Invertek Drives.com

OptiTools Studio

This Document is for use with version 2.0.0.1.

Optitools Studio User Guide Revision 1.00

Invertek Drives Ltd adopts a policy of continuous improvement and whilst every effort has been made to provide accurate and up to date information, the information contained in this User Guide should be used for guidance purposes only and does not form the part of any contract.

Revision History

Revision	Note	Date
1.00	First Release	23/04/2014

This User Guide is intended to provide an overview of the functions of the Optitools Studio. It is assumed that the reader is familiar with the operation of the respective drive range products to a high level, and this user guide is not intended to explain the operation of those functions, only to provide data on how the PC software operates. For the Function Block Editor, it is assumed that the reader has an understanding of function block programming in general, and of logic and arithmetic functions.

1. Optitools Studio Overview

1.1. Introduction

Optitools Studio is a dedicated PC software support package for Invertek Drives product range. The software has four key functions:-

- Parameter Editor
  - Parameter Upload
  - Parameter Download
  - Parameter Compare
  - Parameter Access Control
  - Parameter Reset
  - Online Parameter Access
- Drive Control / Monitor
  - Real time monitoring of drive status and variables
  - Real time control of the drive for commissioning
- Function Block Editor
  - Create dedicated Function Block programs
  - Simulation of created Programs
  - Program Upload
  - Program Download
- Scope / Datalogger
  - Real time scopemeter for data logging and measurement
  - Access to the drive internal data logger for supported products

This document describes the features and functions of the software, specifically the Function Block Editor, which allows function block programs to be created and used with supported products (Optidrive P2, HVAC and HVAC Eco.)
The purpose of this document is to show the user how to prepare and upload their own programmes to meet their application requirements. The document consists of the following parts:

- Basic principles
- What you'll need and where to get it
- Licencing the programme
- Getting Started
- Step by Step first programme
- The Functions explained
- Some Example Programmes

The document is not intended to show how to prepare programmes to meet particular requirements, or the general principles of programming for industrial applications. There are no 'tips and tricks' explaining which functions should be used in a sequencer for a cleaning machine for example. However, it should be clear that, with the wide variety of functions available, quite complex programmes can be prepared for many different applications. This allows the elimination of many external components, including PLCs, logic units, relays etc. leading to considerable cost reduction and improved reliability.
This document assumes the user has some familiarity with drives and drive technology, as well as a basic understanding of logic and mathematical functions. The user should also have experience with the parameters and functions of the P2 or HVAC drive, preferably using the Parameter Editor which is part of the OptiTools Studio software.
Note that, due to upgrades and changes, there may be slight variations between the screen shots used in this document and the appearance of the latest OTS version. The functionality is not affected. This document was created using Version 2.0.0.1 of OTS.

1.2. System Requirements

Optitools Studio requires the following:-

- A PC with
  - Windows XP
  - Windows Vista
  - Windows 7
  - Windows 8
- Windows Installer 3.1
- .NET Framework 3.5 SP1

In order to communicate with a drive from the PC, the following options bay be used

- Communication via Bluetooth
  - Requires a Bluetooth interface or dongle to be present on the PC
  - Requires an Optistick OPT-2-STICK-IN
  - Requires the Optistick to be paired with the PC
- Communication via RS485
  - Requires a PC Connection kit OD-485AD-IN

1.3. Basic Principles

The PLC programming function is included in all P2 and HVAC drives as standard. It is used and activated by setting certain parameters and uploading a programme, prepared in a PC, using OptiTools Studio (OTS). OTS is a software package, developed and released by Invertek Drives Ltd which can be freely downloaded from their website.
http://www.invertekdrives.com/variable-speed-drives/software/optitools-studio.aspx
OTS also includes many other useful features for working with Invertek drives.
Once installed, programmes can be easily prepared using a graphical 'drag, drop and connect' approach. The programmes may be simulated within the PC, and if required can then be uploaded to a P2 or HVAC drive using Invertek standard interfaces and OTS. By setting a few parameters in the drive the PLC programme may be then activated.
It is possible to try out the Function Block Editor in OTS as soon as OTS has been downloaded, but in order save, print or upload a programme it is necessary to purchase a Licence from Invertek.
The following sections describe in detail how to do this.

1.4. Licencing

All features of Optitools Studio except the function block editor are fully functional without any licence requirements. In order to use the PLC function, a valid licence is required. Licences are provided for an OTS installation on a single PC, and may be transferred from one PC to another. Licences allow unlimited uploads, storage etc.
Two types of Licence are available:-

- A Trial Licence, valid for 60 Days with some functional restrictions
- A full Licence, valid indefinitely without any functional restrictions

Full licences are available to purchase from your local Invertek Sales Partner. You will be issued with an order code. Once the necessary authorisation has been obtained, follow the procedure shown below.
60 day evaluation Licences with certain limitations can also be activated. Follow the same procedure, but check the "Request Evaluation Licence" box, fill out the rest of the form, but leave the order code blank.
This will allow all functions to be used.
The OTS Version and Licence Status can be displayed under: Help >About OTS.
The other functions of OTS (Parameter Editor, Scope/data logger, Drive Control and Monitor) do not need Licencing for full functionality.

1.4.1. Optitools Studio Licencing Procedure - Applying for an Evaluation Licence

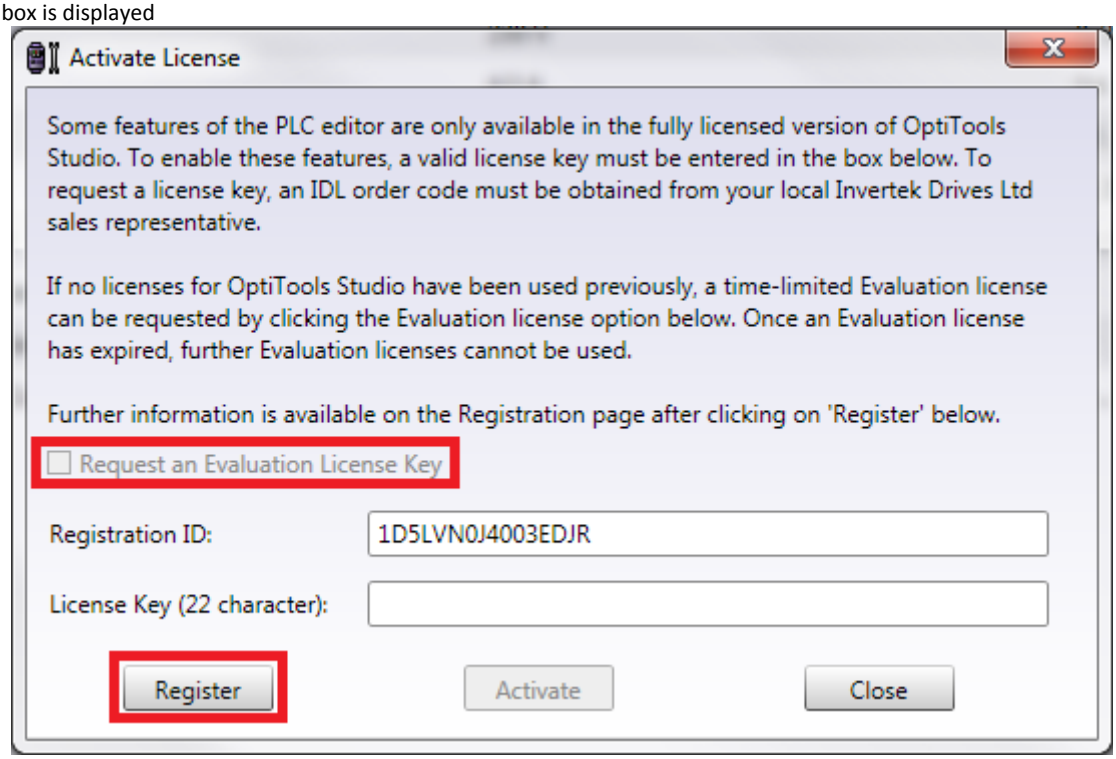
Evaluation licences are valid for 60 days, and allow use of the PLC programming function without this period. Only one evaluation licence may be used, and when this expires, the option to apply for a further evaluation licence will no longer be available.
Evaluation licences allow full use of the software with the following restrictions

- Maximum 60 Day usage
- Projects may not be loaded to an Optistick

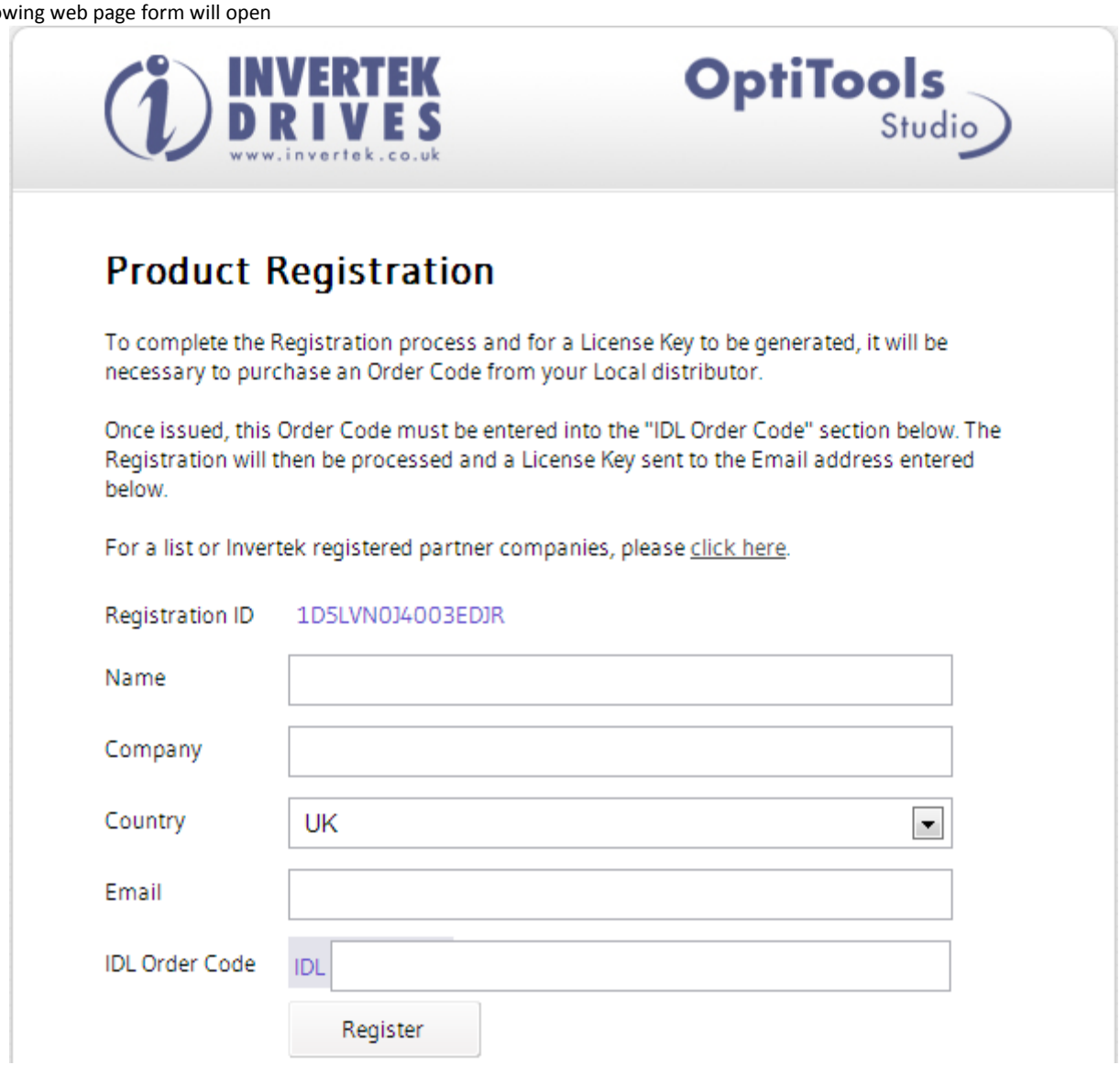
To apply for an evaluation licence

- A working internet connection is required throughout the process
- Open the OptiTools Studio Software
- On the toolbar, click on **Help** > **Activate Licence**

