

Practical Manual

EEE4118F

INTERFACING THE DELTA DVP-PLC WITH AN HMI SCREEN



Authors:

Dino Kingsley-Rizzo

John Mkwebo

Katerina Itopoulos

Nicolen Pillay

Luke Baatjes

Dominic de Maar

Table of Contents

1. Pre-Reading	3
1.1 Delta DVP-PLC	3
1.1.1 Hardware	3
1.1.2 Software	6
2. Pre-Practical	8
2.1 ISPSOft.....	8
2.2 DCISOft	14
2.3 DOPSOft	16
2.3.1 DOP-B07E415 HMI (Older Model No White Line Along Bezel).....	16
2.3.2 DOP-107EV HMI (Newer Model with White Line Along Bezel)	19
3. Practical	23
3.1 Setting up the PLC	23
3.2 Create a Functional HMI Screen	31
3.3 Design a Digital Controller and Implement using Visual Studio.....	31
4. Table of Device Types and their Respective Addresses	32

1. Pre-Reading

Get to Know the Delta DVP Series

1.1 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](#)
- [DVP-06XA](#)
- [DVP-PS01](#)
- [DOP-B07E415](#)

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

- [ISPSOft](#)
- [DOPSOft](#)
- [DCISOft](#)

1.1.1 Hardware

DELTA 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



Figure 1-DVP12SE PLC

DELTA DVP-PS01

The DVP-PS01 is a Delta DIN rail power supply. A DIN power supply is a switched mode power supply that converts unstable input voltage to regulated output voltage.

Features:

- Nominal output voltage of 24V
- Temperature ranges 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)



Figure 2-DVP-PS01 Power Supply

Delta DVP-06XA

The DVP06XA-S extension module allows the connection of four analogue inputs and 2 groups 12 bits digital outputs (voltage/current). The PLC converts the input into a 12-bit digital signal and the output into a 2 points analogue signal, which then are manipulated by using TO and FROM commands in the ladder logic program. There are 49 Controlled Registers (CR, each register has 16-bit) in each module.

Features:

- 24 VDC Supply
- 4 Analogue Inputs
- 2 Analogue Outputs
- Built-in RS485



Figure 3-DVP-06XA Analogue I/O extension module

HMI Model Variants Available in the Lab

There are two Delta HMI Models available for use in the Lab. It is important to note which model you are working with as this will dictate which DOPSoft version you will be using. The DOP-B07E415 model does not have a white line along the bezel of its display while the DOP-107EV model has the white line.

DOP-B07E415

The DOP-B07E415 is a touch screen HMI.

Features:

- Three COM ports that support RS-232, RS-422 and RS-485
- Data can be transmitted or downloaded through RS232 (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.



Figure 4-DOP-B07E415 HMI

Note: Cannot use USB communication port as master

DOP-107EV

The DOP-107EV is also a touch screen HMI and its features are like that of the B07E415.

Notice the white line along the bezel! to help discern the model variant.



Figure 5-DOP-107EV HMI

Communication Parameters

Table 1-HMI 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

1.1.2 Software

ISPSoft

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

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

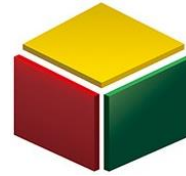


Figure 6-ISPSoft

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. In this lab, the version of DOPSoft you will be using depends on the model of the HMI you are using. If you are using the DOP-B07E415 HMI, then you will use DOPSoft version 2.00.4. **Do not** use version 2.00.07 or version 4.00.10 as they will not work with this HMI model. If you are using the DOP-107EV HMI, then you will use DOPSoft version 4.00.10.

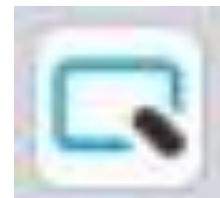


Figure 7-DOPSoft

DCISoft

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

Flowchart for creating a project on DOPSoft

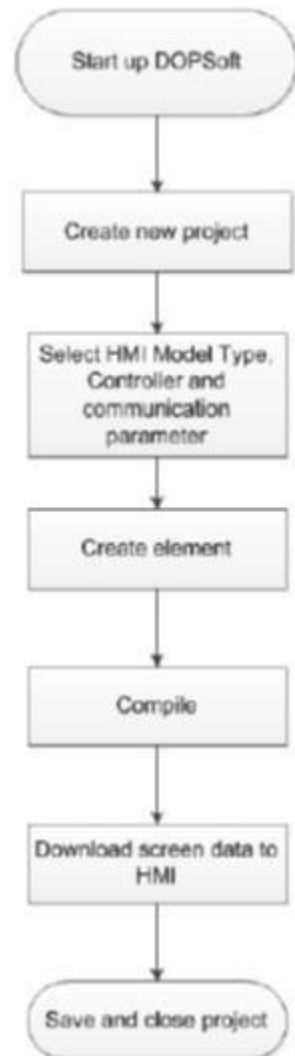


Figure 8-Flowchart for Creating Project in DOPSoft

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 programme.
- The Delta HMIs are programmed using the DOPSoft programme.

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 (remove any cables and re-insert just to make sure but be sure to remember where each cable goes). It is also very **important** to take note of the **HMI model name and number** you are using. For example, The DOP-B07E415 HMI is an older model which will use DOPSoft version 2.00.04 while the newer DOP-107EV HMI will use DOPSoft version 4.00.10. It is important that you are using the correct software version compatible with your HMI model. Further instructions will be given when creating screens for your HMI.

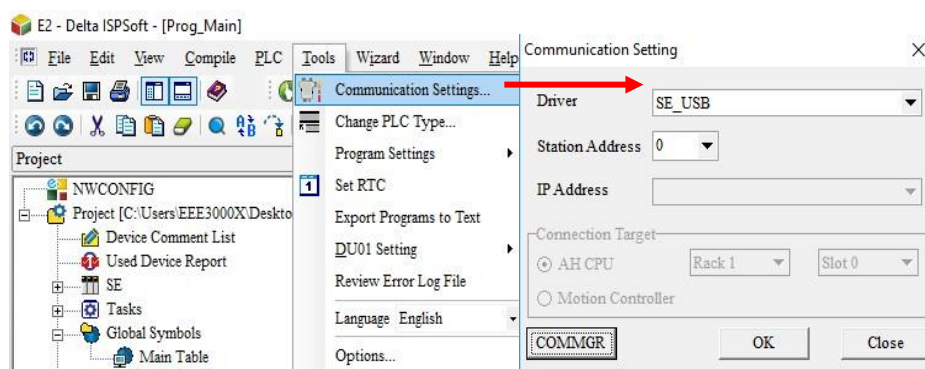
2.1 ISPSOFT

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

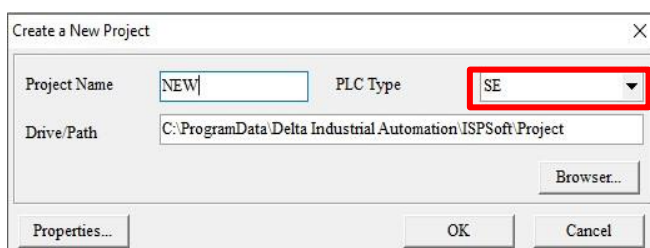
Before starting a new project ensure that the following communication settings have been configured.

Click on **Tools > Communication Settings**

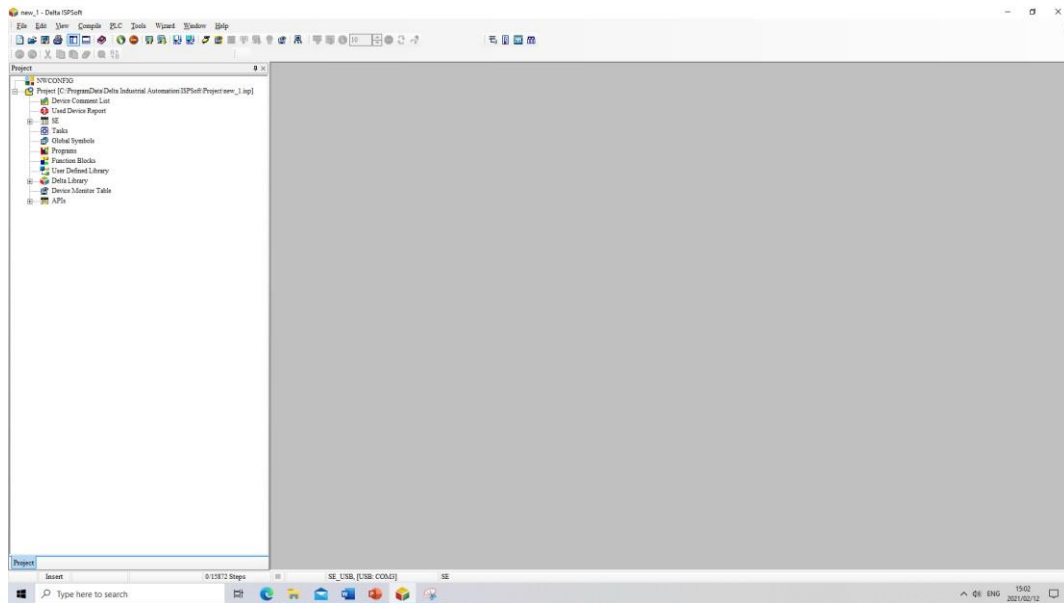
In the **Communication Settings window**, in the Driver list, select **SE_USB** from the drop-down list and click **OK**.



