**Practical Manual**

**EEE4118F**

**INTERFACING THE DELTA DVP-PLC**

**WITH AN HMI SCREEN**

A picture containing emblem, symbol, text, logo

Description automatically generated

**Authors:**

**Dino Kingsley-Rizzo**

**John Mkwebo**

**Katerina Itopoulos**

**Nicolen Pillay**

**Luke Baatjes**

**Dominic de Maar**

**S’thabiso Lushaba**

Table of Contents

[**1.** **Pre-Reading** 3](#_Toc139013809)

[**1.1 Delta DVP-PLC** 3](#_Toc139013810)

[**1.1.1** **Hardware** 3](#_Toc139013811)

[**1.1.2** **Software** 6](#_Toc139013812)

[**2.** **Pre-Practical** 8](#_Toc139013813)

[**2.1** **ISPSoft** 8](#_Toc139013814)

[**2.2** **DCISoft** 14](#_Toc139013815)

[**2.3** **DOPSoft** 16](#_Toc139013816)

[**2.3.1** **DOP-B07E415 HMI (Older Model No White Line Along Bezel)** 16](#_Toc139013817)

[**2.3.2** **DOP-107EV HMI (Newer Model with White Line Along Bezel)** 19](#_Toc139013818)

[**3.** **Practical** 23](#_Toc139013819)

[**3.1** **Setting up the PLC** 23](#_Toc139013820)

[**3.2** **Create a Functional HMI Screen** 31](#_Toc139013821)

[**3.3** **Design a Digital Controller and Implement using Visual Studio** 31](#_Toc139013822)

[**4.** **Table of Device Types and their Respective Addresses** 32](#_Toc139013823)

# **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](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**

**A close-up of a device

Description automatically generated with medium confidenceDELTA DVP-12SE**

Figure -DVP12SE PLC

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 right-side modules
* Has a mini-USB port, Ethernet port and two RS-485 ports
* No battery required for maintenance of DVP-06XA

**A close-up of a device

Description automatically generated with low confidenceDELTA DVP-PS01**

Figure -DVP-PS01 Power Supply

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)

A close-up of a device

Description automatically generated with medium confidence**Delta DVP-06XA**

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

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

**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:**

Figure -DOP-B07E415 HMI

* 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.

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

**A picture containing rectangle, monitor, screenshot, screen

Description automatically generatedDOP-107EV**

Figure -DOP-107EV HMI

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.

**Communication Parameters**

Table -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 |

### **Software**

**ISPSoft**

A picture containing box, container, design

Description automatically generatedISPSoft 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

Figure -ISPSoft

• 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. 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 -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

A picture containing text, receipt

Description automatically generated

Figure -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.

# **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.

## **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.

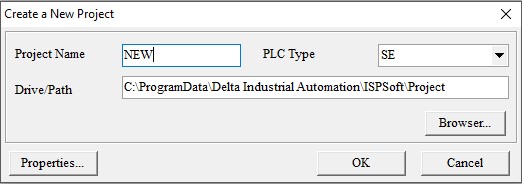
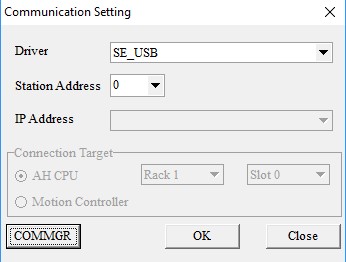
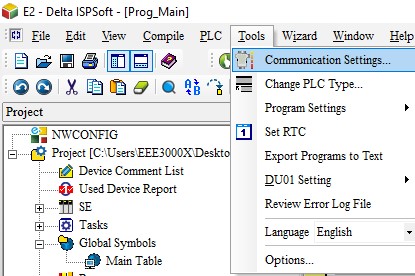
Before starting a new project ensure that the following communication settings have been configured.