The following box is displayed

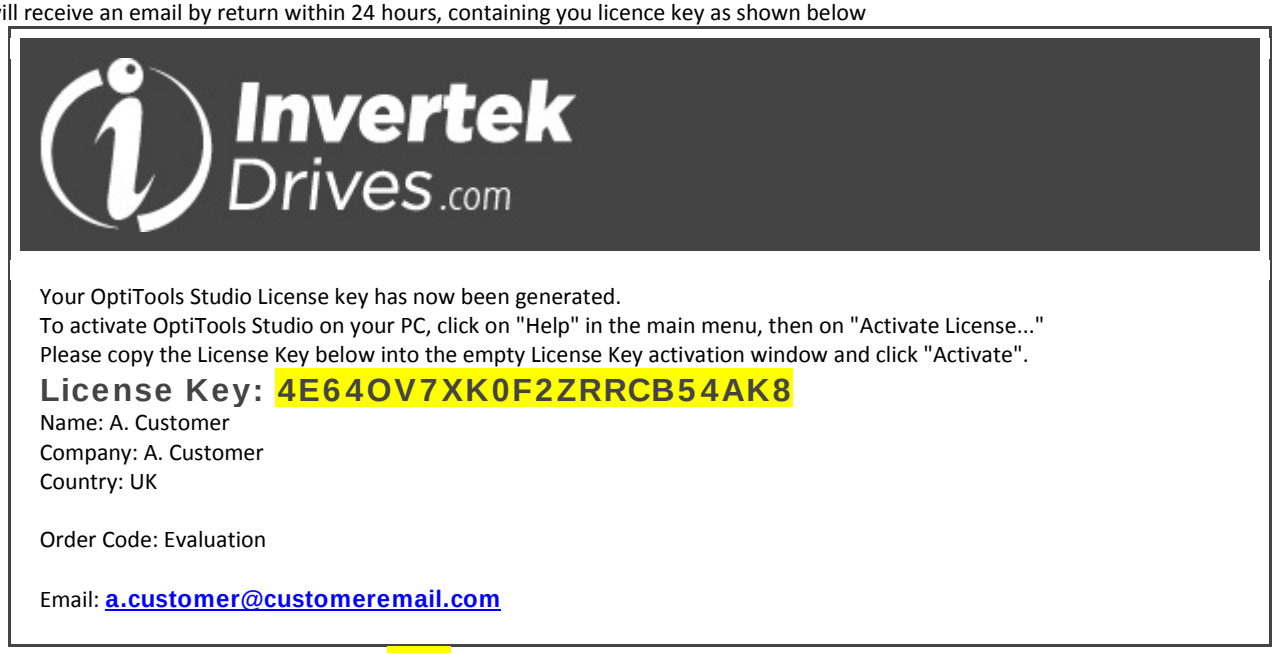


Tick the "**Request Evaluation Licence**" box as highlighted, and then click **Register**.
The following web page form will open



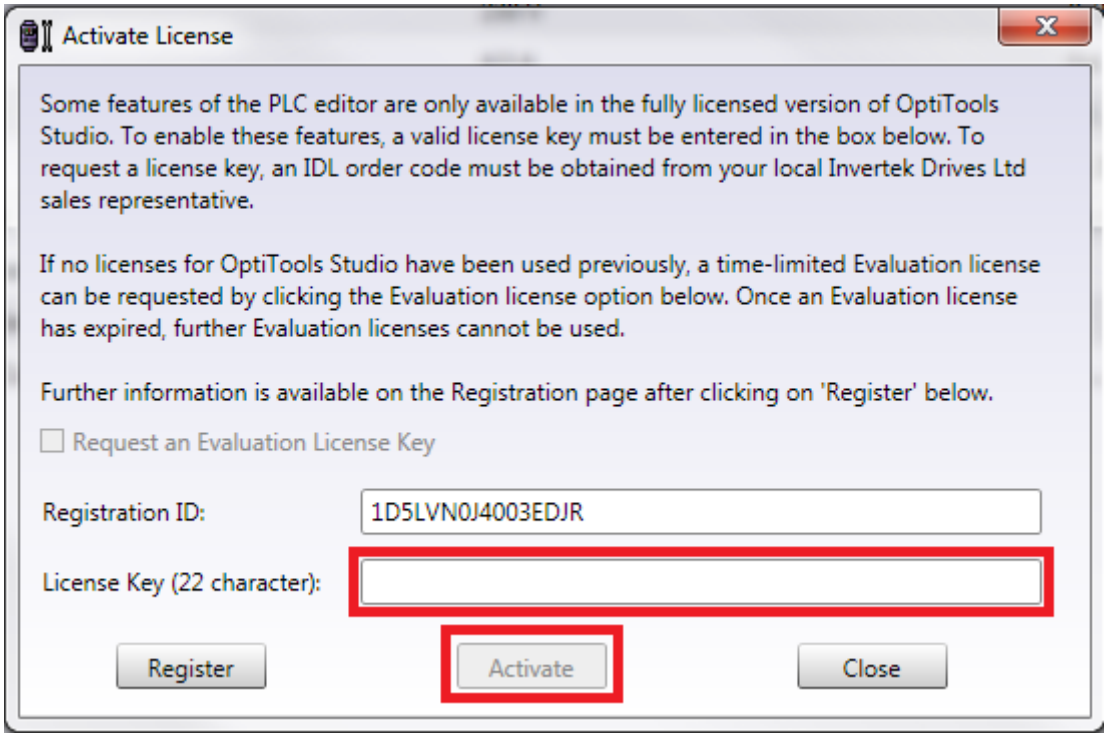
- On the web form, fill out the required details
- Leave the IDL order field blank.
- Click Register.

You will receive an email by return within 24 hours, containing you licence key as shown below



- Copy the licence key, highlighted in yellow above
- In the OptiTools Studio software, on the toolbar, click on **Help** > **Activate Licence**

The following screen will open



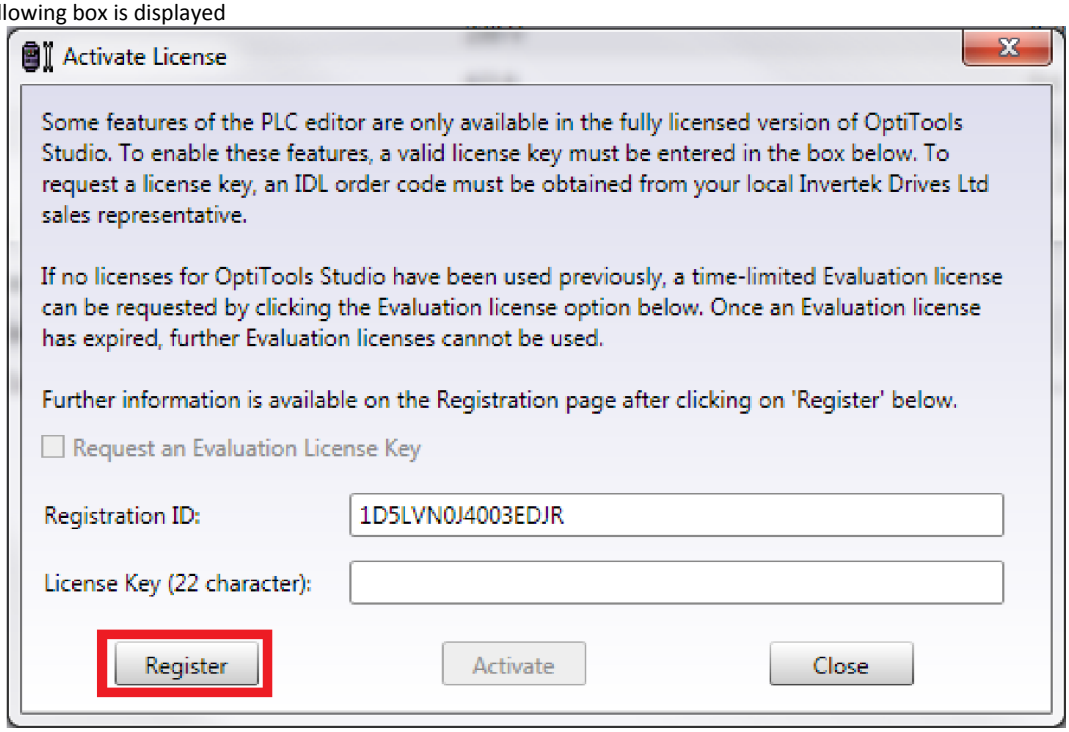
- Paste the licence key into the box shown
- Click on Activate. The licence is now activated

1.4.2. Applying for a Full Licence

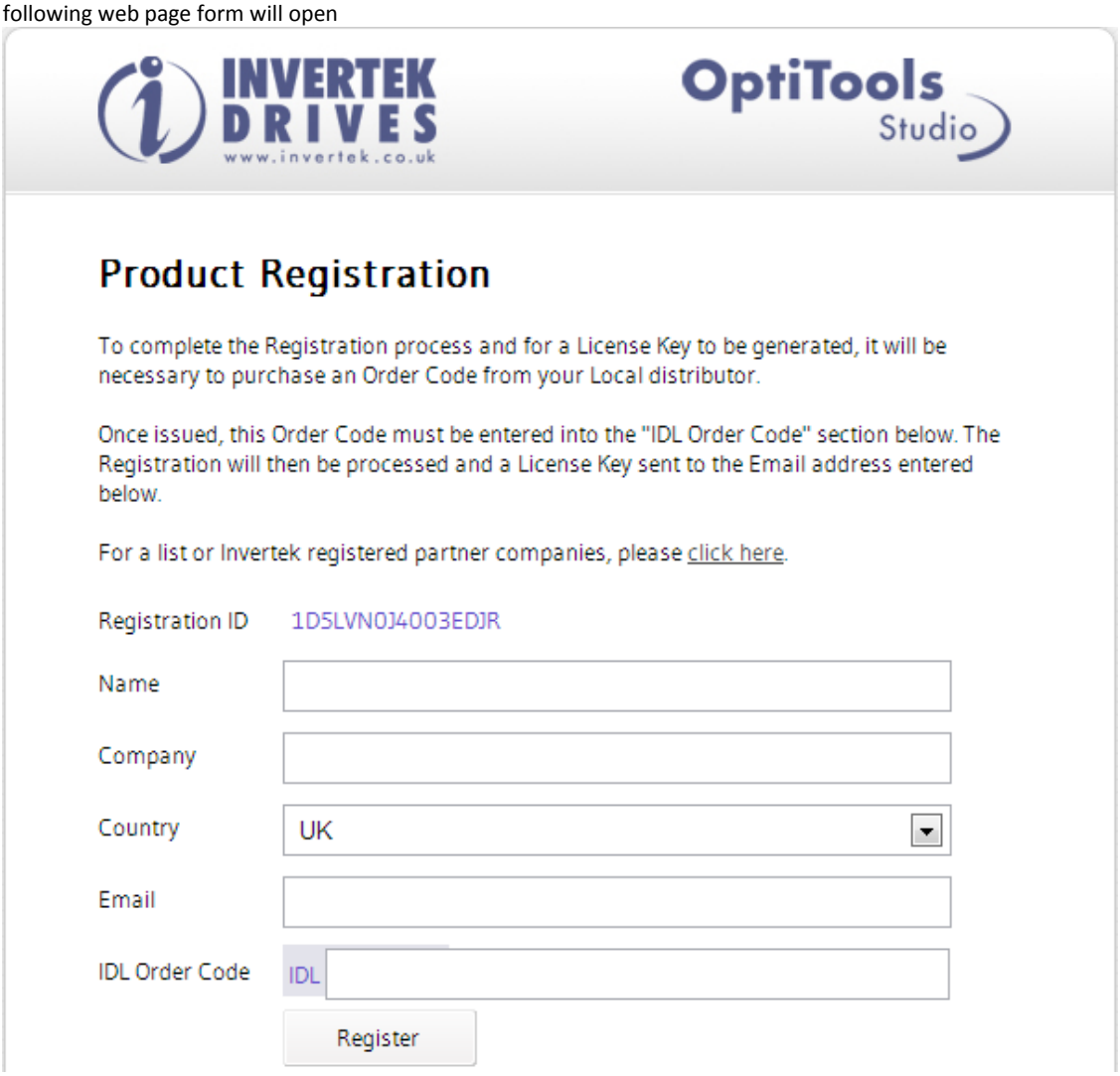
Full licences have no expiry period, and allow unrestricted use of the PLC programming function.

To register your licence, you will need the 6 digit code from your IDL Order Code from your Invertek Sales partner

- A working internet connection is required throughout the process
- Open the OptiTools Studio Software
- Click on Help > Activate Licence
- The following box is displayed



- Tick the “Request Evaluation Licence” box as highlighted, and then click register.
- The following web page form will open



- On the web form, fill out the required details
- In the **IDL Order Code** field, add the code from your Invertek Sales Partner.
- Click Register.
- You will receive an email by return within 24 hours, containing you licence key as shown below

Your OptiTools Studio License key has now been generated.
To activate OptiTools Studio on your PC, click on "Help" in the main menu, then on "Activate License..."
Please copy the License Key below into the empty License Key activation window and click "Activate".