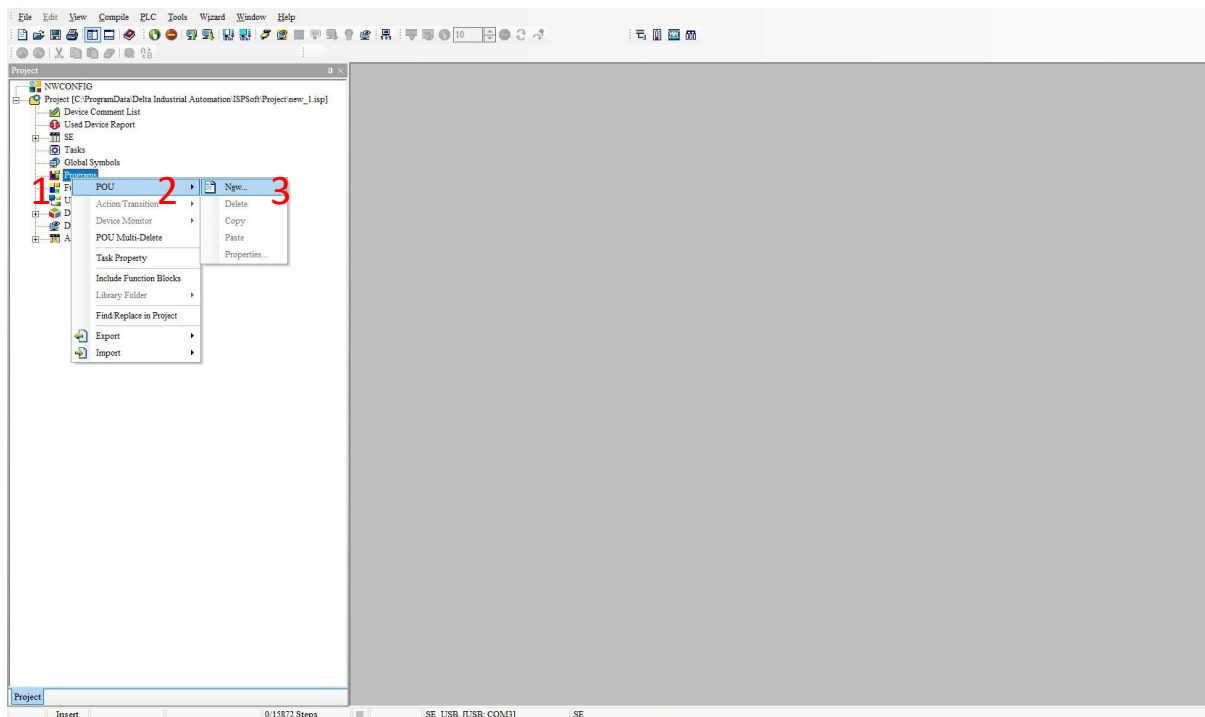
Click on  in the toolbar to add a new project. The following window will appear:



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.



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.

Create Program

POU Name:

Task:

☒ Active

Protection (4~12 Characters):

Enter Password:

Confirmation:

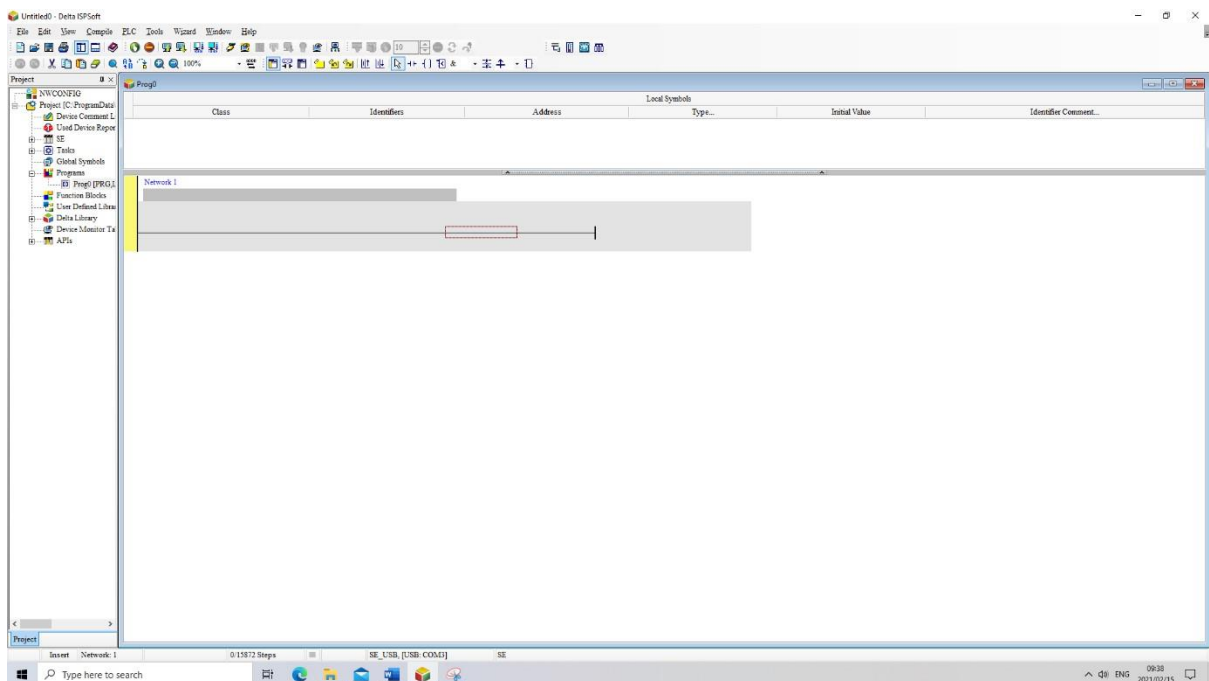
Language:

- ☒ Ladder Diagram (LD)
- ☐ Sequential Function Chart (SFC)
- ☐ Function Block Diagram (FBD)
- ☐ Instruction List (IL)
- ☐ Structure Text (ST)

POU Comment:

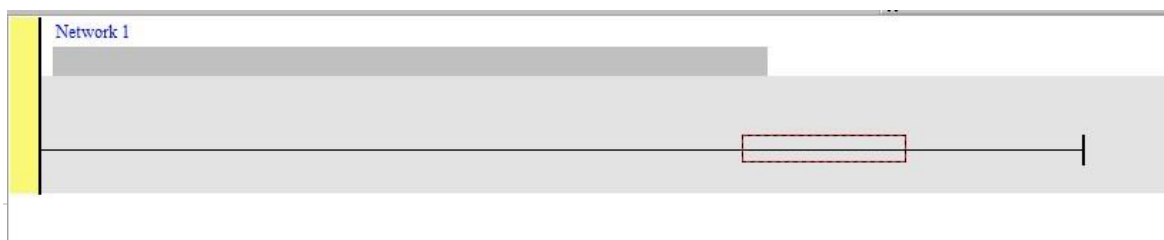
OK Cancel


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   to add a network above or below the current selected network. A blank network looks as follows:




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.



(**Tip:** 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.

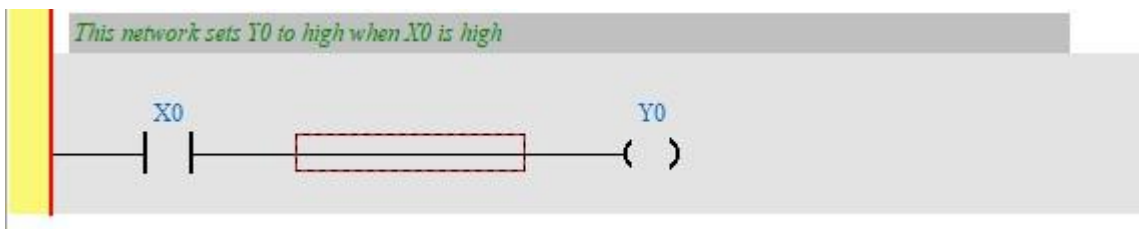


Now that we have an input, we can connect it to an output coil ().

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, but this time we will use the external output pin **Y0**.






The final network should look as above. The grey 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 you can simply wait till the program is ready to proceed and clicking **stop** again or using the physical switch on the PLC will work and should be used (located under the Run indicator light).

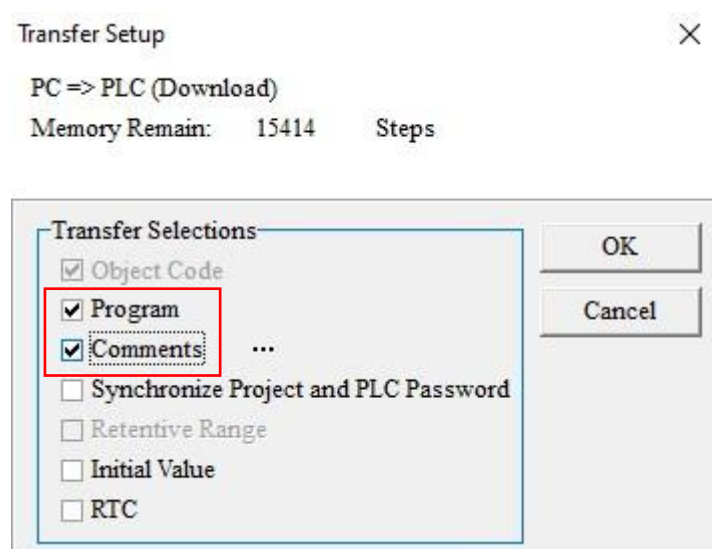
When the PLC has been stopped from running the program can be checked and compiled using  respectively.