Ensure **COMMGR** is running in the background and proceed to 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 **Add** on the COMMGR if you port cannot be found, change connection setup to type: USB(Virtual COM) and choose the Delta PLC port for the communication port. You can find the Delta PLC COM via the device manager.

Click on

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

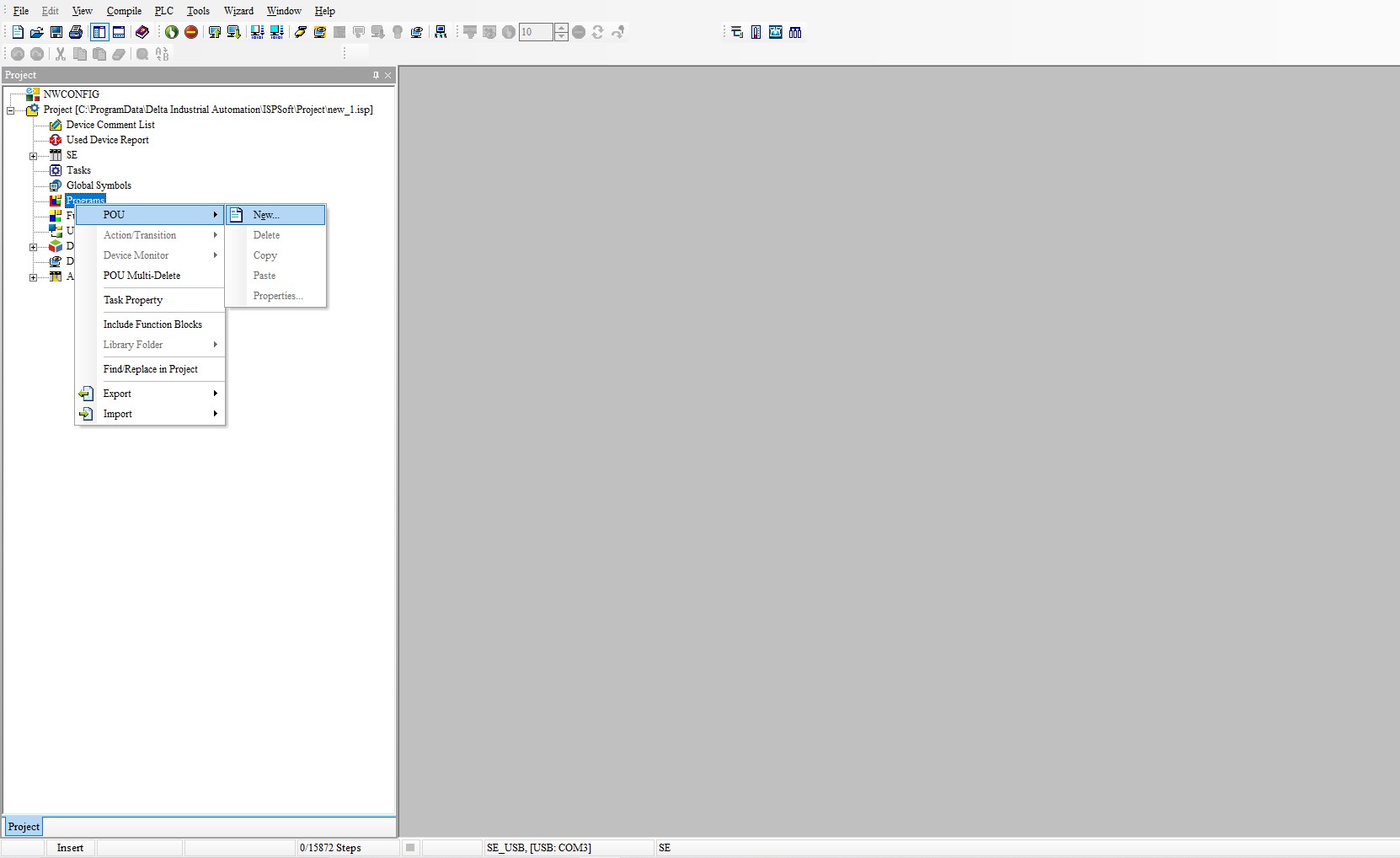


Rename your project and select SE in the PLC Type drop down list. Once ok is clicked a project environment will be displayed.