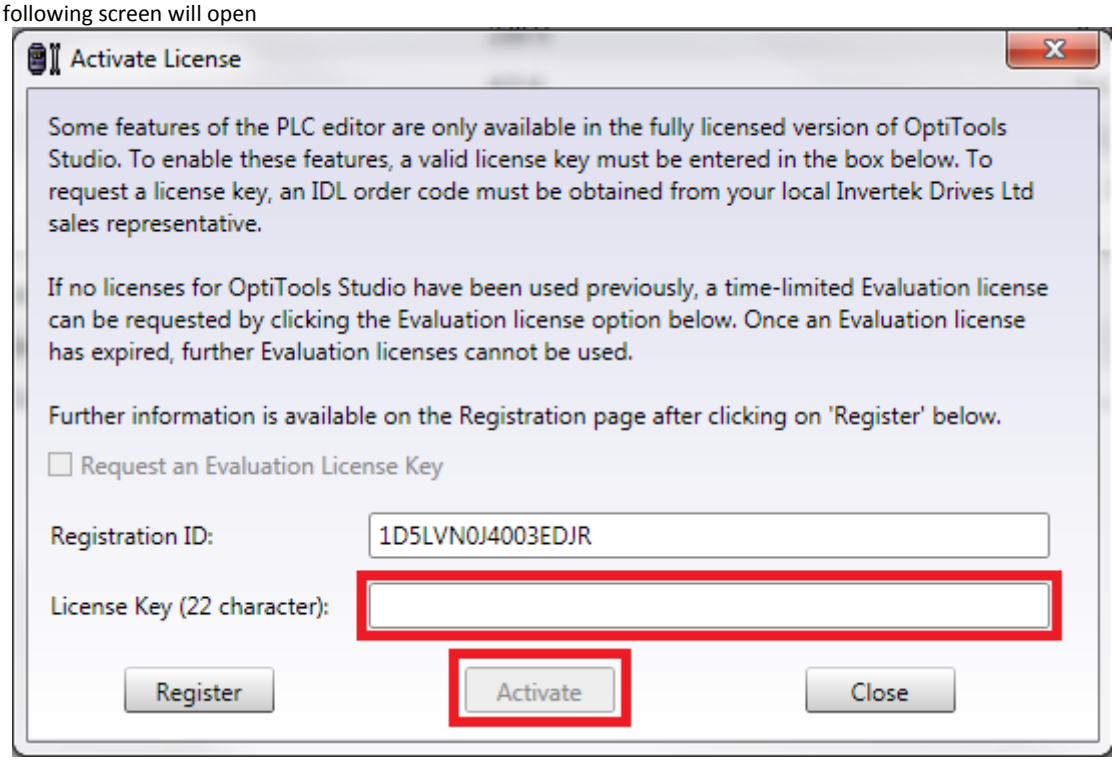
License Key: 4E64OV7XK0F2ZRRCB54AK8

Name: A. Customer
Company: A. Customer
Country: UK

Order Code: Evaluation

Email: a.customer@customeremail.com

- Copy the licence key, highlighted in **yellow** above
- In the OptiTools Studio software, on the toolbar, click on **Help** > **Activate Licence**
- The following screen will open



- Paste the licence key into the box shown
- Click on **Activate**. The licence is now activated

2. Connecting to Drives

2.1. Overview

Optitools Studio supports connection to the following drives

- Optidrive E2
- Optidrive E2 Single Phase Output
- Optidrive P2
- Optidrive HVAC
- Optidrive HVAC Eco
- Optidrive Elevator

Connection can be made directly to a single drive, or to a network of drives. When connecting to a network of drives, each drive must have a unique address assigned in the relevant parameter (P-36 or P5-01). This address assignment must be carried out through the drive keypad. All other settings are then possible in Optitools Studio.

Note that when carrying out firmware updates on supported drives, connection should be made to a single drive only.

2.2. Supported Connection Methods

Connection can be made using one of the following two methods

- RS485 Connection, using the PC Connection Kit OD-485AD-IN
- Bluetooth Connection using the Optistick OPT-2-STICK-IN

The correct connection method and COM port must be selected in the Tools menu to allow communication.

Tools > Select Communication Type >

- Bluetooth or RS485 can be selected here

Tools > Select COM Port >

- Select the required COM Port Number

All available COM ports are displayed

3. Parameter Editor

3.1. Overview

The Parameter Editor appears as follows.

These sections are explained further below

3.2. Toolbar

The toolbar buttons have the following functions

Button	Function
	Selects the Parameter Editor
	Select the Drive Monitor
	Selects the Function Block Editor
	Selects the Scope / Datalogger
	Load a parameter set file
	Save a parameter file
	Copy the parameter set to a new drive
	Reset the drive parameters to factory settings
	Print the parameter set
	Transfer the parameters from a drive
	Write the parameter file to a drive
	Transfer the parameters from an Optistick
	Write the parameter file to the Optistick

3.3. Parameter Group List

When Extended or Advanced parameter access is enabled in a drive, the various parameter groups can be viewed by clicking on the respective tab.

3.4. Parameter Data Area

This area displays a list of the respective drive parameters

ID	Description	Value	Range	Default
P2-01	Levelling Speed (Preset / Jog Frequency / Speed 1)	5.0 Hz	0.0 ... 50.0 Hz	5.0 Hz
P2-02	Run Speed (Preset / Jog Frequency / Speed 2)	50.0 Hz	0.0 ... 50.0 Hz	50.0 Hz
P2-03	Intermediate Speed (Preset / Jog Frequency / Speed 3)	25.0 Hz	0.0 ... 50.0 Hz	25.0 Hz
P2-04	Inspection Speed (Preset / Jog Frequency / Speed 4)	5.0 Hz	0.0 ... 50.0 Hz	5.0 Hz
P2-05	Rescue Mode Speed (Preset / Jog Frequency / Speed 5)	5.0 Hz	0.0 ... 50.0 Hz	5.0 Hz
P2-09	Skip Frequency Centre Point	0.0 Hz	0.0 ... 50.0 Hz	0.0 Hz
P2-10	Skip Frequency Band Width	0.0 Hz	0.0 ... 50.0 Hz	0.0 Hz
P2-11	Analog Output 1 Function Select	1: Drive Healthy		1: Drive Healthy
P2-12	Analog Output 1 Format	0: 0-10V		0: 0-10V
P2-13	Analog Output 2 Function Select	0: Drive Running		0: Drive Running
P2-14	Analog Output 2 (Terminal 11) Format	0: 0-10V		0: 0-10V
P2-15	User Relay 1 Output Function select	8: Motor contactor control		8: Motor contactor control
P2-16	User Relay 1 Upper Limit	100.0 %	0.0 ... 200.0 %	100.0 %
P2-17	User Relay 1 Lower Limit	0.0 %	0.0 ... 200.0 %	0.0 %
P2-21	Display Scaling Factor	0.000	-30.000 ... 30.000	0.000
P2-22	Display Scaling Source	0: Motor Speed		0: Motor Speed
P2-24	Effective Switching Frequency	1: 8kHz		1: 8kHz
P2-25	Fast Deceleration Ramp Time	0.00 s	0.00 ... 240 s	0.00 s
P2-30	Analog Input 1 Format	0: 0 - 10V		0: 0 - 10V
P2-31	Analog Input 1 Scaling	100.0 %	0.0 ... 500.0 %	100.0 %
P2-32	Analog Input 1 Offset	0.0 %	-500.0 ... 500.0 %	0.0 %
P2-33	Analog Input 2 Format	0: 0 - 10V		0: 0 - 10V
P2-34	Analog Input 2 Scaling	100.0 %	0.0 ... 500.0 %	100.0 %

ID: The number or reference of the parameter
Description: The parameter name or title. Hovering the mouse pointer over this will show the full descriptive text
Value: The value of the parameter, and the units. This value may be edited directly
Range: The possible setting range of the parameter
Default: The factory default setting of the parameter
When parameter values are non-default, they are highlighted in blue font.

3.5. Online / Offline Status Indication

Displays the Online / Offline status of the parameter Editor.

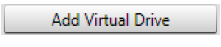
Offline Mode	When working offline, changes are made in the PC memory, and the parameter file may be saved to disk. In this mode, the drives are „Virtual“ and may not exist except in the PC software
Drives in Network	Following a scan of the connected drive network, the available connected drives are displayed. Changes made in the parameters are made on the PC only, and are not transferred automatically to the drive. The user must manual carry out a transfer if required.
Real-Time Edit Mode	In Real Time Edit Mode, changes are made in Real time directly in the drive memory, and may also be saved to disk.

3.6. Drive Data

Displays information about the Virtual or connected drive as follows

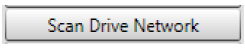
FS 2 P2 230V 1~ 0.75kW 01 *Drive Description*	FS : Frame Size, Drive Type
	Supply Voltage, Input Phases, Power Rating & Type (kW / HP)
	Drive Address, Drive Description Text

3.7. Add Virtual Drive



Button used to add a virtual drive, or to create a specific parameter file when working offline.

3.8. Scan Drive Network

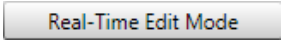


Button used to scan for connected drive(s) and insert them into the project area

3.9. Network Scan Control

This section allows the user to limit the range of drive addresses which are scanned for. Each address is checked sequentially starting from 1, and the maximum range is 63. Restricting the address range speeds up the process of finding drives online.

3.10. Real Time Edit Mode



Selects the Real Time Edit Mode

4. Drive Control / Monitor

4.1. Introduction

The Drive Control / Monitor function provides the ability to

- Monitor various drive parameters and signals in real time
- View values in multiple drives simultaneously
- Take control of a drive directly from the PC software

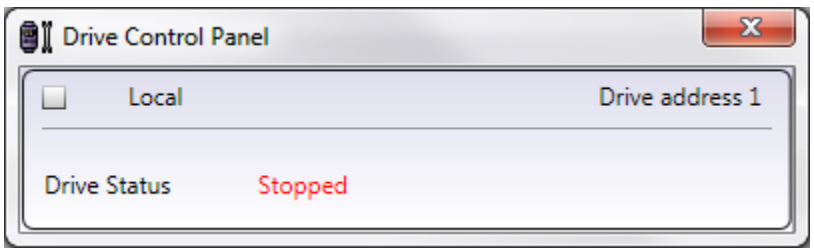
The Drive Control / Monitor appears as follows.

4.2. Toolbar

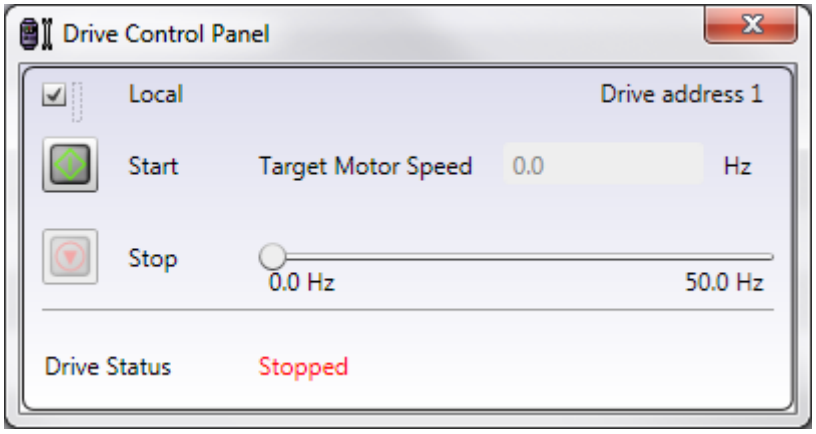
The toolbar functions are essentially the same as the parameter editor, except for the following additional functions

Button	Function
	Enables Real Time Monitor Mode, where values are displayed in real time from connected drives
	Disables Real Time Monitor Mode
	Displays the Drive Control Panel

Clicking on the button to display the Drive Control Panel whilst Offline brings up the following Control Panel



The drive status only is displayed here. Ticking the Local selection box allows the user to take direct control over the drive. The Control Panel Display changes to the following.



The user can now directly Start and Stop the drive from the PC software, and select the required speed.
Warning: Ensure operation is safe before using this function

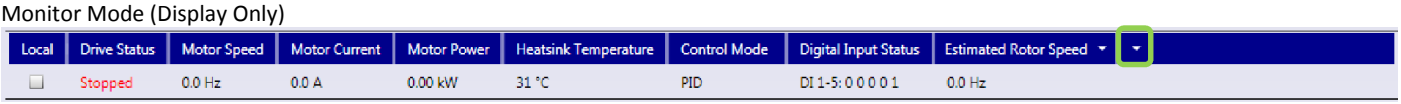
4.3. Drive Network Information Area

This screen area is as described in the previous sections for the Parameter Editor Function.