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 successful. 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 you can simply wait till the program is ready to proceed and clicking **run** again or using the physical switch on the PLC will 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

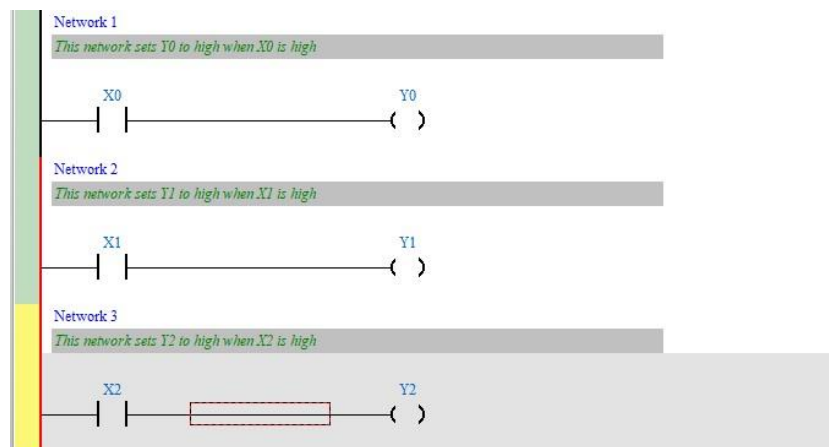
The PLC makes use of different device types, each with their own range of addresses. For example, X0-X7 are the address of the external inputs of the PLC. X0 and X1 correspond to the red and green buttons, while Y0-Y3 are the addresses of the external output pins. All the other device addresses can only 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



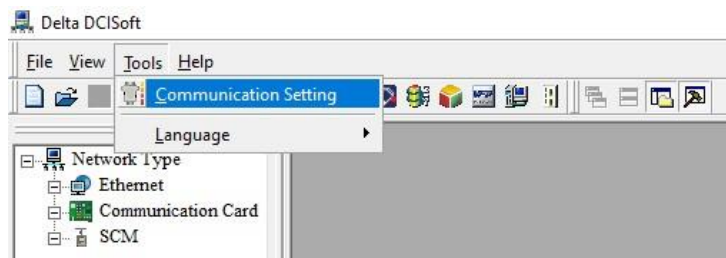
Add the necessary components into network 2 and its contact and coil,



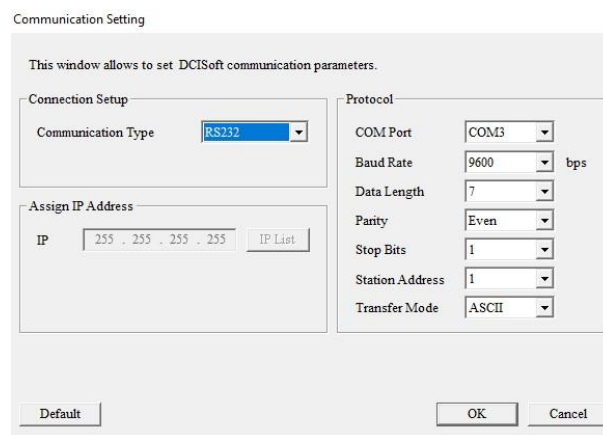
Once you've saved, compiled and downloaded everything to the PLC, you must establish a connection between the PLC and HMI.

2.2 DCISoft

Open DCISoft (version 1.12). Once DCISoft is open, you'll see the following screen:



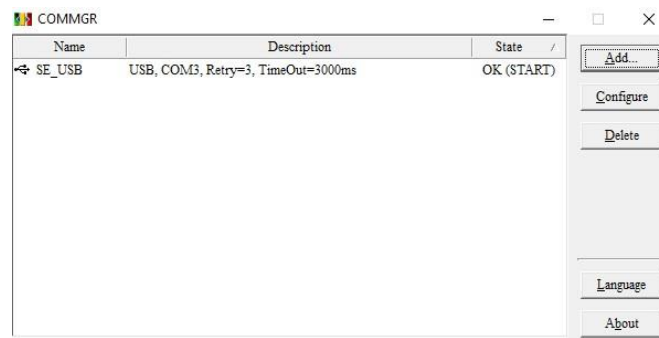
Navigate to the Tools menu as shown and click Communication Setting. Make sure the Communication Type is set to RS232.




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 screen ().

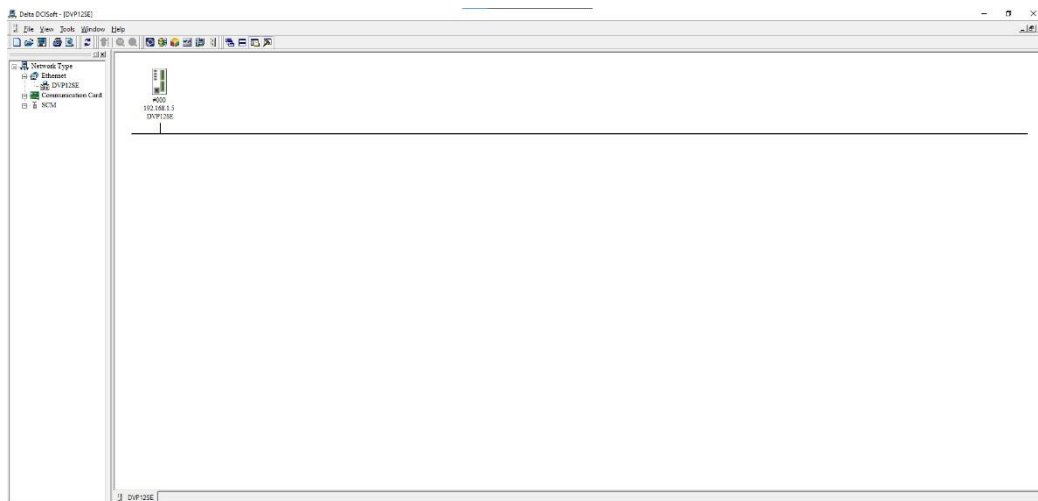


Double click the icon to bring up a menu:

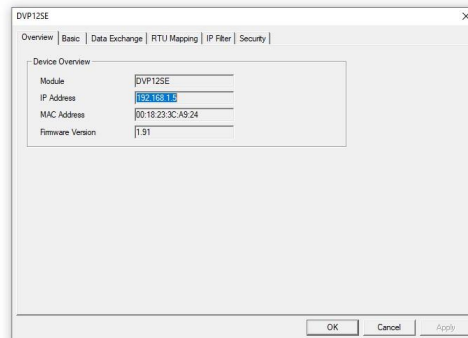


Make sure the COM Port in the Communication Setting window is the same as the COM Port set for SE_USB in the COMMGR window above. You may close the COMMGR window then click OK in the

Communication Setting Window. 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” (See next page).



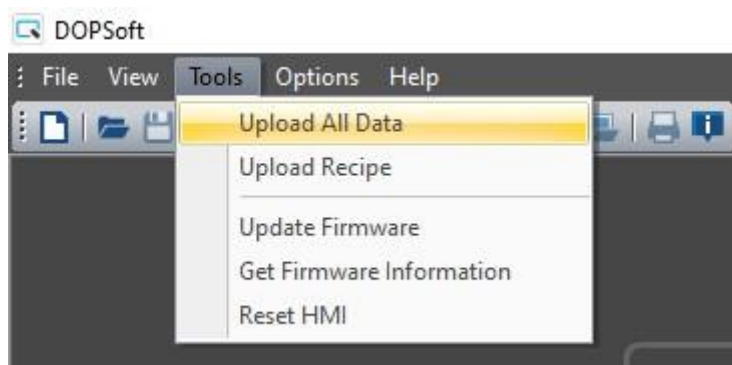
The connection has now been established and you may minimise the DCISoft window.

2.3 DOPSoft

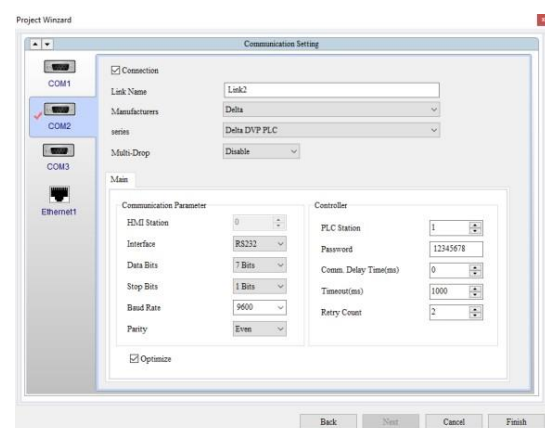
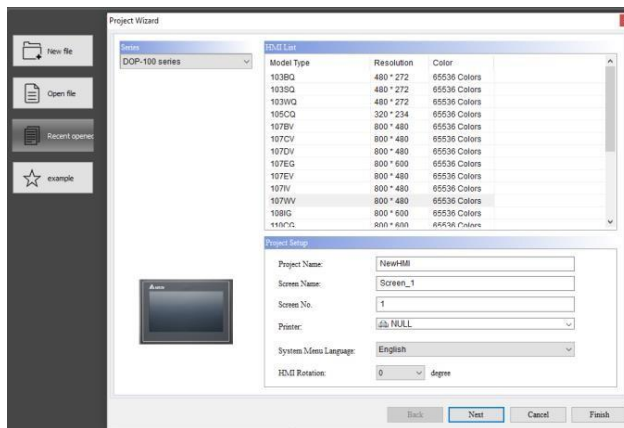
Once the connection has been established, open up DOPSoft (you'll either have version 2.00.04 or version 4.00.10 depending on the HMI model you are using). As mentioned before, it is important to make sure the HMI is properly connected to the PC, as well as taking note of the model you are using and updating the firmware (**only DOP107EV HMI**).