A screenshot of a computer

Description automatically generated with medium confidence

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.

A screenshot of a program

Description automatically generated with medium confidence

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

A picture containing screenshot, text, software, display

Description automatically generated

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:

A close-up of a computer screen

Description automatically generated with low confidence

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.

A picture containing line, font, text, white

Description automatically generated

(**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.

A picture containing text, font, symbol, line

Description automatically generated

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.

A picture containing text, sketch, font, white

Description automatically generated

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

A picture containing text, line, font, diagram

Description automatically generated

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.

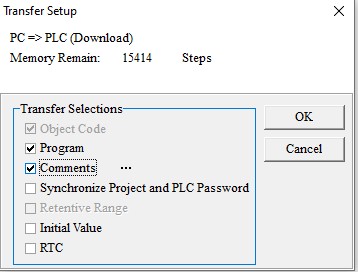
**\*sometimes** ISPsoft has a communication error with the PLC and can’t stop the PLC via the virtual button. In this scenario you can simply wait till the program is ready to proceed and clicking **stop** again or using the physical switch on the PLC will work and should be used (located under the Run 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, .



A screenshot of a computer

Description automatically generated with low confidence

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

## **DCISoft**

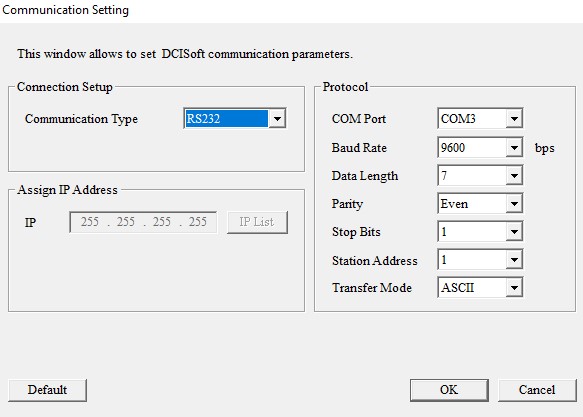
Open DCISoft (version 1.12). Once DCISoft is open, you’ll see the following screen:

A picture containing text, software, screenshot, multimedia software

Description automatically generated

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

Communication Type is set to RS232.



A picture containing text, font, screenshot, logo

Description automatically generatedGo 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:

A screenshot of a computer

Description automatically generated with medium confidence

***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:

A picture containing screenshot, rectangle

Description automatically generated

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).

A screenshot of a computer

Description automatically generated

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

## **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).The PLC to Desktop connection should have two USB connections, the other one is for the HMI. Ensure the connections are all as should be as well as take note of the model you are using and updating the firmware (**only DOP107EV HMI**).

