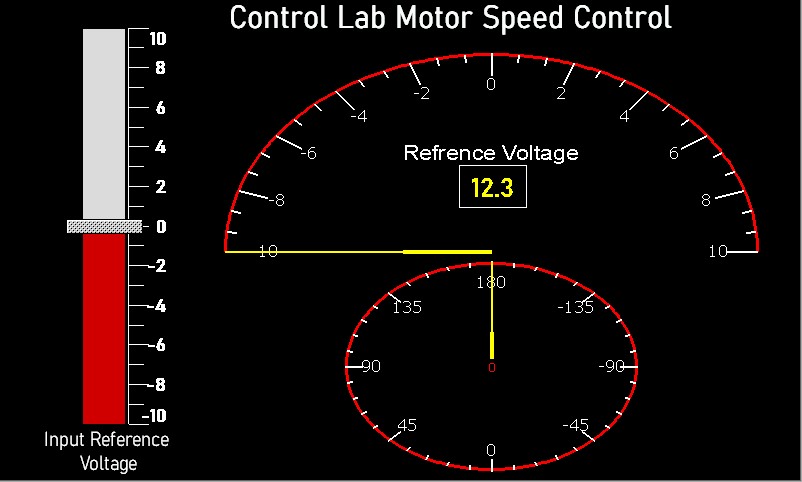


**D/A Conversion Characteristic Graph of CH5~CH6**

## 2. Creating a Usable HMI

Now that we have set up the PLC to have output voltages that can be varied and input voltages that can be read, we want to create an HMI screen that allows us to fully control said output voltage and read the input voltage.

This is where you can show your creativity and make whatever interface you think best suits the practical. As an example here is an interface that has been created:



How it looks is entirely up to you the only requirments is that it must have a dial/slider/keypad or any other input element that allows us to change the output voltage of the PLC.

It is likely that you will have to create more Networks on the PLC programme in order to achieve this.

Hint: The D/A convertor graph is very useful for the design of the HMI screen. You will have to use various operations (such as the basic mathematical operations like SUB, ADD, DIV, MUL) on ISPSoft to convert your values from a digital input on the slider to a voltage output on the meter, and vice versa (converting an analogue input to a digital output).

## 3. Creating a digital controller inVisual Studio

You would need to use Visual Studio (the 2019 version) to create your controller. Just like with the HMI, the basic elements should be the same but the design is completely to your discretion.

**4. Table Of Device Types And Their Respective Addresses**