2.3.1 DOP-B07E415 HMI (Older Model No White Line Along Bezel)

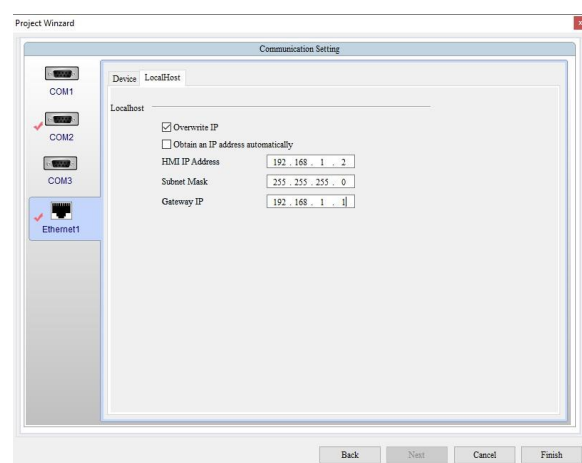
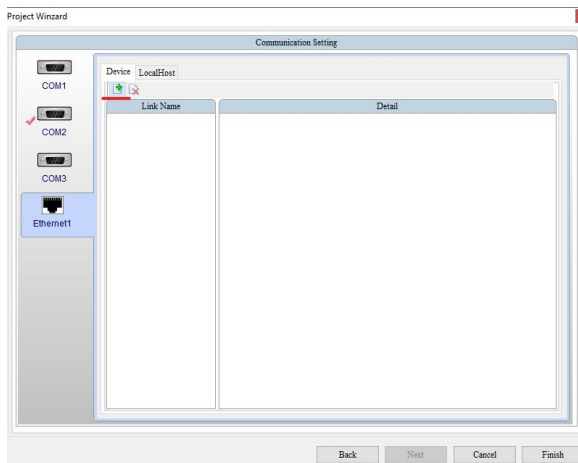
Once you open up **DOPSoft version 2.00.04**, go to the tools in the top left corner, and there will be an option in the drop-down menu that says, "Reset HMI". Click Reset HMI. This will factory reset the screen and change it to its default state (this screen will have buttons which will allow you to check the system settings and therefore the model's name).



Next, click New File and select the model of HMI you are using (DOP-B07E415 in this case):

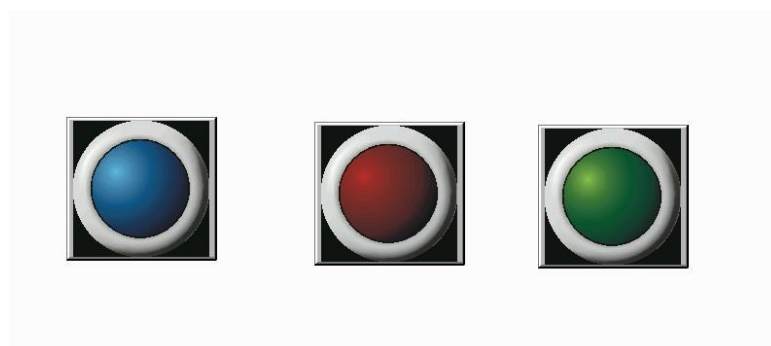


Click Next and then select Ethernet and click Add Device. Change Link name to “DVP_SE” and set Controller IP: Port to “192.168.1.5” (the IP address that was set in DCISoft). Next click Local Host. Check the “Overwrite IP” box and uncheck the “Obtain IP address automatically” box. Set the HMI IP address 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 and click “finished”. **Do not attempt to update firmware on this model.**

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

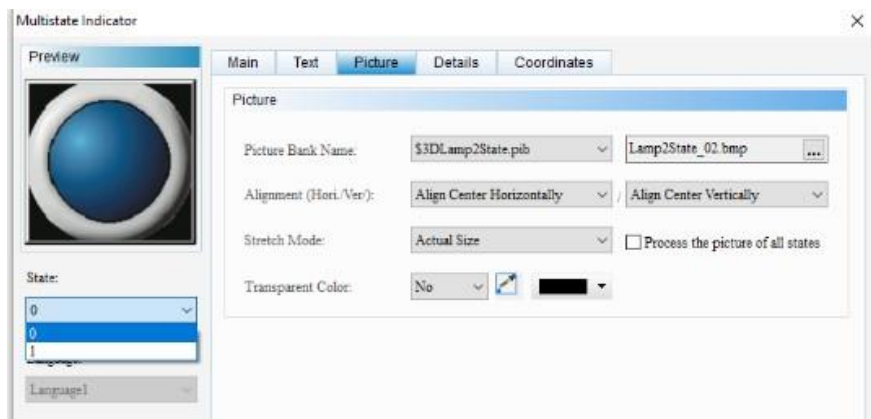


To create each button:

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 (**State 1**) where the button is being pressed the light is on.

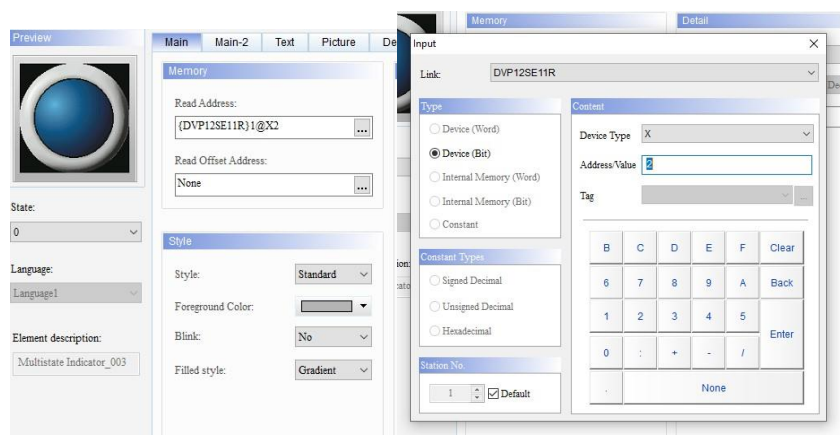
Go to the Picture tab on the multistate indicator settings. Make sure that the State selected is **0** and then go to the picture bank name drop list and select \$3DLamp2State.pib. Choose the darker looking lamp indicators for state 0 to signal that they are OFF!



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:



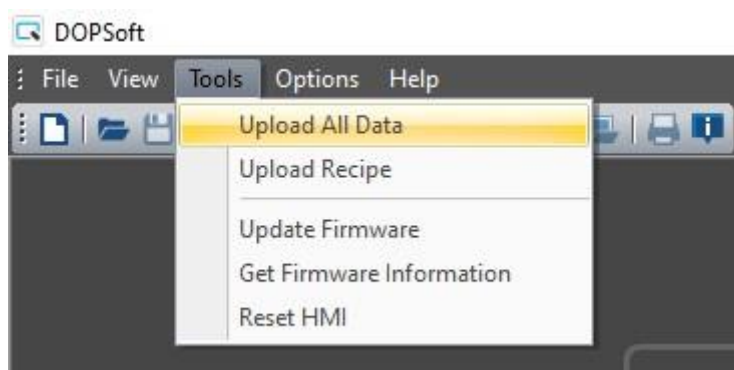
For each indicator we will assign a read address of X0, X1 and X2 corresponding to the networks created in ISPSOFT earlier and the labels on the HMI for each button respectively (for example Multistate Indicator_001 corresponds to X0):



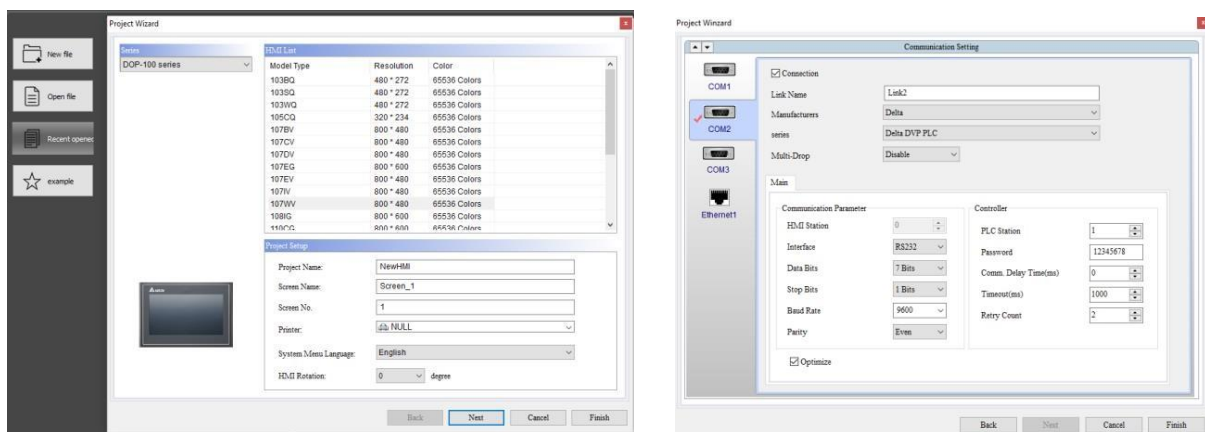
Compile the HMI program by clicking Tools in the toolbar and selecting Compile. Now download the program to the HMI by clicking Download Screen, again under the Tools menu. If you encounter an error, once again simply click Download Screen again and it should work. Now you can interact with the buttons and see how the HMI responds. To see a response, make sure the run light on the PLC is lit.