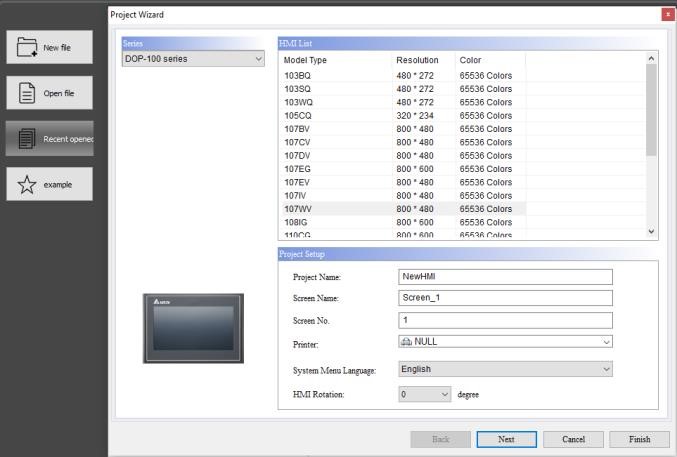
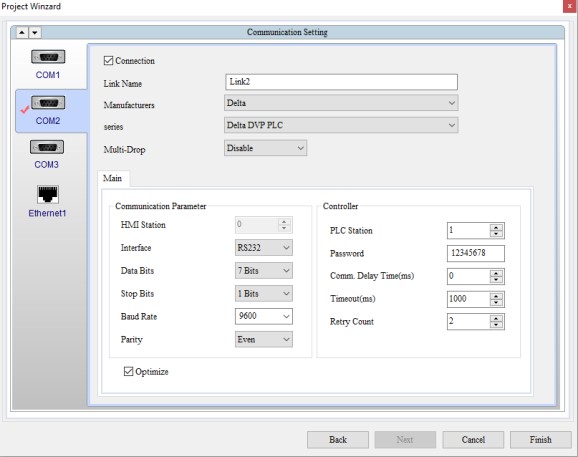
### **DOP-B07E415 HMI (Older Model No White Line Along Bezel)**

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).

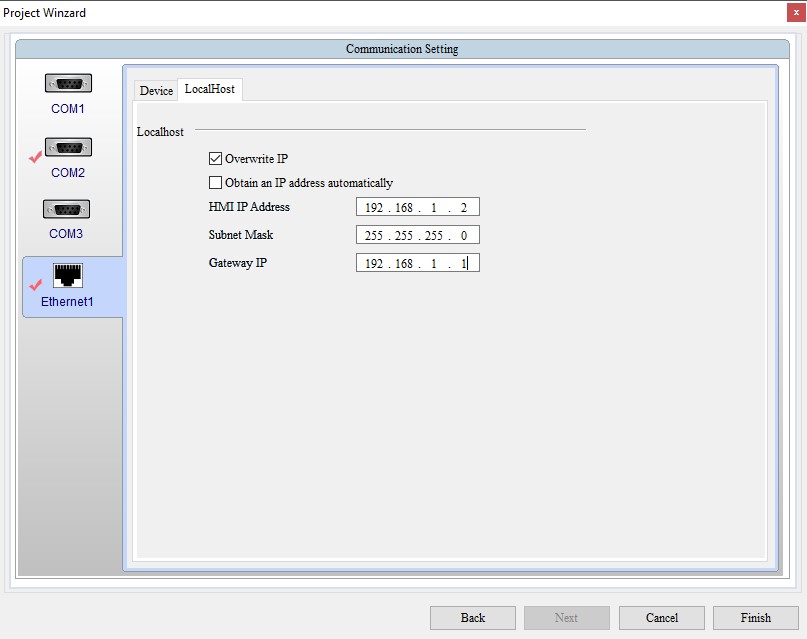
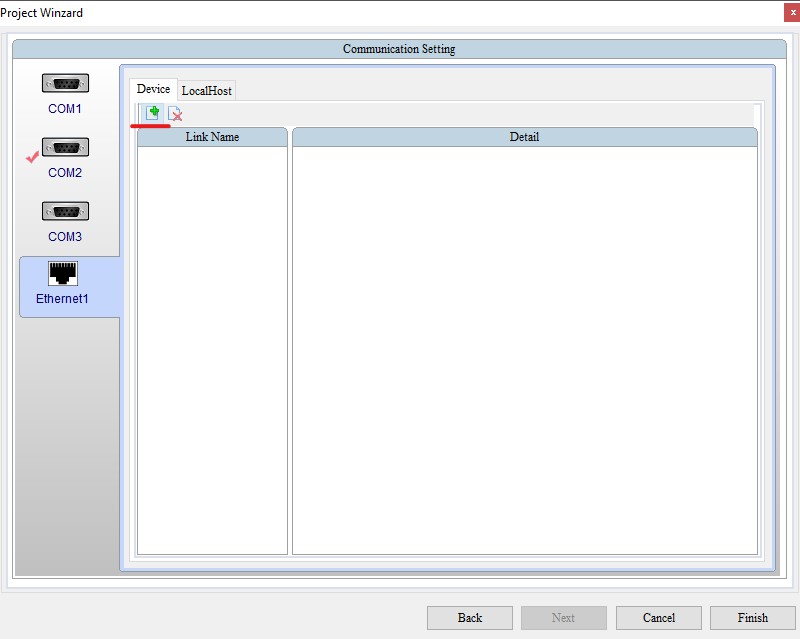
A screenshot of a computer