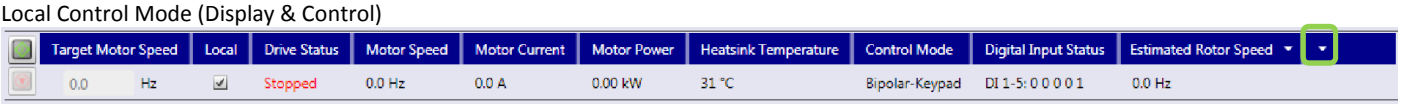
4.4. Drive Monitor Bar

The Drive Monitor bar appears as follows.

Monitor Mode (Display Only)



Local Control Mode (Display & Control)



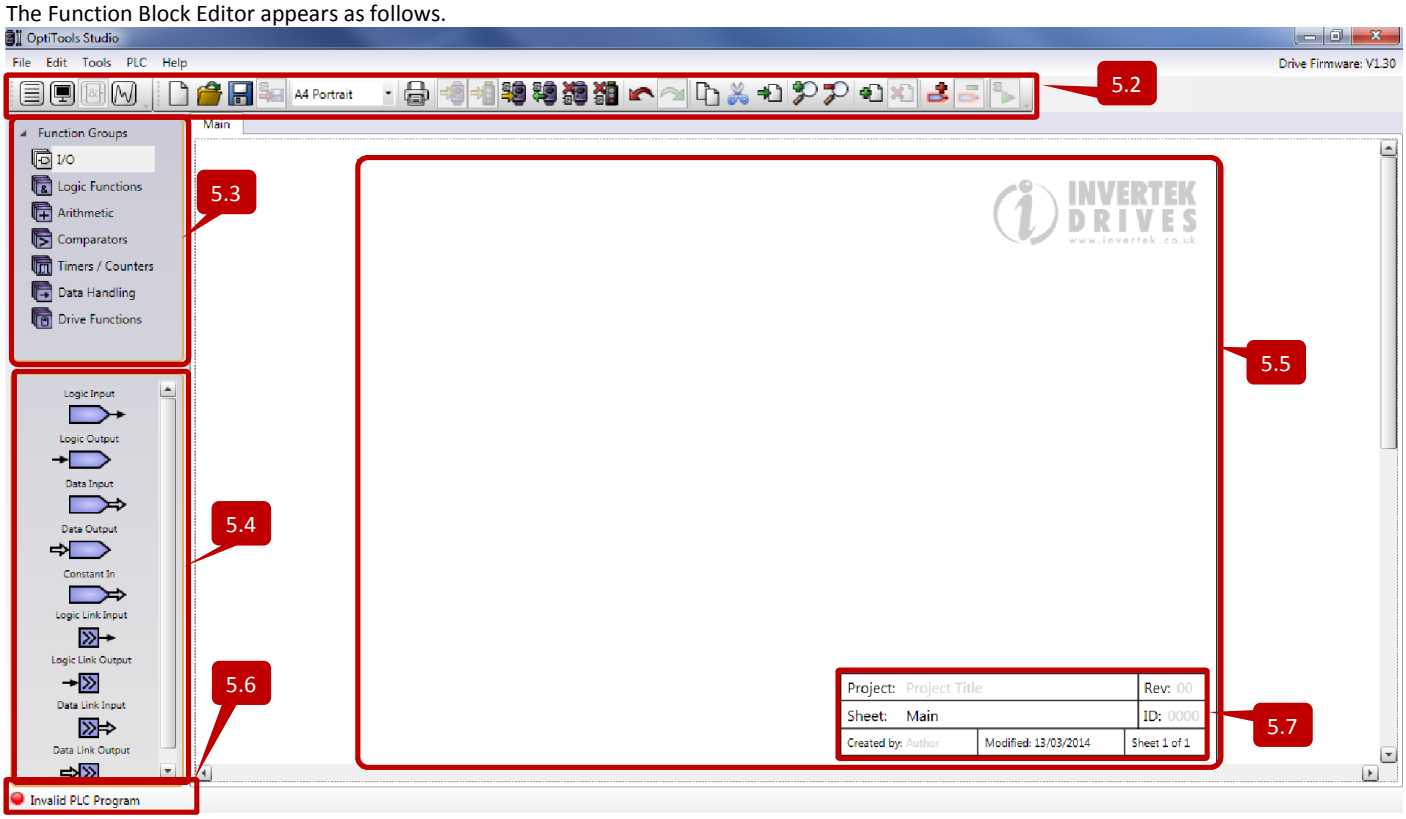
The displayed values are as follows (Note: Not all functions are supported for all drive types)
Drive Status: Status of the connected drive, e.g. Stopped, Running, Tripped
Motor Speed: Output Frequency or Speed
Motor Current: Motor Load Current in Amps
Motor Power: Motor consumed power in kW
Heatsink Temperature: Temperature of the drive heatsink in °C
Control Mode: The selected control mode in P-12 / P1-12
Digital Input Status: Status of the drive digital inputs
Other values may be selected and displayed, by clicking the arrow (highlighted in the green square above) and selecting the required option to add the value to the screen.

5. Function Block Editor

5.1. Overview

This section describes the how the function block editor works, and which parameters should be changed within the drive to ensure correct operation.

The Function Block Editor appears as follows.



5.2. The Toolbar

Button	Function
	Selects the Parameter Editor
	Select the Drive Monitor
	Selects the Function Block Editor
	Selects the Scope / Datalogger
	Creates a New Function Block Program
	Loads an existing Function Block Program (.plc)
	Saves the current Function Block Program (.plc)
	Saves the current Function Block Program in compiled (.pdg) format
	Prints the current Function Block Program
	Transfer the current Function Block Program to a connected drive
	Transfer the current Function Block Program to a connected Optistick
	Opens a compiled (.pdg) format Function Block Program, and transfers to a connected drive
	Reads the Function Block Program from the drive memory in compiled (.pdg) format. Note : Only possible of the program is not protected
	Erases the Function Block Program from the connected drive

Button	Function
	Erases a Function Block Program from the Optistick memory
	Undo the last action
	Redo the last action
	Copy the selected Function Blocks
	Cut the Selected Function Blocks
	Paste the selected Function Blocks
	Zoom In
	Zoom Out
	Add a new Page to the project
	Delete a page from the project
	Highlight the errors
	Hide the errors
	Open the simulator

5.3. Function Block Groups Windows

This section shows the different groups of function blocks that are available. Clicking on a group opens the list of available function blocks in the window below.

5.4. Function Blocks Window

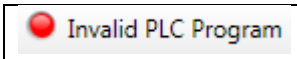
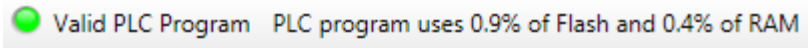
The function blocks windows lists the available blocks in each group.

5.5. The Workspace

Function Block Programs are created in the workspace area. The area page size can be changed using the page size selection on the tool bar, and multiple pages may be added or removed using the toolbar buttons.

5.6. Program Status

This section indicates the status of the program as follows

	The created program is not valid and contains errors. It may be saved, but may not be transferred to a drive or Optistick. Use the Highlight Errors function to display the errors preventing compiling and transferred.
	The created program contains no errors, and may be saved or transferred to a drive or Optistick. The Flash memory and RAM memory required by the program are displayed.

5.7. User Project Data

This section allows the user to enter data regarding the project, and maintain version control.

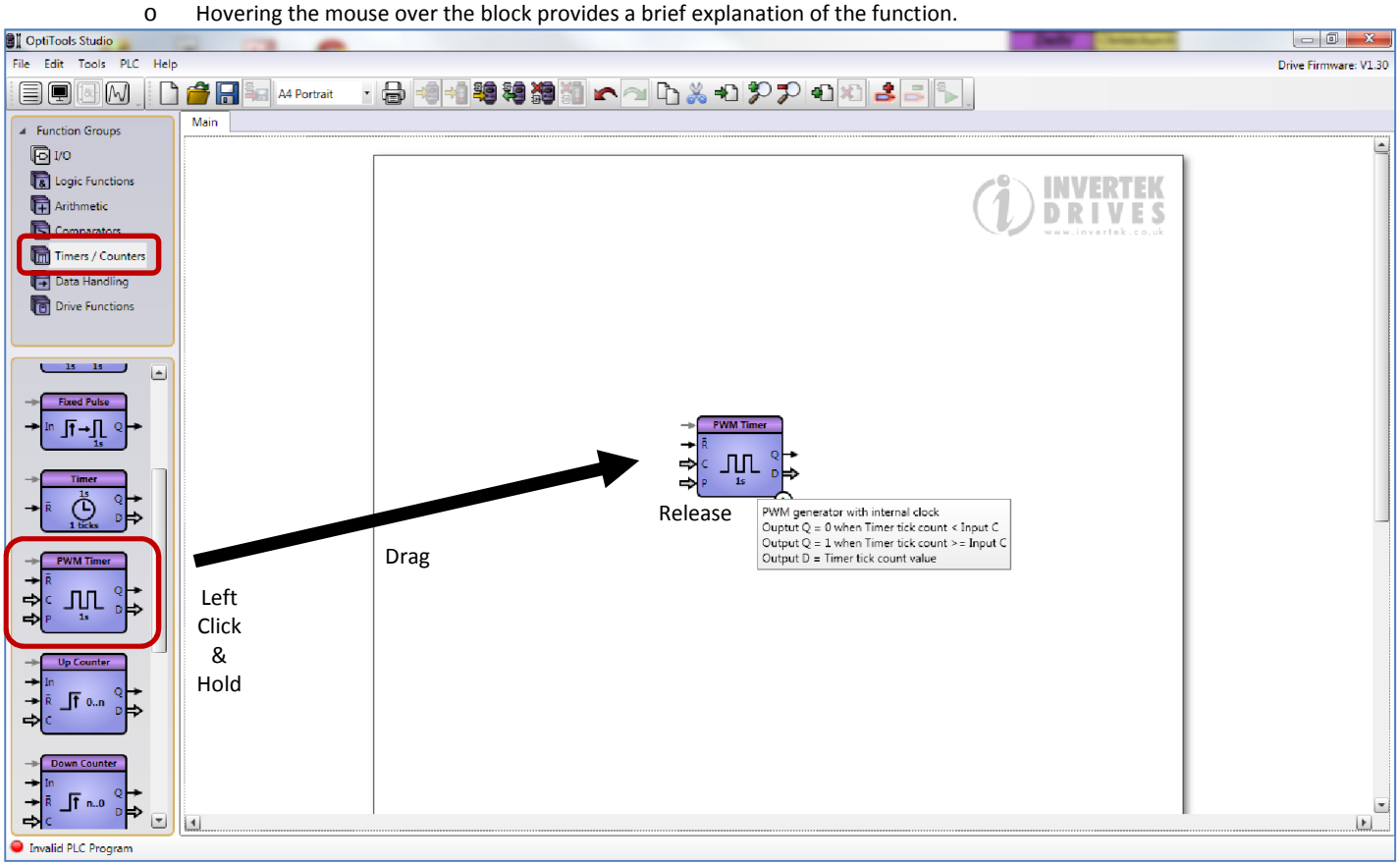
5.8. Creating a Program

Programs are created by selecting from the library of available Function Groups, dragging and dropping these into the worksheet and connecting them up. When complete, the program may be simulated before uploading to a drive (online monitoring is not possible). It will always be necessary to set some parameters within the drive to ensure the programme functions correctly, this is explained more fully later in this chapter.