2.3.2 DOP-107EV HMI (Newer Model with White Line Along Bezel)

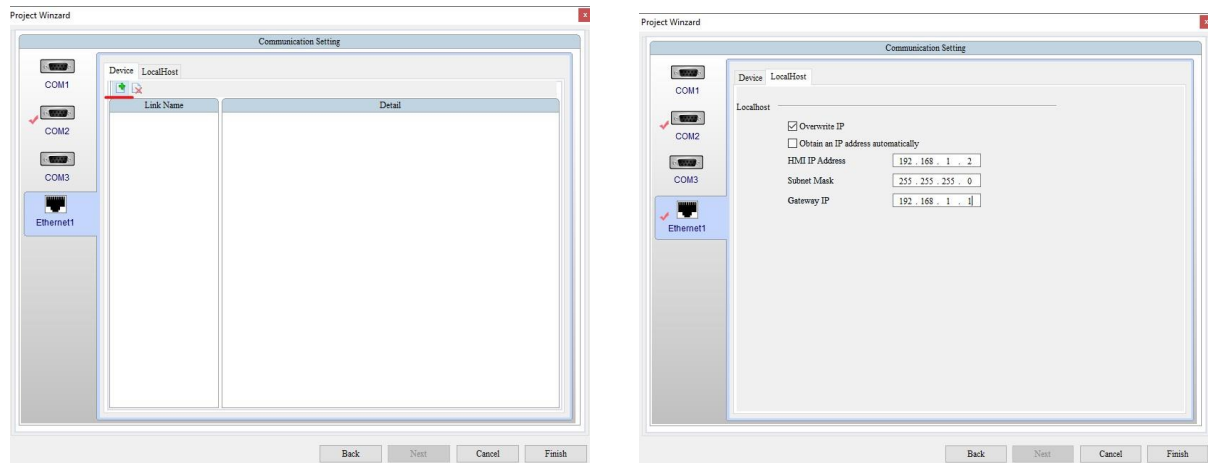
Once you open up **DOPSoft version 4.00.10**, go to the tools in the top left corner, and there will be 2 options in the drop-down menu that say, “Reset HMI” and “Update Firmware”. First, select Reset HMI. This will factory reset the screen and change it to its default state (this screen will have buttons which will allow you to check the system settings and therefore the model’s name).



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

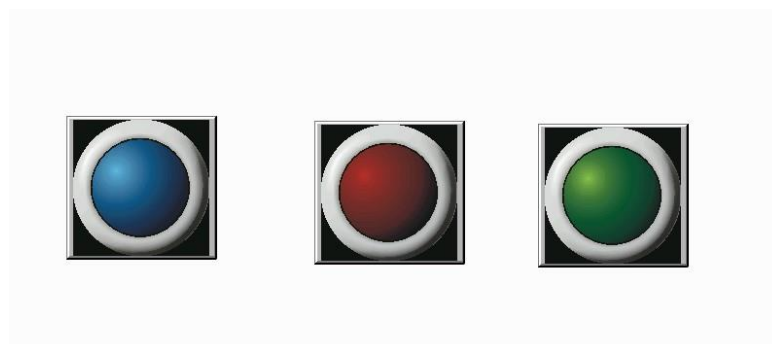


Click Next and then select Ethernet and click Add Device. Change Link name to “DVP_SE” and set Controller IP: Port to “192.168.1.5” (the IP address that was set in DCISoft). Next click Local Host. Check the “Overwrite IP” box and uncheck the “Obtain IP address automatically” box. Set the HMI IP address 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”. In this case you can simply click update firmware again and it should work. Alternatively, navigate to “Options”, then click on “Change model” and change the HMI type to “DOP-107EV 65535 Colours”.

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

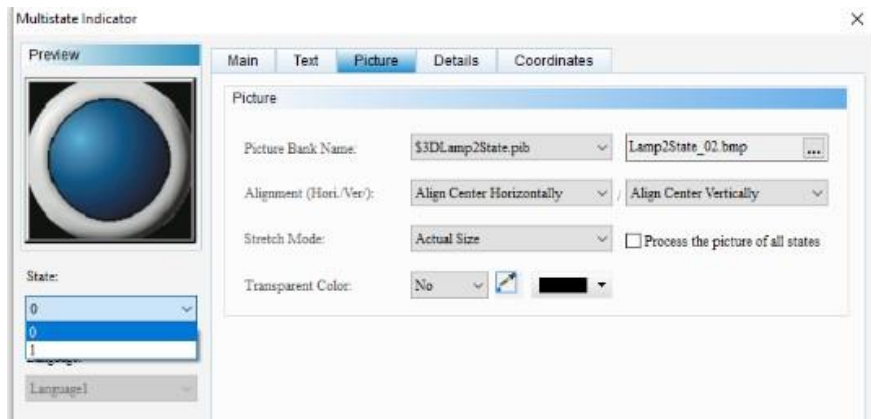


To create each button:

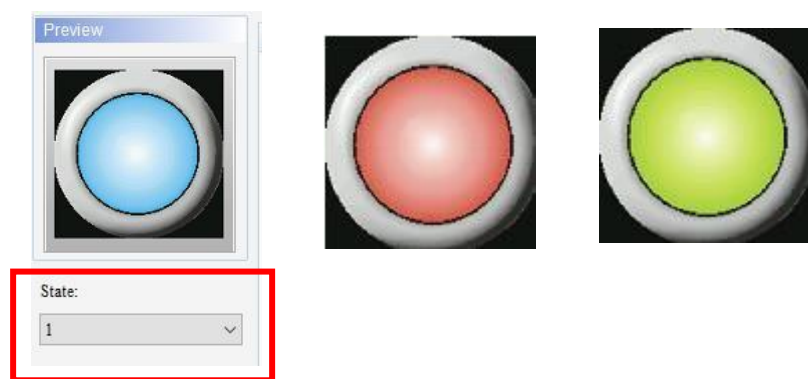
Right click in the blank space, select **Indicator, Multi State Indicator**

Each Button Will have two states, one where the button is not being pressed which is considered **State 0** and the second one (**State 1**) where the button is being pressed the light is on.

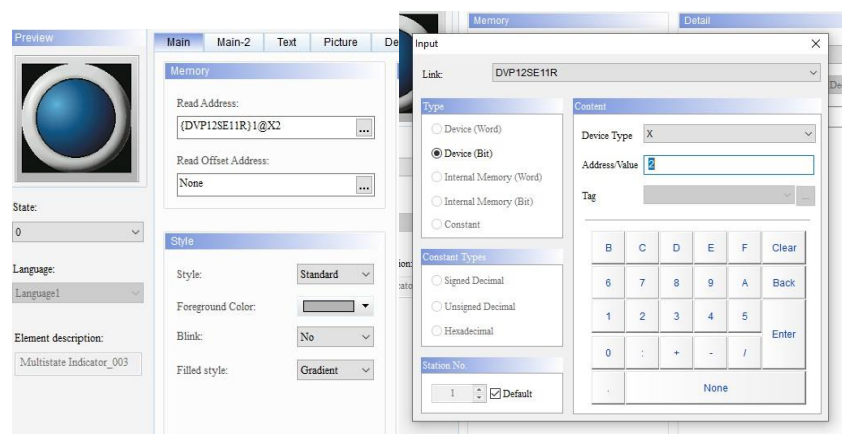
Go to the Picture tab on the multistate indicator settings. Make sure that the State selected is **0** and then go to the picture bank name drop list and select \$3DLamp2State.pib. Choose the darker looking lamp indicators for state 0 to signal that they are OFF!



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:

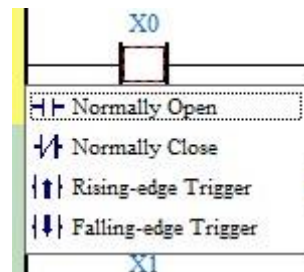


For each indicator we will assign a read address of X0, X1 and X2 corresponding to the networks created in ISPSOFT earlier and the labels on the HMI for each button respectively (for example Multistate Indicator_001 corresponds to X0):

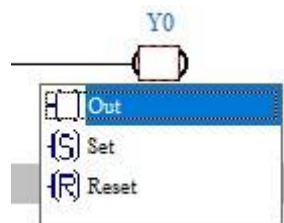


Compile the HMI program by clicking Tools in the toolbar and selecting Compile. Now download the program to the HMI by clicking Download Screen, again under the Tools menu. If there isn't a **"Tools"** tab, then click on the **"Project tab"** and the compile and Download Screen options should be there. If you encounter an error, once again simply click Download Screen again and it should work. Now you can interact with the buttons and see how the HMI responds. To see a response, make sure the run light on the PLC is lit.

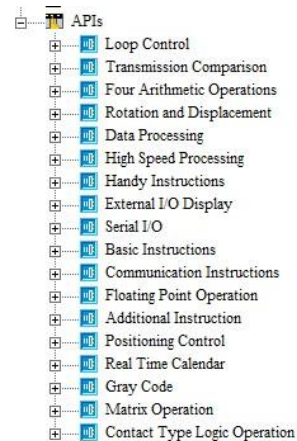
While the HMI DOPsoft tool is very user friendly and obvious to understand, the PLC's ISPsoft has a lot more tools to offer that weren't explored in the examples. Each contact when double clicked has options for when to register a high signal:



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 APIs to use and not enough time in this context 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:

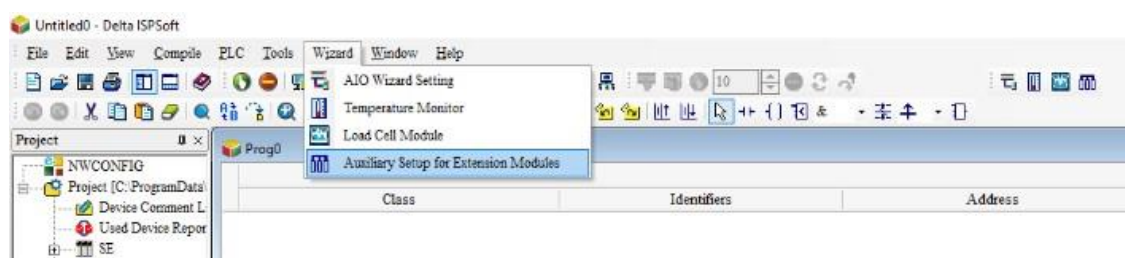
1. Program the PLC to output an analogue reference voltage to the DAC and accept input voltages.
2. Create an interface on the HMI to change said output voltage from any value between -10v to +10v.
3. Create a digital controller using Visual Studio 2019.