Description automatically generated with medium confidence

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:

A picture containing circle, colorfulness, graphics, art

Description automatically generated

To create each button:

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

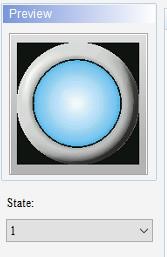
Each Button Will have two states, one where the button is not being pressed which is considered **State 0** and the second one (**State 1**) where the button is being pressed the light is on.

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!

A screenshot of a computer

Description automatically generated

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):

A picture containing text, screenshot, software, computer icon

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

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.

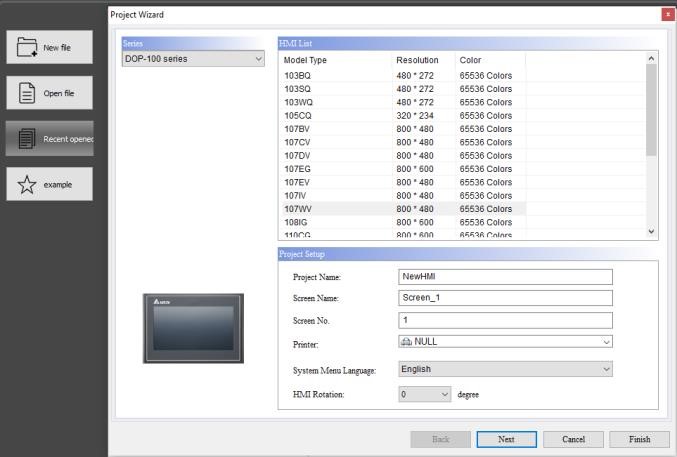
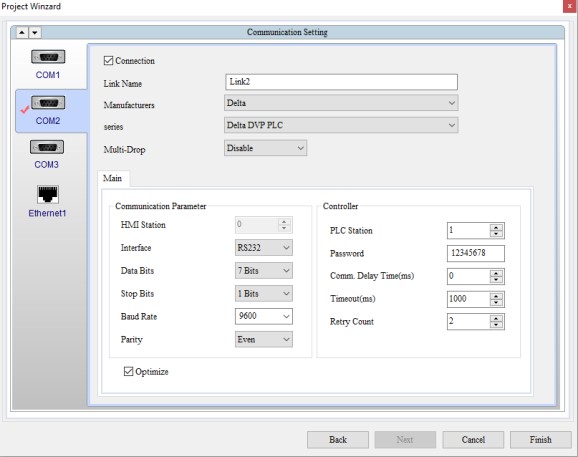
### **DOP-107EV HMI (Newer Model with White Line Along Bezel)**

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).