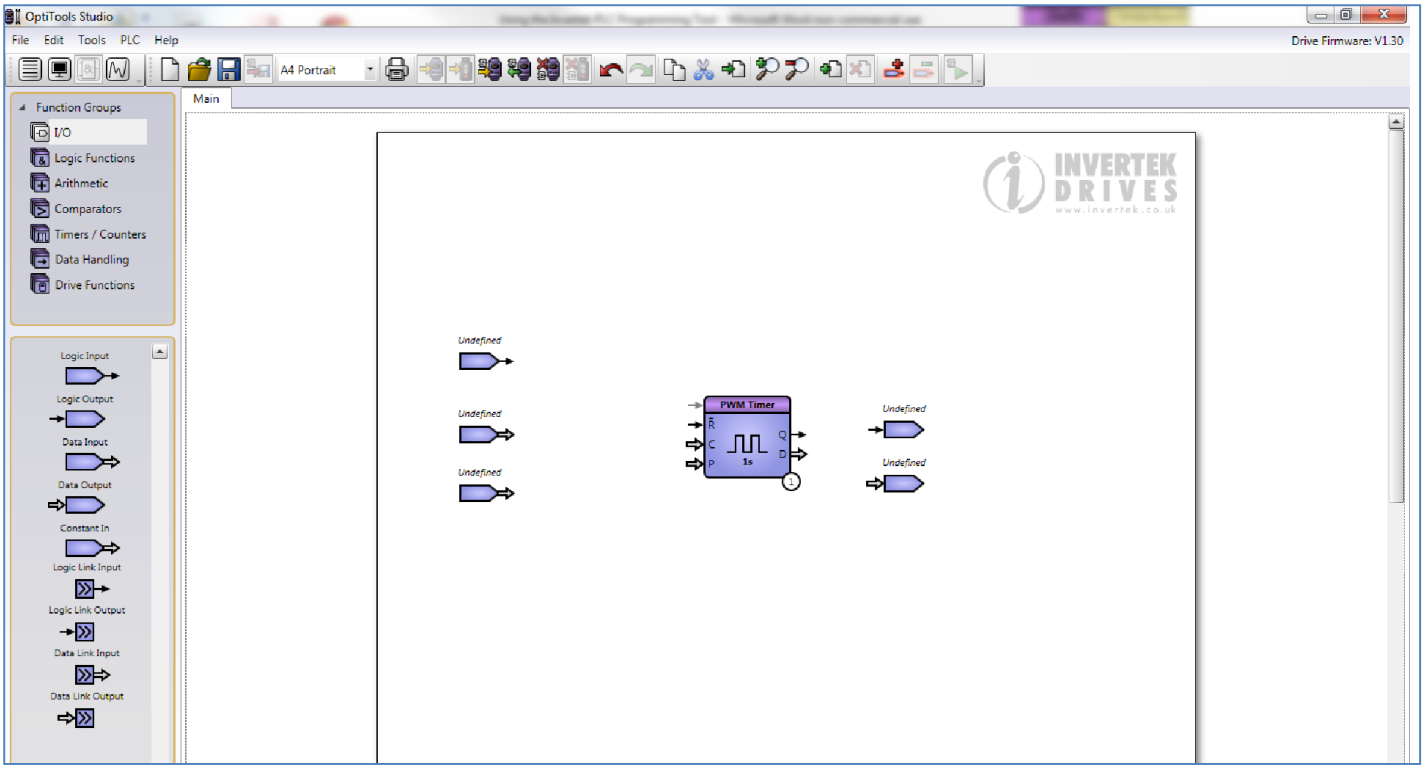
5.8.1. Working in the Worksheet

The following is a step by step example of creating a PLC program. The function is not important at this stage; the example is intended to provide guidance on how the Function Block Editor operates.

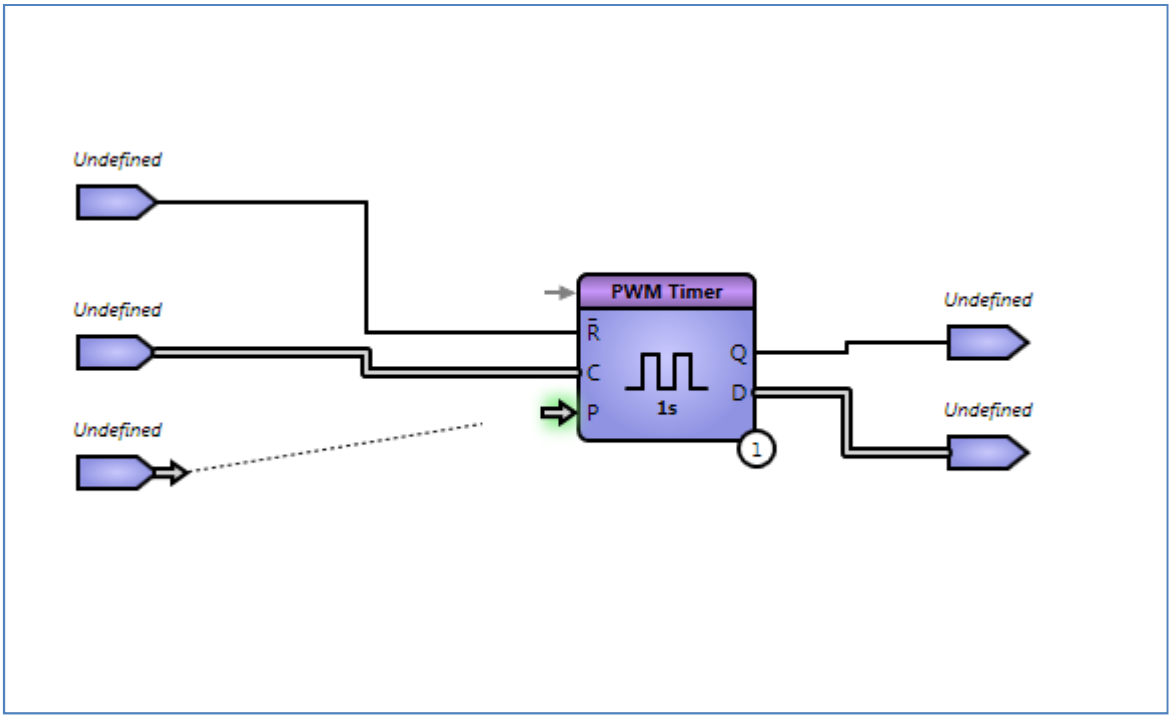
- In the Function Block Editor > Function Groups, select Timers / Counters
- Drag and drop a PWM Timer block into the worksheet
  - Hovering the mouse over the block provides a brief explanation of the function.



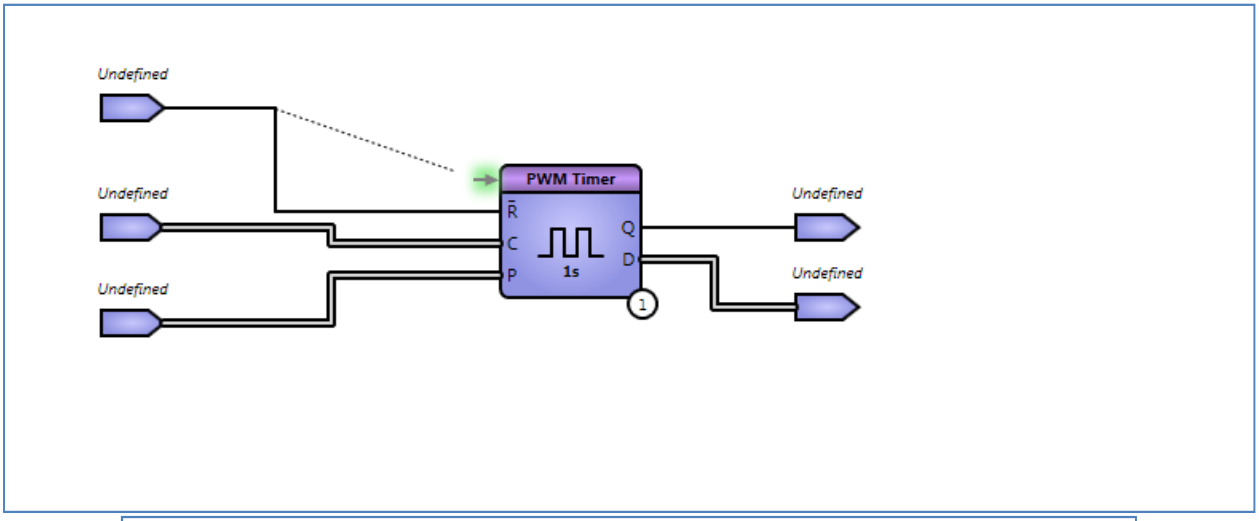
- Click on the I/O Function Group
- From the I/O section, drag and drop the following items into the workspace
  - A Logic Input
  - Two Data Inputs
  - A Logic output
  - A Data
- Note that the logic inputs and outputs are connected using thin arrows, whilst the data inputs and outputs use thick arrows. Logic signals must be either 1 or 0 (TRUE or FALSE), while data can be 12 bit, percentage or other values. Logic and data cannot be directly connected, hence the different arrows.

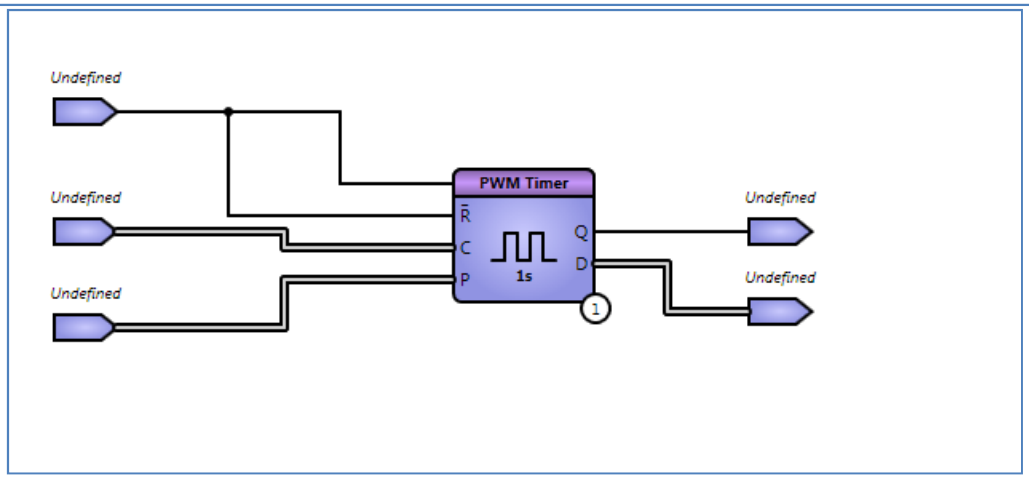


- The inputs and outputs can now be connected to the function block.
- Simply click on the arrow, and drag to the required block connection point, then release.
  - Possible connections are highlighted in green as shown below.



- For additional links from any node, *hold down the shift key* and drag from the node as before.

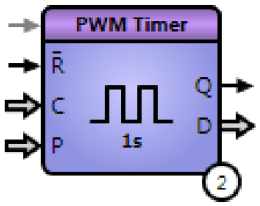




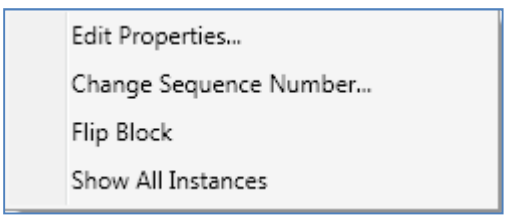
- Any line or function block me may be selected and dragged at any time
- Multiple blocks and lines may be selected with the mouse and moved simultaneously
- The toolbar buttons and standard windows short cuts allow cut, copy and paste actions

5.8.2. Function Blocks Overview

Each different function block has a unique graphical representation.