3.1 Setting up the PLC

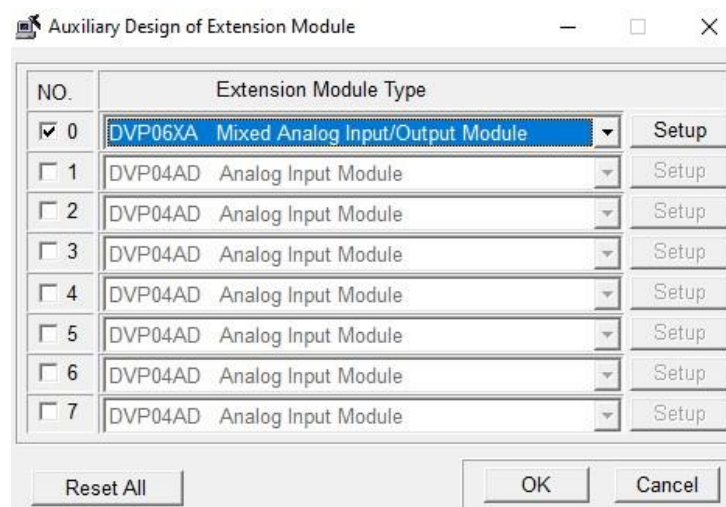
The PLC itself can only output, and accept, Logic states of 0 and 1. For this practical we wish to output a range of voltages from -10v to +10v. Unfortunately, the DVP-06XA analogue I/O extension module only allows for analogue outputs in the range of 0V to +10V. We would also like to accept a range 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 ISPSOft again and go to the top of the toolbar and select **Wizard > Auxiliary Setup for Extension Modules**.

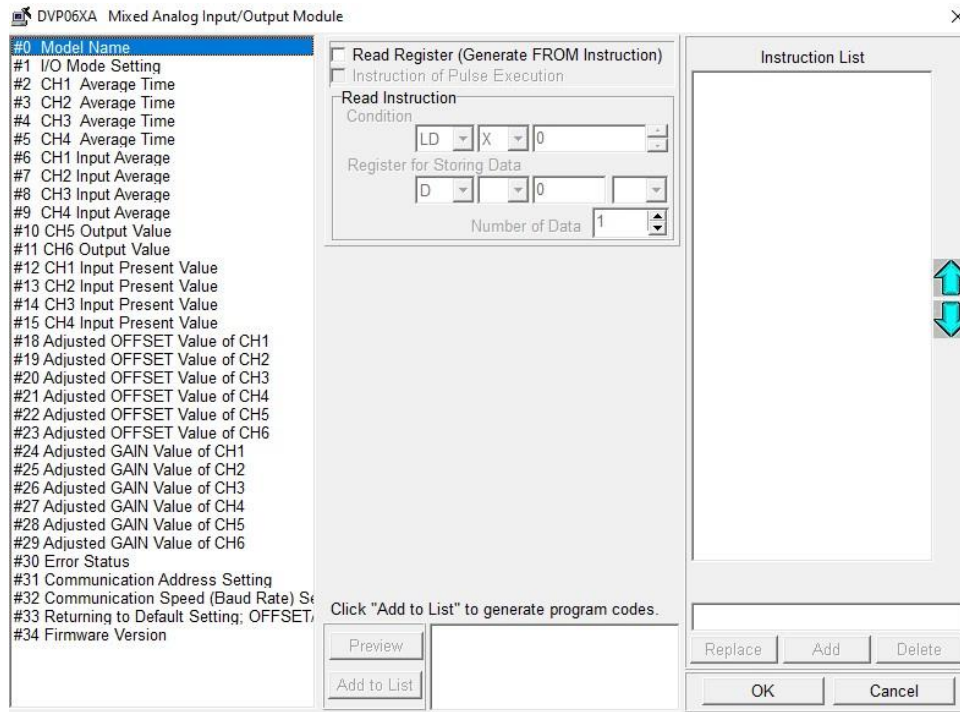


A window will pop-up where you can select the model of the extension module you would like to setup:



Click the No.0 box and from the drop-down list select **DVP06XA** and click Setup. If there is no option for **DVP06XA**, then choosing **DVP06XA-S2** will work fine.

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.

DVP06XA Mixed Analog Input/Output Module

#0 Model Name
#1 I/O Mode Setting
#2 CH1 Average Time
#3 CH2 Average Time
#4 CH3 Average Time
#5 CH4 Average Time
#6 CH1 Input Average
#7 CH2 Input Average
#8 CH3 Input Average
#9 CH4 Input Average
#10 CH5 Output Value
#11 CH6 Output Value
#12 CH1 Input Present Value
#13 CH2 Input Present Value
#14 CH3 Input Present Value
#15 CH4 Input Present Value
#18 Adjusted OFFSET Value of CH1
#19 Adjusted OFFSET Value of CH2
#20 Adjusted OFFSET Value of CH3
#21 Adjusted OFFSET Value of CH4
#22 Adjusted OFFSET Value of CH5
#23 Adjusted OFFSET Value of CH6
#24 Adjusted GAIN Value of CH1
#25 Adjusted GAIN Value of CH2
#26 Adjusted GAIN Value of CH3
#27 Adjusted GAIN Value of CH4
#28 Adjusted GAIN Value of CH5
#29 Adjusted GAIN Value of CH6
#30 Error Status
#31 Communication Address Setting
#32 Communication Speed (Baud Rate) Setting
#33 Returning to Default Setting; OFFSET, GAIN
#34 Firmware Version

☐ Read Register (Generate FROM Instruction)
☐ Instruction of Pulse Execution
Read Instruction
Condition
LD X 0
Register for Storing Data
D 0
Number of Data 1
☒ Write Register (Generate TO Instruction)
☐ Instruction of Pulse Execution
Condition
LD M 1000
Set Value
CH1 Voltage Input Mode -10V ~ +10
CH2 Voltage Input Mode -10V ~ +10
CH3 Voltage Input Mode -10V ~ +10
CH4 Voltage Input Mode -10V ~ +10
CH5 Voltage Output Mode 0V ~ +10
CH6 Voltage Output Mode 0V ~ +10
Click "Add to List" to generate program codes.
Preview
Add to List

Instruction List
LD M1000
TO K0 K1 H0000 K1
Replace Add Delete
OK Cancel

Note: You may notice that the CH5 Set Value option shows **0V ~ +10V**. This is the channel used to output analogue voltages and can only output voltages in that range.

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**.

DVP06XA Mixed Analog Input/Output Module

#0 Model Name
 #1 I/O Mode Setting
 #2 CH1 Average Time
 #3 CH2 Average Time
 #4 CH3 Average Time
 #5 CH4 Average Time
 #6 CH1 Input Average
 #7 CH2 Input Average
 #8 CH3 Input Average
 #9 CH4 Input Average
 #10 CH5 Output Value
 #11 CH6 Output Value
 #12 CH1 Input Present Value
 #13 CH2 Input Present Value
 #14 CH3 Input Present Value
 #15 CH4 Input Present Value
 #18 Adjusted OFFSET Value of CH1
 #19 Adjusted OFFSET Value of CH2
 #20 Adjusted OFFSET Value of CH3
 #21 Adjusted OFFSET Value of CH4
 #22 Adjusted OFFSET Value of CH5
 #23 Adjusted OFFSET Value of CH6
 #24 Adjusted GAIN Value of CH1
 #25 Adjusted GAIN Value of CH2
 #26 Adjusted GAIN Value of CH3
 #27 Adjusted GAIN Value of CH4
 #28 Adjusted GAIN Value of CH5
 #29 Adjusted GAIN Value of CH6
 #30 Error Status
 #31 Communication Address Setting
 #32 Communication Speed (Baud Rate) Setting
 #33 Returning to Default Setting; OFFSET, GAIN, and AVERAGE
 #34 Firmware Version

☐ Read Register (Generate FROM Instruction)
 Instruction of Pulse Execution
 Read Instruction
 Condition LD X 0
 Register for Storing Data D 0
 Number of Data 1

☒ Write Register (Generate TO Instruction)
 Instruction of Pulse Execution
 Condition LD M 1000
 Set Value D 1
 Number of Data 1

Click "Add to List" to generate program codes.

Preview Add to List

Instruction List

```
LD M1000
TO K0 K1 H0000 K1
LD M1000
TO K0 K10 D1 K1
```

Replace Add Delete

OK Cancel

D1 is now the address which sets the output voltage value of CH5. A D register was chosen because 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** (See next page).

DVP06XA Mixed Analog Input/Output Module

#0 Model Name
 #1 I/O Mode Setting
 #2 CH1 Average Time
 #3 CH2 Average Time
 #4 CH3 Average Time
 #5 CH4 Average Time
 #6 CH1 Input Average
 #7 CH2 Input Average
 #8 CH3 Input Average
 #9 CH4 Input Average
 #10 CH5 Output Value
 #11 CH6 Output Value
 #12 CH1 Input Present Value
 #13 CH2 Input Present Value
 #14 CH3 Input Present Value
 #15 CH4 Input Present Value
 #18 Adjusted OFFSET Value of CH1
 #19 Adjusted OFFSET Value of CH2
 #20 Adjusted OFFSET Value of CH3
 #21 Adjusted OFFSET Value of CH4
 #22 Adjusted OFFSET Value of CH5
 #23 Adjusted OFFSET Value of CH6
 #24 Adjusted GAIN Value of CH1
 #25 Adjusted GAIN Value of CH2
 #26 Adjusted GAIN Value of CH3
 #27 Adjusted GAIN Value of CH4
 #28 Adjusted GAIN Value of CH5
 #29 Adjusted GAIN Value of CH6
 #30 Error Status
 #31 Communication Address Setting
 #32 Communication Speed (Baud Rate) Setting
 #33 Returning to Default Setting; OFFSET, GAIN, and AVERAGE
 #34 Firmware Version

☒ Read Register (Generate FROM Instruction)
 Instruction of Pulse Execution
 Read Instruction
 Condition LD M 1000
 Register for Storing Data D 2
 Number of Data 1

Click "Add to List" to generate program codes.