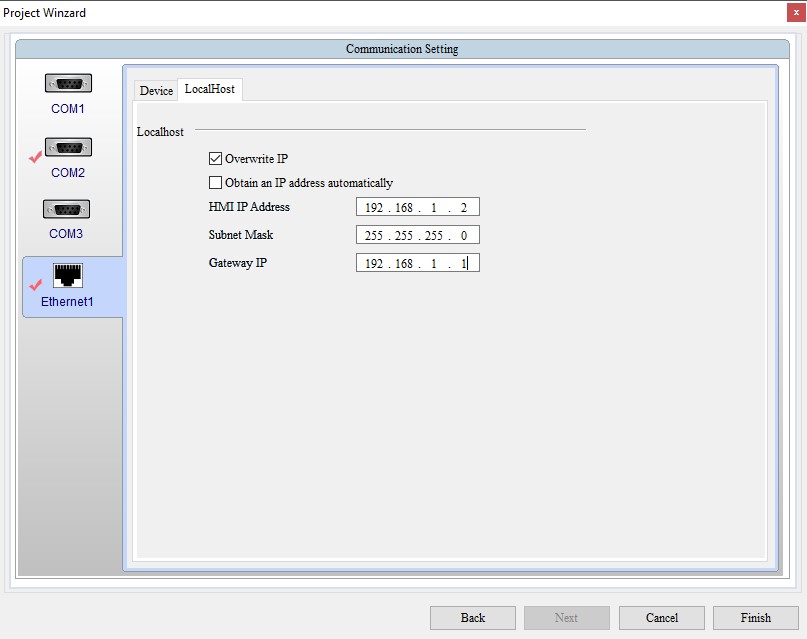
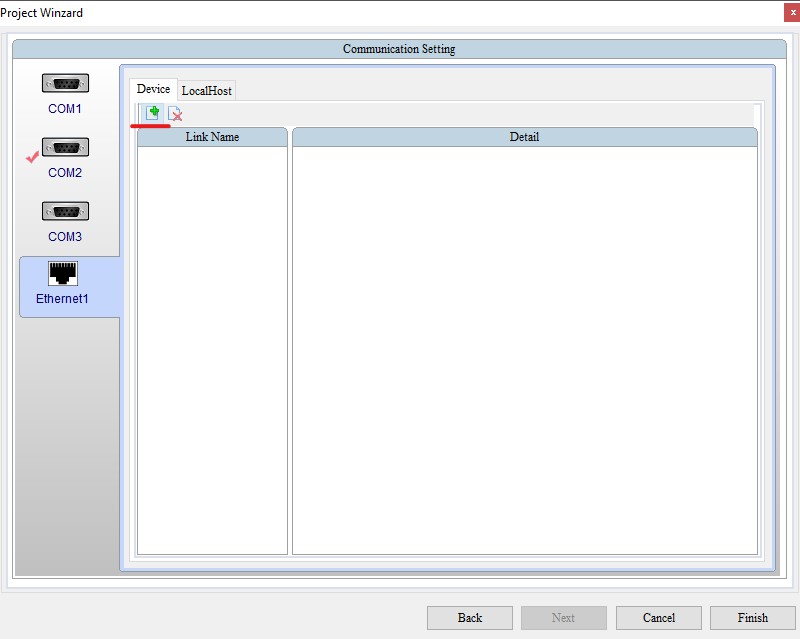
A screenshot of a computer

Description automatically generated with medium confidence

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:

A picture containing circle, colorfulness, graphics, art

Description automatically generated

To create each button:

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

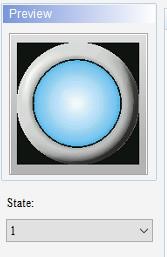
Each Button Will have two states, one where the button is not being pressed which is considered **State 0** and the second one (**State 1**) where the button is being pressed the light is on.

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!

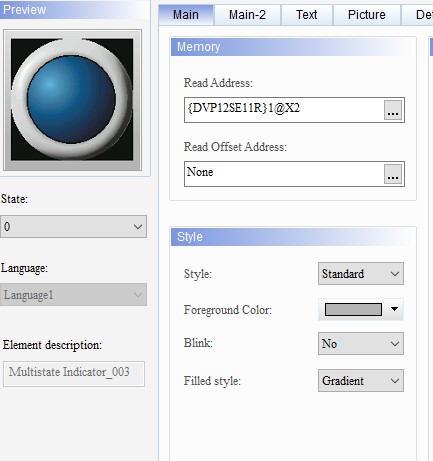
A screenshot of a computer

Description automatically generated

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):