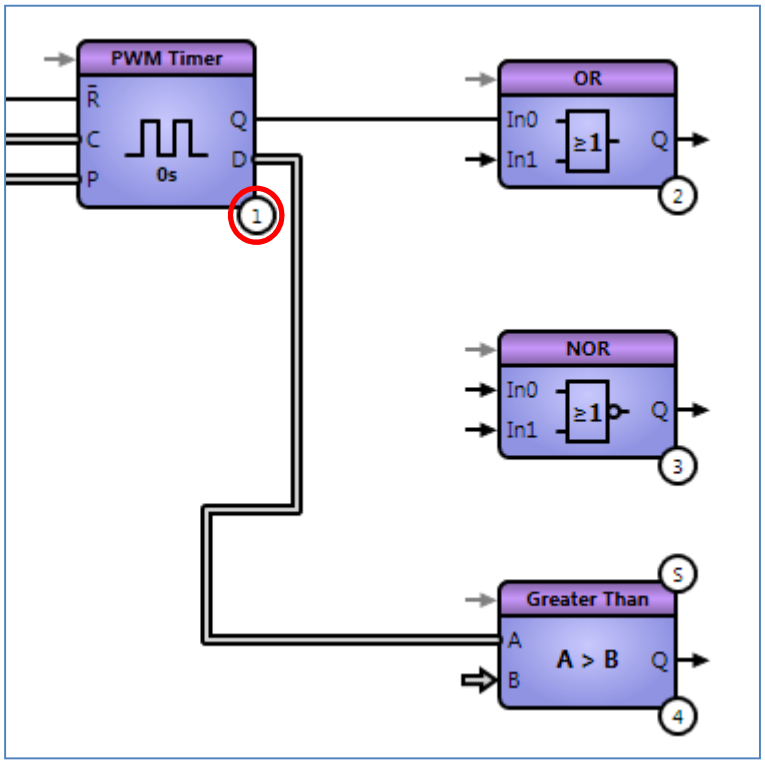
- Arrows pointing towards the block are inputs
  - Narrow arrows are logic inputs
  - Wide arrows are data inputs
- Arrows pointing away from the block are outputs
  - Narrow arrows are logic inputs
  - Wide arrows are data inputs
- The grey arrow at the top left is the Enable Block input, which allows blocks to selectively be enabled or disabled
  - When left disconnected, the block is always enabled
- Some function blocks have additional properties (refer to the descriptions later in this section). Right clicking on the block brings up the context menu shown below

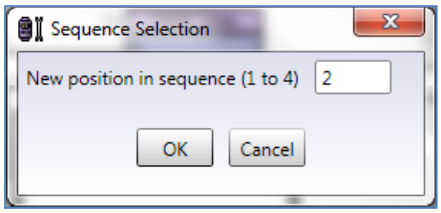


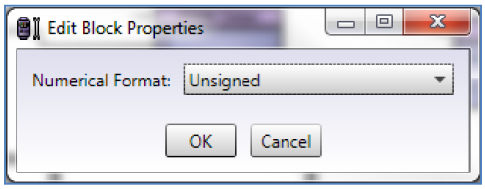
  - Edit Properties – Edits properties of the block specific to each function block, refer to later block descriptions

- Change Sequence Number - The sequence number refers to the order in which the functions are executed, which can affect program operation. Change here, or by double clicking on the sequence number at the bottom right of the function block.
- Flip Block – flips the block horizontally, which can help when laying out complex sequences
- Show All Instances – Highlights all instances where this particular block or I/O point is used in the program

· Each function block has a sequence number, which defines the order in which the blocks are executed. The number is shown in the bottom right corner of the block.

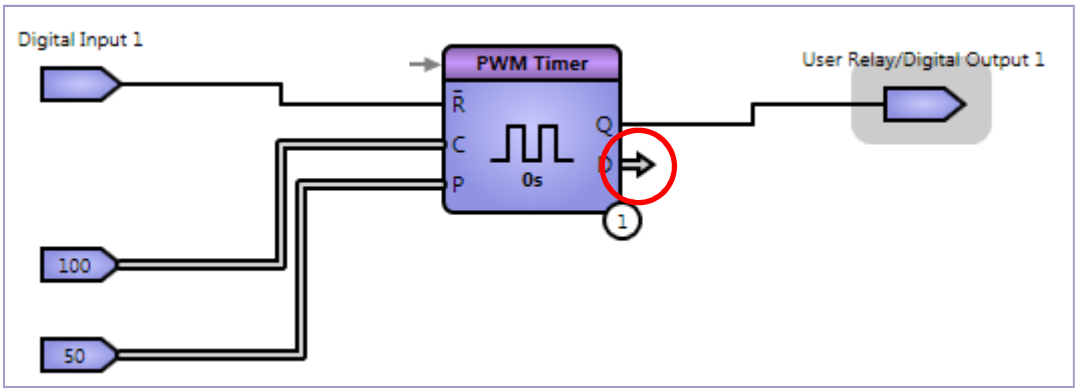


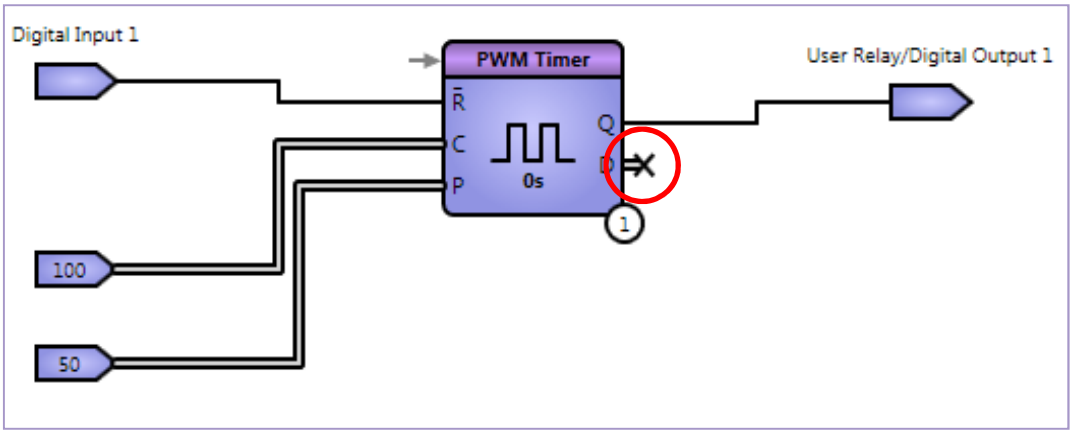




5.8.3. Terminating unused Outputs

If any block outputs are not required, they must be terminated to allow the program to be successfully compiled. To terminate an output, right mouse click on it.



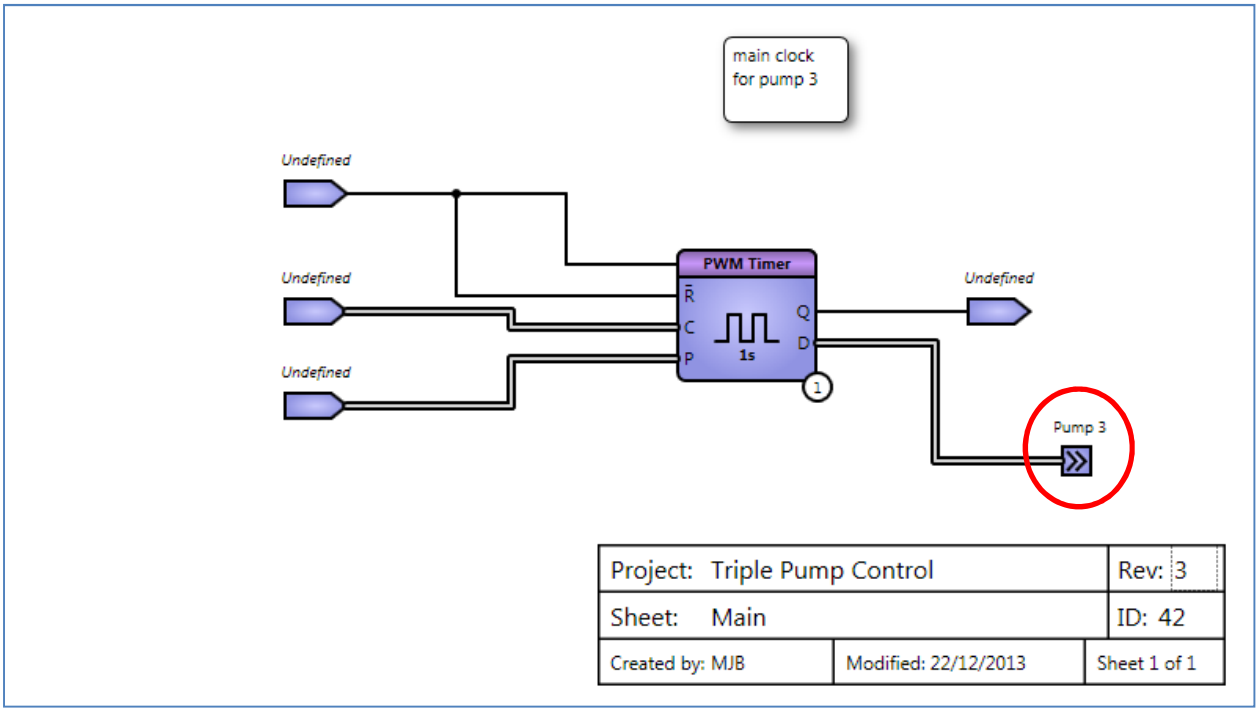


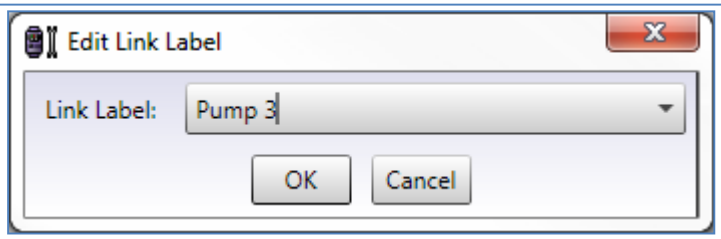
5.8.4. Comments

It is possible to add comments on the worksheet by right clicking in an empty space and selecting Add Comment.

5.8.5. Using Logic and Data Links

Logic and Data links allow signals to be linked across multiple sheets or areas of the program without direct connections. Both Logic and Data Links have a single output, which must be uniquely named. The respective link inputs may then be used multiple times at different points in a program. Links can be added to other sheets (see below) by selecting a link from the I/O family connecting it and naming it by double clicking on it. It will then automatically connect to other links (on the same or other sheets) of the same name.

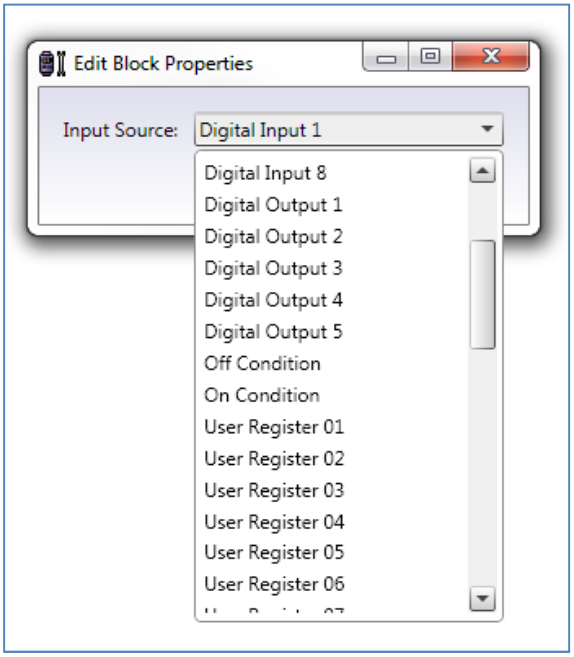




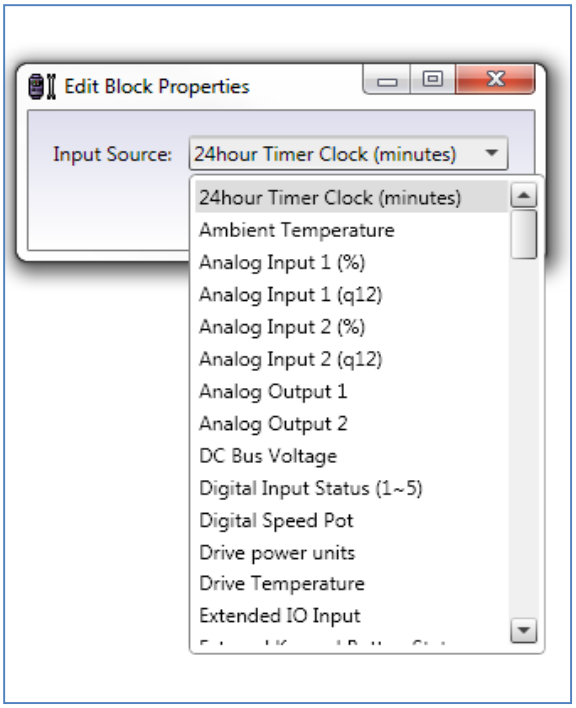
Where a larger program is required, additional worksheets can be added at any time using the button on the toolbar

5.8.6. Selecting Logic and Data Input Sources

A programme is not complete until we have added the necessary connections that suit the functions of the drive. That is, each input or output must be connected to a suitable function within the drive. Double clicking on an I/O component will bring up a box with a drop down list of all possible connections. For example, a logic input will offer a long list of possibilities:

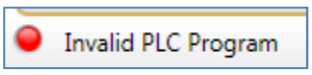


A data input has a similar selection:

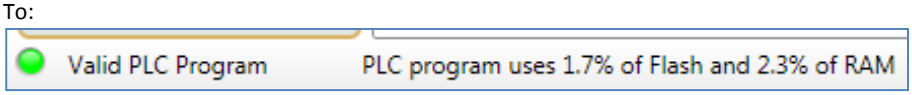


Some of the inputs are the inputs of the drive itself; digital inputs for logic inputs, analog inputs for data, but some are simply registers for use in the programme, or other functions within the drive. How these can be used will be explained later.