Preview Add to List

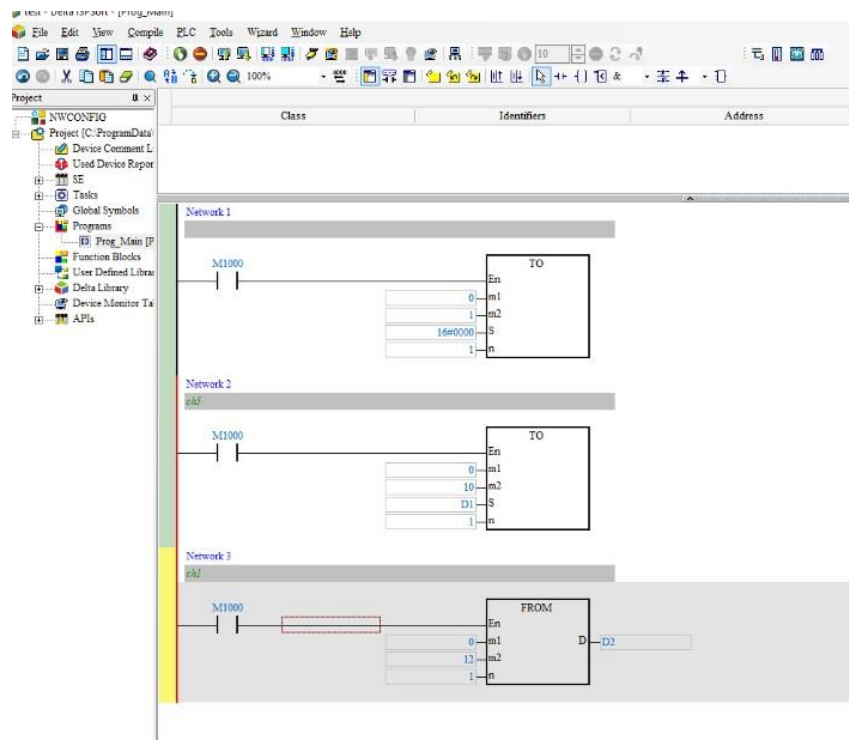
Instruction List

```
LD M1000
TO K0 K1 H0000 K1
LD M1000
TO K0 K10 D1 K1
LD M1000
FROM K0 K12 D2 K1
```

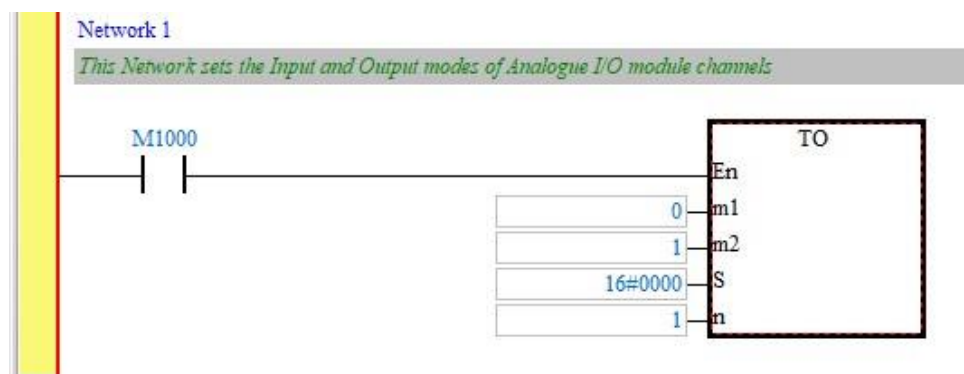
Replace Add Delete

OK Cancel

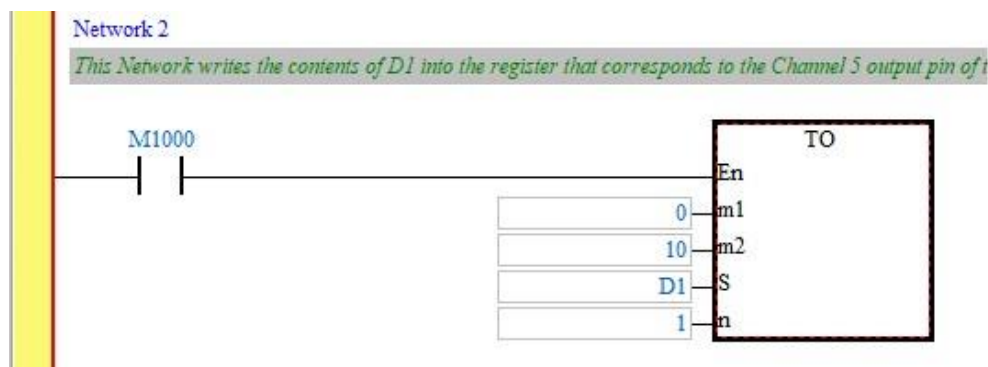
Once this is done click **OK**, and then **OK** again on the Wizard window. Your main program will now have 3 new networks. Each corresponding to the CR configurations that you made earlier:



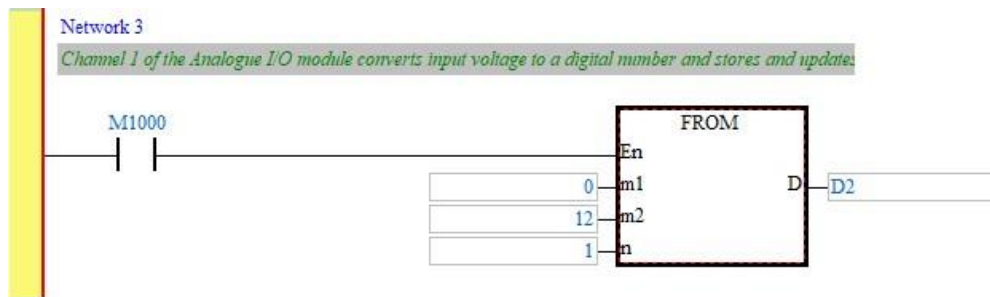
Network 1's function is illustrated by its comment:



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**:



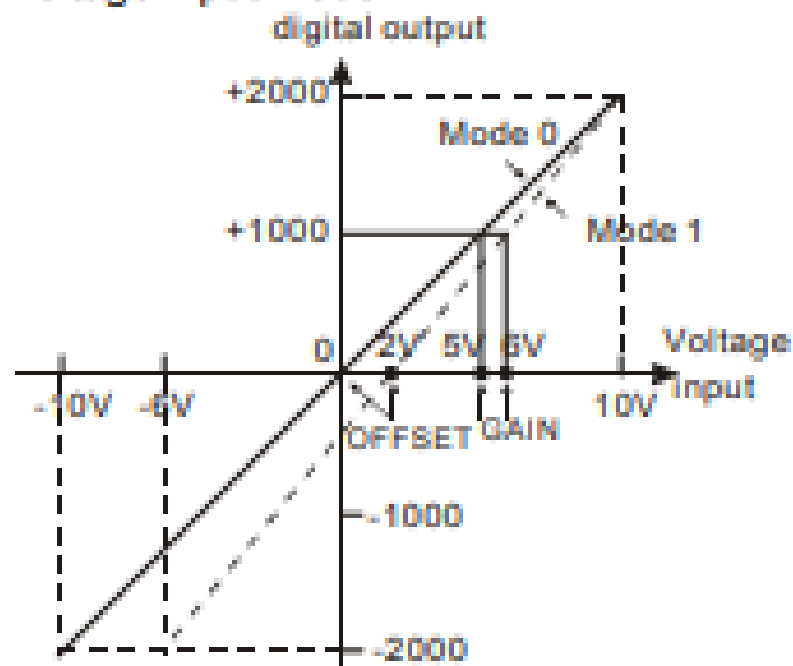
Network 3 Channel 1 of the Analogue I/O module converts input voltage to a digital number and stores and updates the value in **D2**:



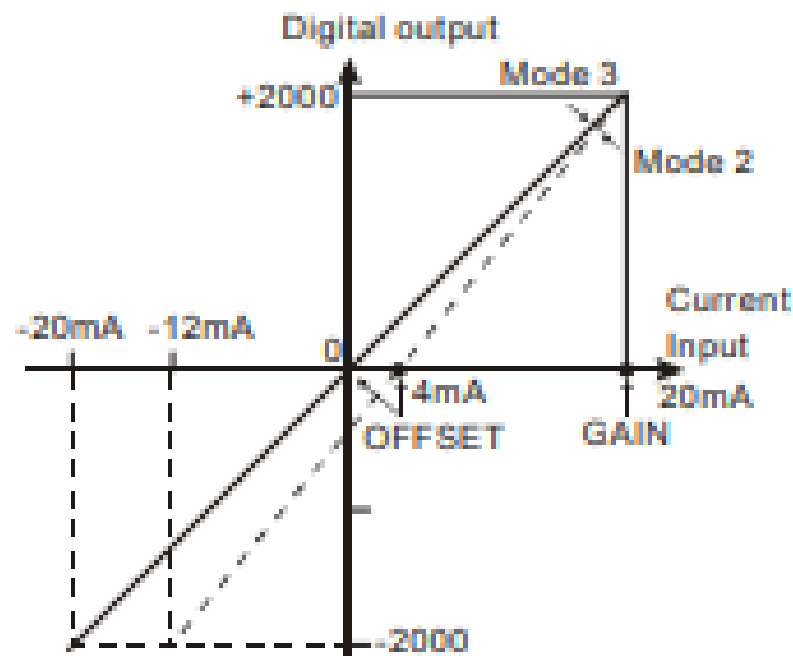
In Each network **M1000** is used as it has a constant state of 1, ensuring that the API block's (TO/FROM) are constantly enabled while the PLC is running.