A screenshot of a computer

Description automatically generated with medium confidence

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:

A picture containing text, font, screenshot, line

Description automatically generated

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

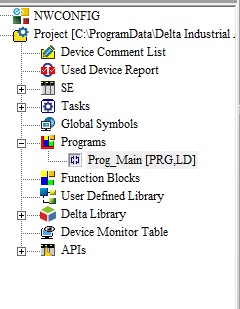
A screenshot of a computer program

Description automatically generated with low confidence

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

A picture containing text, screenshot, font, number

Description automatically generated



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).

# **Practical**

**Main objectives of the practical**:

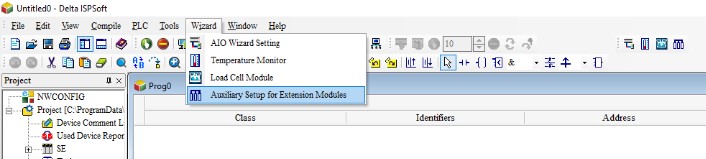
1. Program the PLC to output an analogue reference voltage to the DAC and accept input voltages.
2. Create an interface on the HMI to change said output voltage from any value between -10v to +10v.
3. Create a digital controller using Visual Studio 2019.

## **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**. If the Wizard tab isn’t available, you may switch over to IS



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

A screenshot of a computer

Description automatically generated with medium confidence

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:

A screenshot of a computer program

Description automatically generated with low confidence

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.

A screenshot of a computer program

Description automatically generated with low confidence

**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**.

A screenshot of a computer program

Description automatically generated with medium confidence

**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).

A screenshot of a computer

Description automatically generated with medium confidence

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:

A screenshot of a computer

Description automatically generated

Network 1’s function is illustrated by its comment:

A screenshot of a computer

Description automatically generated with medium confidence

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:**

A screenshot of a computer

Description automatically generated with medium confidence

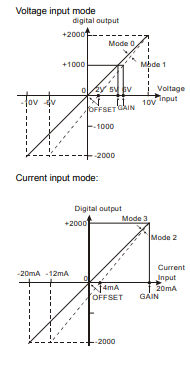
Network 3 Channel 1 of the Analogue I/O module converts input voltage to a digital number and stores and updates the value in **D2**:

A picture containing text, line, diagram, parallel

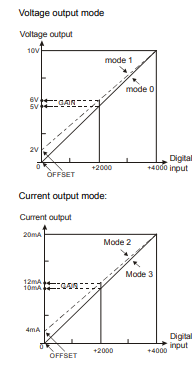
Description automatically generated

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:



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



## **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:

A picture containing text, screenshot, measuring instrument, clock

Description automatically generated

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.

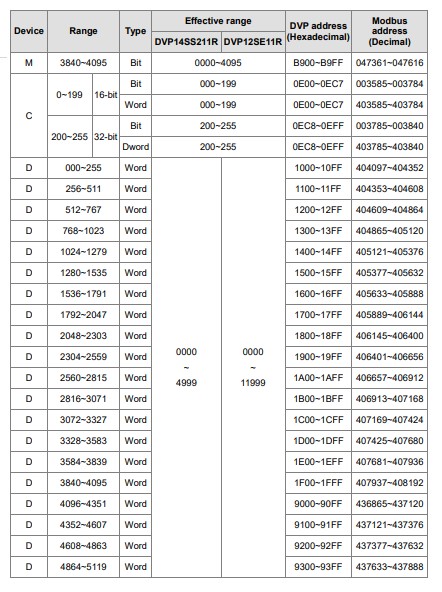
It is likely that you will have to create more Networks on the PLC programme in order to achieve this.

Hint: The D/A convertor graph is very useful for the design of the HMI s5creen. 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).

## **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.

# **Table of Device Types and their Respective Addresses**



A picture containing text, screenshot, number, document

Description automatically generated