Once all the inputs and outputs have been allocated, if the programme is complete, the indicator at the bottom left of the screen with change from:



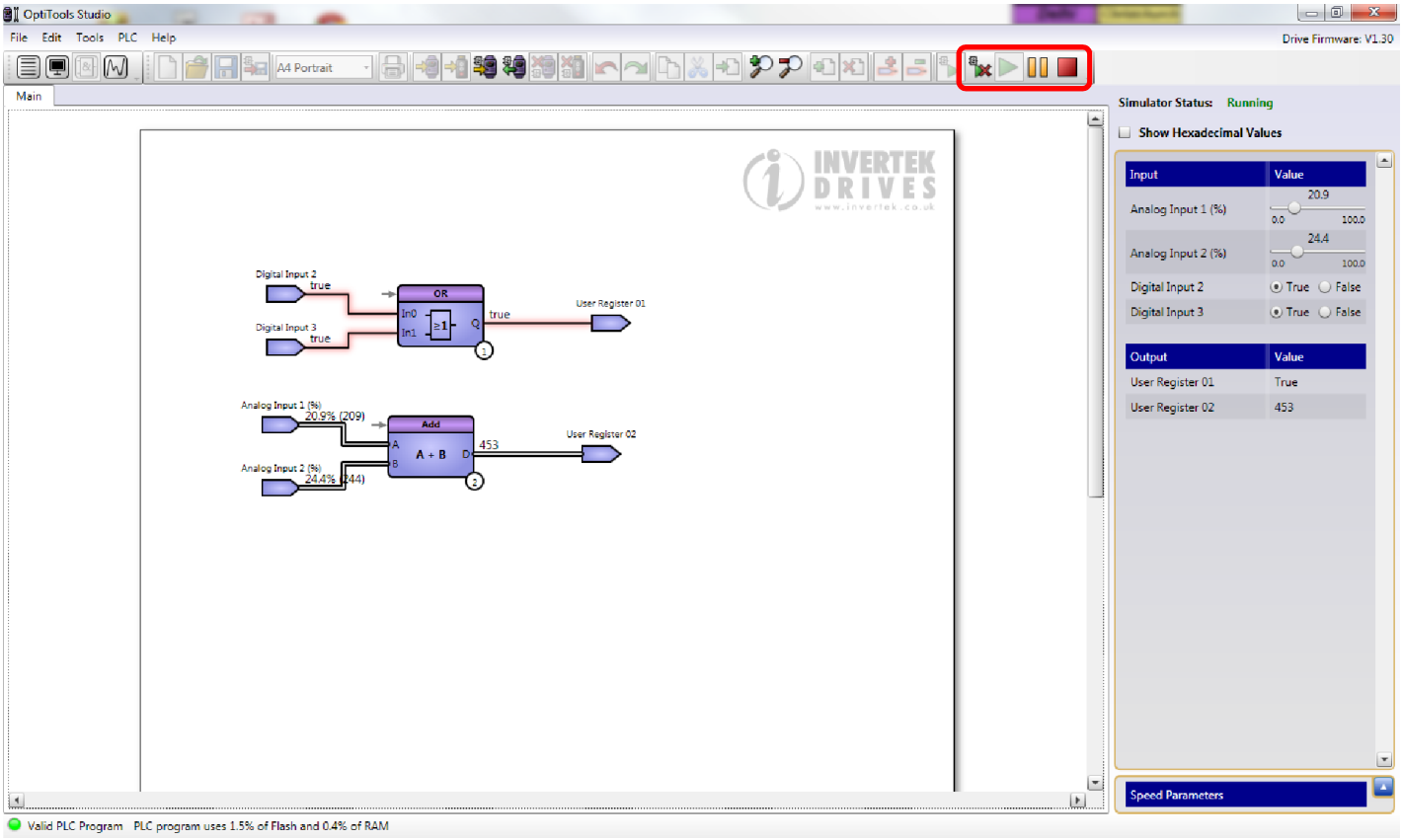
To:



As mentioned earlier, the amount of available memory is also shown.

The programme can now be simulated.

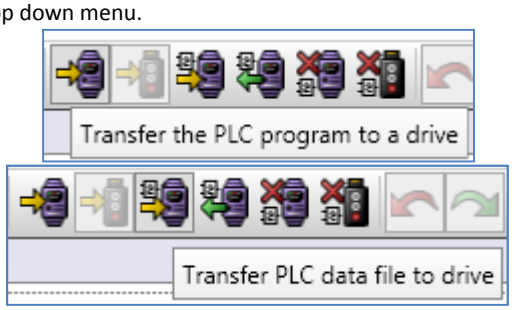
5.9. Simulation



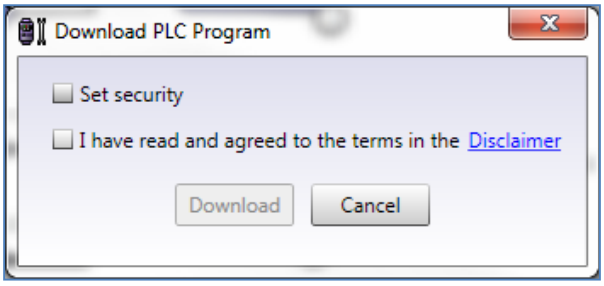
In the above example, simulation has been selected using the green arrow at the right of the toolbar, and is controlled (stopped, paused, started) using similar buttons. Remember to start the simulation with the large green arrow or nothing will happen! The control screen on the right automatically displays the inputs and outputs, with buttons and drag bars to change values where appropriate. You can set the numbers more accurately by clicking on them and typing in the values, or using left and right arrows on the keyboard to increment buttons left or right.
In this example, both digital inputs to the OR gate are high, so User register 01 is TRUE. The analog inputs are added and the result in User register (45.3%) is displayed.
For the programme to work, the registers must be connected to functions within the drive. This is done by setting parameters within the drive.

5.10. Uploading

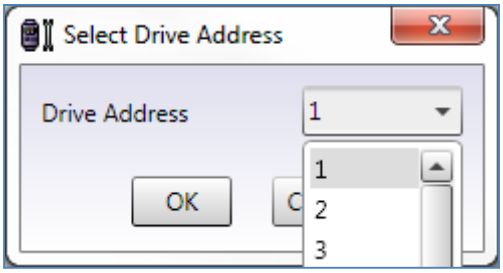
If the simulation shows that the programme is suitable for testing within the drive, the programme can now be uploaded (provided you have a Licence) for further evaluation. To do this, connect a suitable drive to the PC using either a USB to RS485/RJ45 interface, or an Optistick and Bluetooth connection as described earlier.
Check the connection using the Parameter Editor in OTS.
Now you can upload using the buttons or drop down menu.



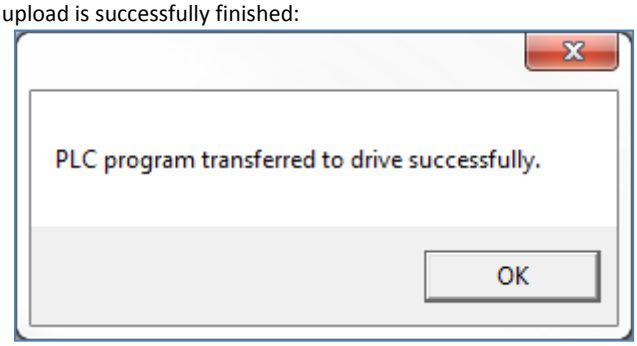
If the programme is uploaded as a data file, it cannot be viewed or edited by downloading from the drive, only copied.
When the file is uploaded you will have to check a box indicating you have read the Disclaimer. This is simply to ensure Invertek are not responsible for faults, damage or injury etc. caused by the programme. You will also be able to set a security bit, which will prevent the PLC programme from being downloaded or copied. Use this if you are sure you never want to copy or edit the programme from the drive. If you wish to copy the programme and parameters from the drive, but wish to ensure no one can read the programme, upload the programme as a data file as described above.



The programme will prompt you for a drive address. By default this will be 1, but if there are several drives on the same buss, you may have changed it by setting P5-01 to another value.



When you upload the programme the upload progress will be indicated in the bottom right hand corner of the screen. At the end of the upload the programme will report the upload is successfully finished:



And the drive display will briefly flash **dL – PLC**.

You can also upload the programme to an Optistick, but there are certain limitations at present. The Optistick should contain a parameter set that matches the drive in power, type and brand. This is subject to change.

You can also copy programmes and parameter sets from a drive to an Optistick, providing the security bit has not been set

5.11. Parameter settings

It is now necessary to set parameters to ensure the programme functions correctly within the drive.

5.11.1. Enabling the Programme.

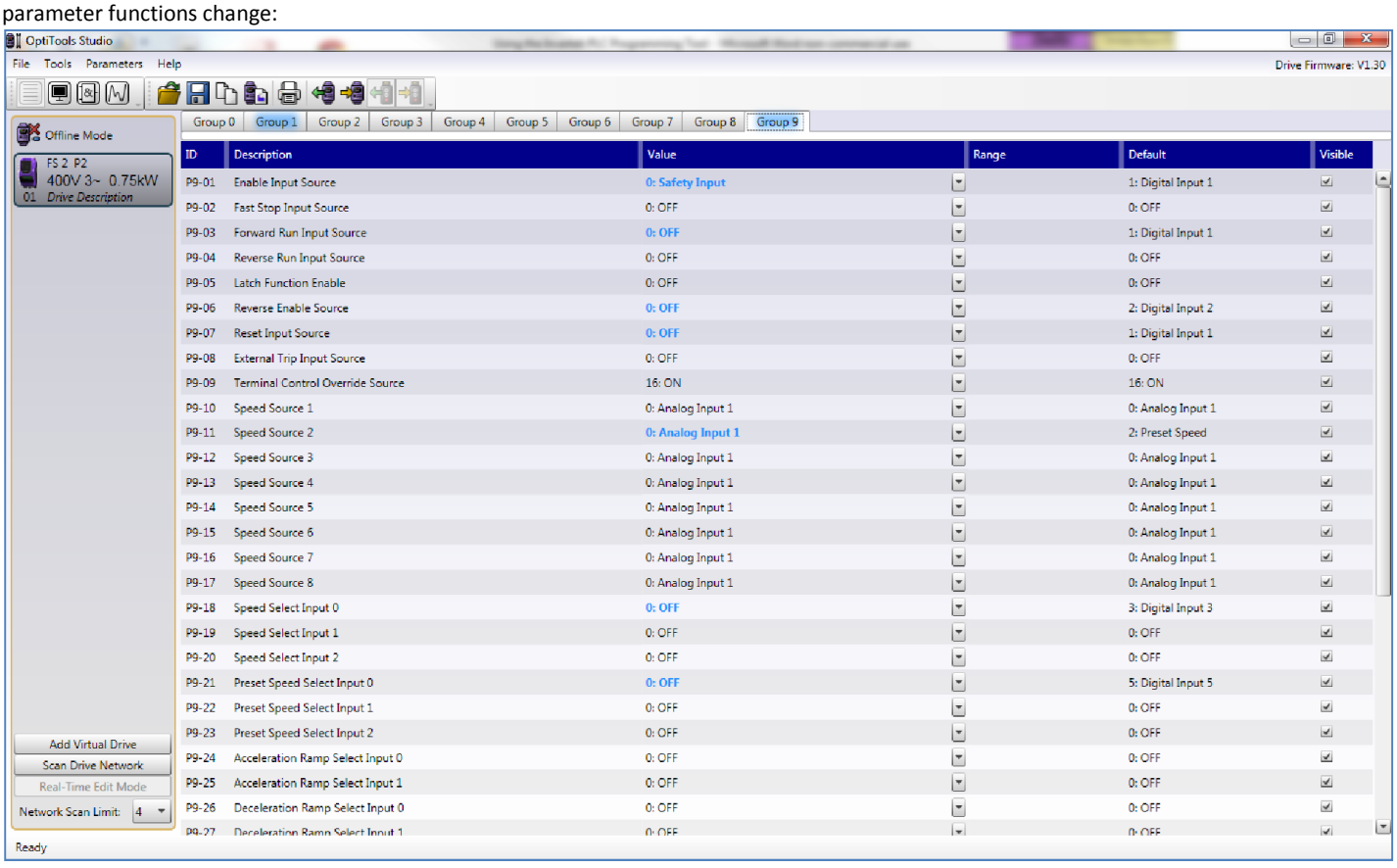
The programme in the drive must always be enabled by setting Parameter 6-10 to 1. To do this you'll need to set Parameter P1-14 = 201 to access the advanced parameters.