Now that we have set up the digital to analogue 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 on the next page show the A/D Conversion Characteristic Curve of CH1 ~ CH4:

Voltage input mode

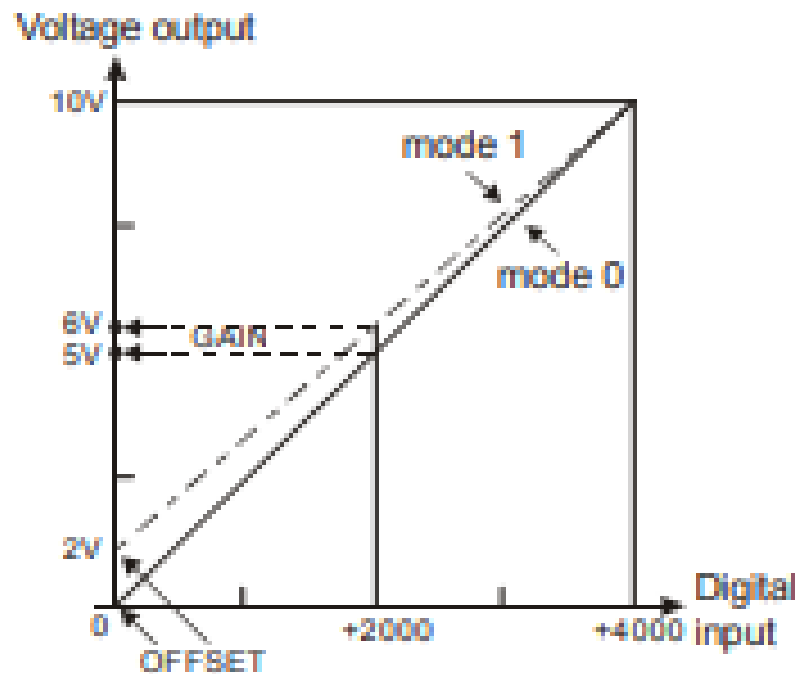


Current input mode:

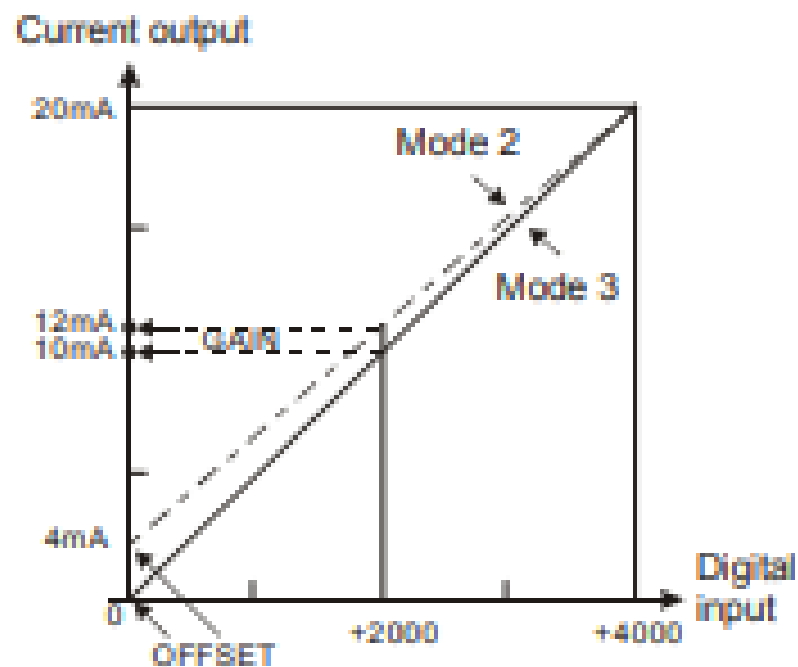


The illustrations below show the D/A Conversion Characteristic Curve of CH5 ~ CH6:

Voltage output mode



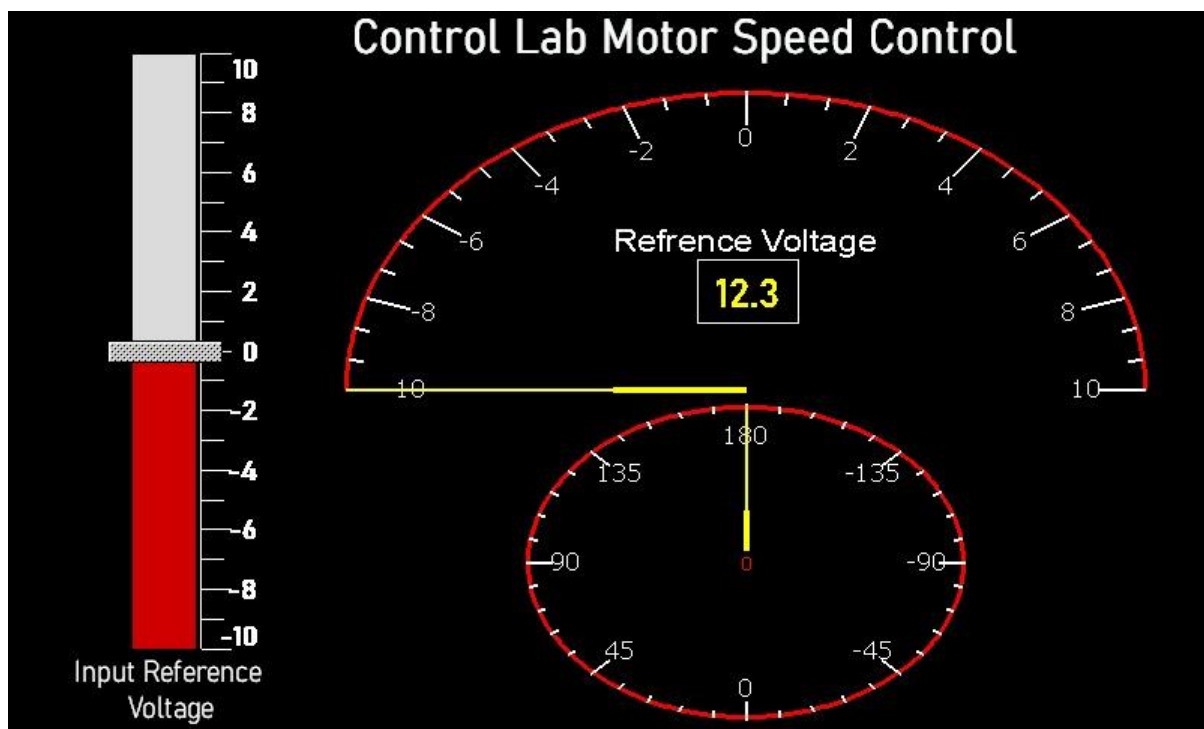
Current output mode:



3.2 Create a Functional HMI Screen

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 requirements are 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).

3.3 Design a Digital Controller and Implement using Visual 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

Device	Range		Type	Effective range		DVP address (Hexadecimal)	Modbus address (Decimal)
				DVP14SS211R	DVP12SE11R		
M	3840~4095		Bit	0000~4095		B900~B9FF	047361~047616
C	0~199	16-bit	Bit	000~199		0E00~0EC7	003585~003784
			Word	000~199		0E00~0EC7	403585~403784
	200~255	32-bit	Bit	200~255		0EC8~0EFF	003785~003840
			Dword	200~255		0EC8~0EFF	403785~403840
D	000~255		Word	0000 ~ 4999	0000 ~ 11999	1000~10FF	404097~404352
D	256~511		Word			1100~11FF	404353~404608
D	512~767		Word			1200~12FF	404609~404864
D	768~1023		Word			1300~13FF	404865~405120
D	1024~1279		Word			1400~14FF	405121~405376
D	1280~1535		Word			1500~15FF	405377~405632
D	1536~1791		Word			1600~16FF	405633~405888
D	1792~2047		Word			1700~17FF	405889~406144
D	2048~2303		Word			1800~18FF	406145~406400
D	2304~2559		Word			1900~19FF	406401~406656
D	2560~2815		Word			1A00~1AFF	406657~406912
D	2816~3071		Word			1B00~1BFF	406913~407168
D	3072~3327		Word			1C00~1CFF	407169~407424
D	3328~3583		Word			1D00~1DFF	407425~407680
D	3584~3839		Word			1E00~1EFF	407681~407936
D	3840~4095		Word			1F00~1FFF	407937~408192
D	4096~4351		Word			9000~90FF	436865~437120
D	4352~4607		Word			9100~91FF	437121~437376
D	4608~4863		Word			9200~92FF	437377~437632
D	4864~5119		Word			9300~93FF	437633~437888

Device	Range	Type	Effective range		DVP address (Hexadecimal)	Modbus address (Decimal)
			DVP14SS211R	DVP12SE11R		
D	5120~5375	Word	N/A	0000 ~ 11999	9400~94FF	437889~438144
D	5376~5631	Word			9500~95FF	438145~438400
D	5632~5887	Word			9600~96FF	438401~438656
D	5888~6143	Word			9700~97FF	438657~438912
D	6144~6399	Word			9800~98FF	438913~439168
D	6400~6655	Word			9900~99FF	439169~439424
D	6656~6911	Word			9A00~9AFF	439425~439680
D	6912~7167	Word			9B00~9BFF	439681~439936
D	7168~7423	Word			9C00~9CFF	439937~440192
D	7424~7679	Word			9D00~9DFF	440193~440448
D	7680~7935	Word			9E00~9EFF	440449~440704
D	7936~8191	Word			9F00~9FFF	440705~440960
D	8192~8447	Word			A000~A0FF	440961~441216
D	8448~8703	Word			A100~A1FF	441217~441472
D	8704~8959	Word			A200~A2FF	441473~441728
D	8960~9215	Word			A300~A3FF	441729~441984
D	9216~9471	Word			A400~A4FF	441985~442240
D	9472~9727	Word			A500~A5FF	442241~442496
D	9728~9983	Word			A600~A6FF	442497~442752
D	9984~9999	Word			A700~A7FF	442753~443008
D	10000~11999	Word			A710~AEDF	442767~44768