5.11.2. Making Internal Connections

As stated earlier, the inputs and the outputs of the PLC programme must be connected to functions within the drive for the programme to work. Some of these functions will be inputs and outputs to the drive such as digital inputs, analog inputs and outputs, relays, displays etc. Some will be functions that are internally connected, such as PID values, ramp settings, and other internal functions. There are also internal registers (see above) and parameters that can be accessed. Many of these functions are already used within the drive, and are therefore connected. For example, analogue input 1 is connected to Speed Source 1, and digital input 2 is connected to the Reverse Enable Source.
In order for the programme to work, it will be necessary to 'disconnect' some of these functions and reconnect them to the appropriate inputs and outputs in the programme.
You can see the existing internal connections by looking at group 9 parameters, most conveniently in the Parameter Editor of OTS.

ID	Description	Value	Range	Default	Visible
P9-01	Enable Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-02	Fast Stop Input Source	0: OFF		0: OFF	☑
P9-03	Forward Run Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-04	Reverse Run Input Source	0: OFF		0: OFF	☑
P9-05	Latch Function Enable	0: OFF		0: OFF	☑
P9-06	Reverse Enable Source	2: Digital Input 2		2: Digital Input 2	☑
P9-07	Reset Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-08	External Trip Input Source	0: OFF		0: OFF	☑
P9-09	Terminal Control Override Source	16: ON		16: ON	☑
P9-10	Speed Source 1	0: Analog Input 1		0: Analog Input 1	☑
P9-11	Speed Source 2	2: Preset Speed		2: Preset Speed	☑
P9-12	Speed Source 3	0: Analog Input 1		0: Analog Input 1	☑
P9-13	Speed Source 4	0: Analog Input 1		0: Analog Input 1	☑
P9-14	Speed Source 5	0: Analog Input 1		0: Analog Input 1	☑
P9-15	Speed Source 6	0: Analog Input 1		0: Analog Input 1	☑

In order to 'disconnect' these internal connections, it is necessary to edit these parameters. This can only be done by setting Parameter P1-13 to 0, effectively removing the pre-defined functions in that parameter. With P1-13 set to 0, drop down options appear in group 9, and many parameter functions change:



ID	Description	Value	Range	Default	Visible
P9-01	Enable Input Source	0: Safety Input		1: Digital Input 1	☑
P9-02	Fast Stop Input Source	0: OFF		0: OFF	☑
P9-03	Forward Run Input Source	0: OFF		1: Digital Input 1	☑
P9-04	Reverse Run Input Source	0: OFF		0: OFF	☑
P9-05	Latch Function Enable	0: OFF		0: OFF	☑
P9-06	Reverse Enable Source	0: OFF		2: Digital Input 2	☑
P9-07	Reset Input Source	0: OFF		1: Digital Input 1	☑
P9-08	External Trip Input Source	0: OFF		0: OFF	☑
P9-09	Terminal Control Override Source	16: ON		16: ON	☑
P9-10	Speed Source 1	0: Analog Input 1		0: Analog Input 1	☑
P9-11	Speed Source 2	0: Analog Input 1		2: Preset Speed	☑
P9-12	Speed Source 3	0: Analog Input 1		0: Analog Input 1	☑
P9-13	Speed Source 4	0: Analog Input 1		0: Analog Input 1	☑
P9-14	Speed Source 5	0: Analog Input 1		0: Analog Input 1	☑
P9-15	Speed Source 6	0: Analog Input 1		0: Analog Input 1	☑
P9-16	Speed Source 7	0: Analog Input 1		0: Analog Input 1	☑
P9-17	Speed Source 8	0: Analog Input 1		0: Analog Input 1	☑
P9-18	Speed Select Input 0	0: OFF		3: Digital Input 3	☑
P9-19	Speed Select Input 1	0: OFF		0: OFF	☑
P9-20	Speed Select Input 2	0: OFF		0: OFF	☑
P9-21	Preset Speed Select Input 0	0: OFF		5: Digital Input 5	☑
P9-22	Preset Speed Select Input 1	0: OFF		0: OFF	☑
P9-23	Preset Speed Select Input 2	0: OFF		0: OFF	☑
P9-24	Acceleration Ramp Select Input 0	0: OFF		0: OFF	☑
P9-25	Acceleration Ramp Select Input 1	0: OFF		0: OFF	☑
P9-26	Deceleration Ramp Select Input 0	0: OFF		0: OFF	☑
P9-27	Deceleration Ramp Select Input 1	0: OFF		0: OFF	☑

In particular, the basic functions (run, enable, stop, reverse etc.) are disconnected from the digital inputs, and Speed Source 2 (on this HVAC drive) is also changed. The changes are highlighted in blue, and the defaults (i.e. former values) can be seen in the column to the right. If these

functions are required, it will be necessary to reconnect them either to logical (or previous) values, or to an input or output with the programme. The drop down box shows the possible values.
Note that digital input 1 was connected to the Enable input, the Run forward input and the Reset input. These functions must be connected to something (i.e. existing drive functions or outputs within a programme) in order for the drive to run and stop.
Note that the default settings vary between P2 and HVAC drives.
As it is always necessary to make some parameter changes to run a programme, it makes sense to use the Parameter Editor and the Function Block Editor together, saving the parameter list and the programme with the same name for example.

5.12. Some other features

5.12.1. Execution Order

We have already seen that the order of execution of the function blocks can be changed by editing the number at the bottom right of the function block.

5.12.2. Signed Numbers

As previously mentioned the letter at the top right of the block indicate whether the block is handling signed (S) or unsigned (U) numbers. Signed 16 bit numbers (15 bit + sign bit) are used by default, so numbers can be handled between -32767 to +32767. This is also the way numbers are stored in user registers. However, if you write a 16bit unsigned number to a register, although it will appear as a signed number, you can still read it as a 16 bit unsigned number.
Clearly some care is needed moving between signed and unsigned numbers.

5.12.3. Scaling of Numbers

Some function blocks allow the inclusion of a decimal point (by selecting the number of decimal places). This is only for display purposes – i.e. the decimal point will be shown in the display. It does not affect the way the number is handed within the programme; it is treated as though there were no decimal points.

5.12.4. 16 and 32 Bit Words

Most of the calculations that are carried out in the drive use 16 bit words. This means that for a multiplication function blocks two 16 bit words will be multiplied together resulting in 16 bit word, with a corresponding loss of accuracy. This is probably satisfactory for most operations, but loss of resolution can occur when dividing and then multiplying for example. To overcome this, some 32 bit function blocks are available, and will be described later. Also provided are some multiply then divide function blocks that produce a 32 bit number (following the multiplication) internally which results in a 16 bit word when internally divided. Again, detailed explanations follow.

5.12.5. Frequency Scaling

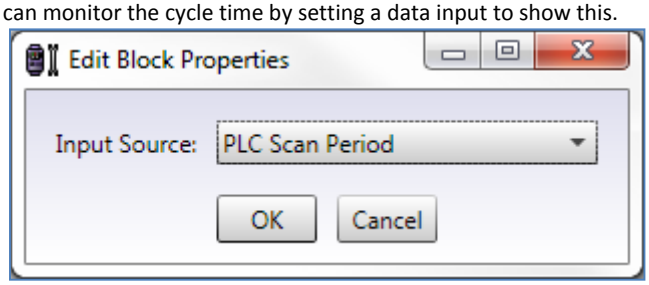
Real values such as Analog input signals are handled as 12 bit numbers (0 – 4096). However, to simplify scaling, Frequency values are treated as 0 – 3000, where 3000 corresponds to the maximum frequency. Therefore it is necessary to rescale any analog calculations if they are used for frequency control. There is a function block to do this.

5.12.6. Data as a Percentage

For some functions, such as an Analog input, the option of working in percentage is provided. Values between 0 and 100% will be handled as 0 – 1000. If percentage and 12 bit numbers are mixed, confusing but logical results will be obtained. For example, adding 50% (500) to 2750 using an add function block results in 3250. Again, function blocks are provided to make conversion.

5.12.7. Programme Execution Speed

The programme execution time in the drive depends on many factors. A complex programme may have a cycle time of up to 100ms, but most programmes will be a lot quicker. You can monitor the cycle time by setting a data input to show this.



5.12.8. Programme Size

The programme size is also dependent of the number and complexity of blocks used. Typically up to 100 blocks may be used with the existing memory capacity. When a valid programme is completed, the amount of memory used is shown next to the valid PLC programme indicator in the bottom left hand part of the screen (see later)
The next section of this document takes a simple example of a Function Block programme and goes through all the steps to enable it to operate in a drive.

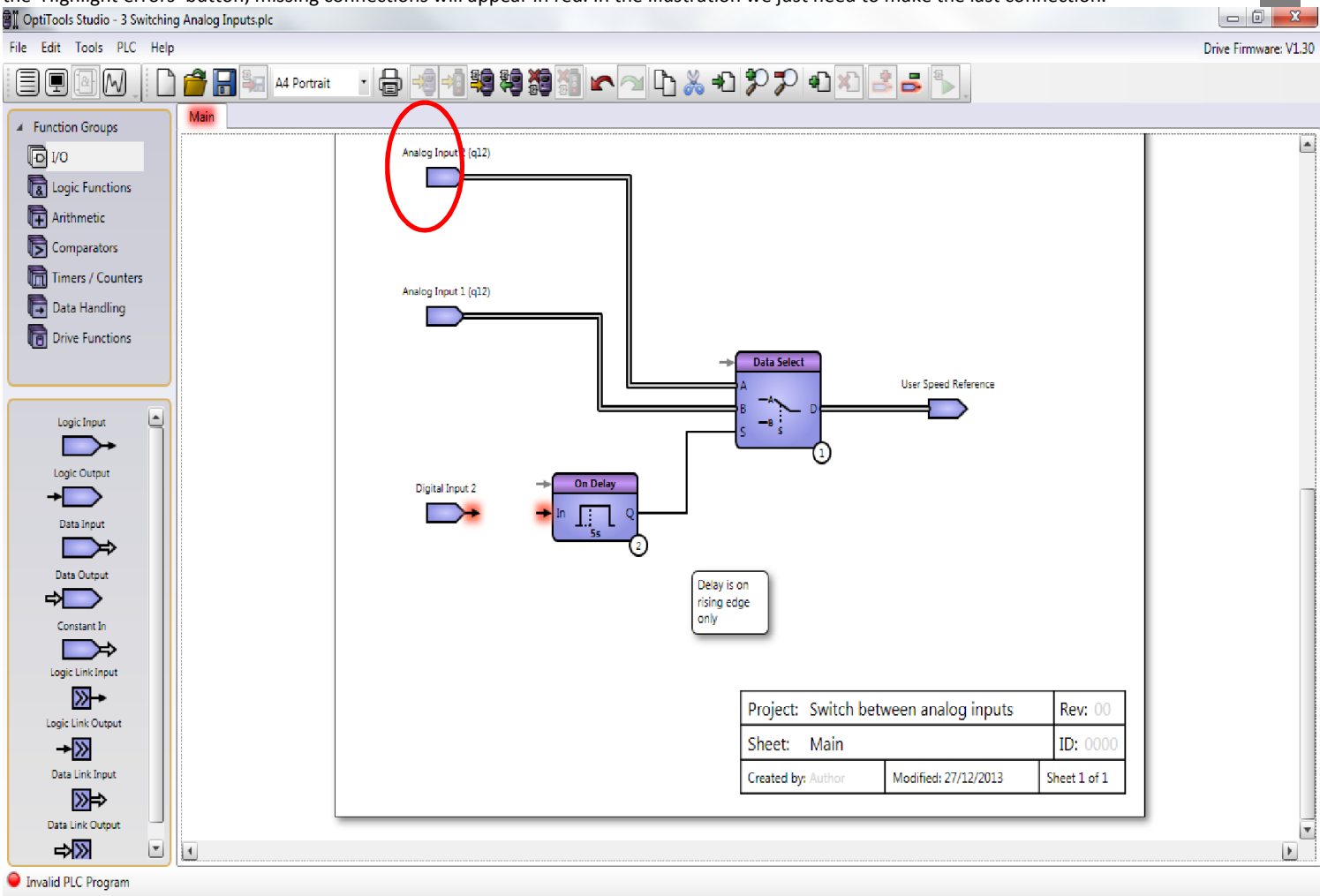
5.13. Programme Example

The following programme uses digital input 2 to switch the frequency control between the two analog inputs. The signal from the digital input is delayed by 5 seconds.
Basic Steps:
1. Draw the programme in OTS Function Block Editor
2. Simulate the programme to check it works
3. Upload the programme to a drive
4. Change the parameters in the drive (Using the keypad, or more conveniently, the parameter editor in OTS) to suit the programme.
5. Test the operation within the drive.

Of course, in a real application, it would be necessary to define carefully the programme requirements, including for example what happens if the drive is stopped or tripped externally, to ensure the programme meets the practical needs of the application.

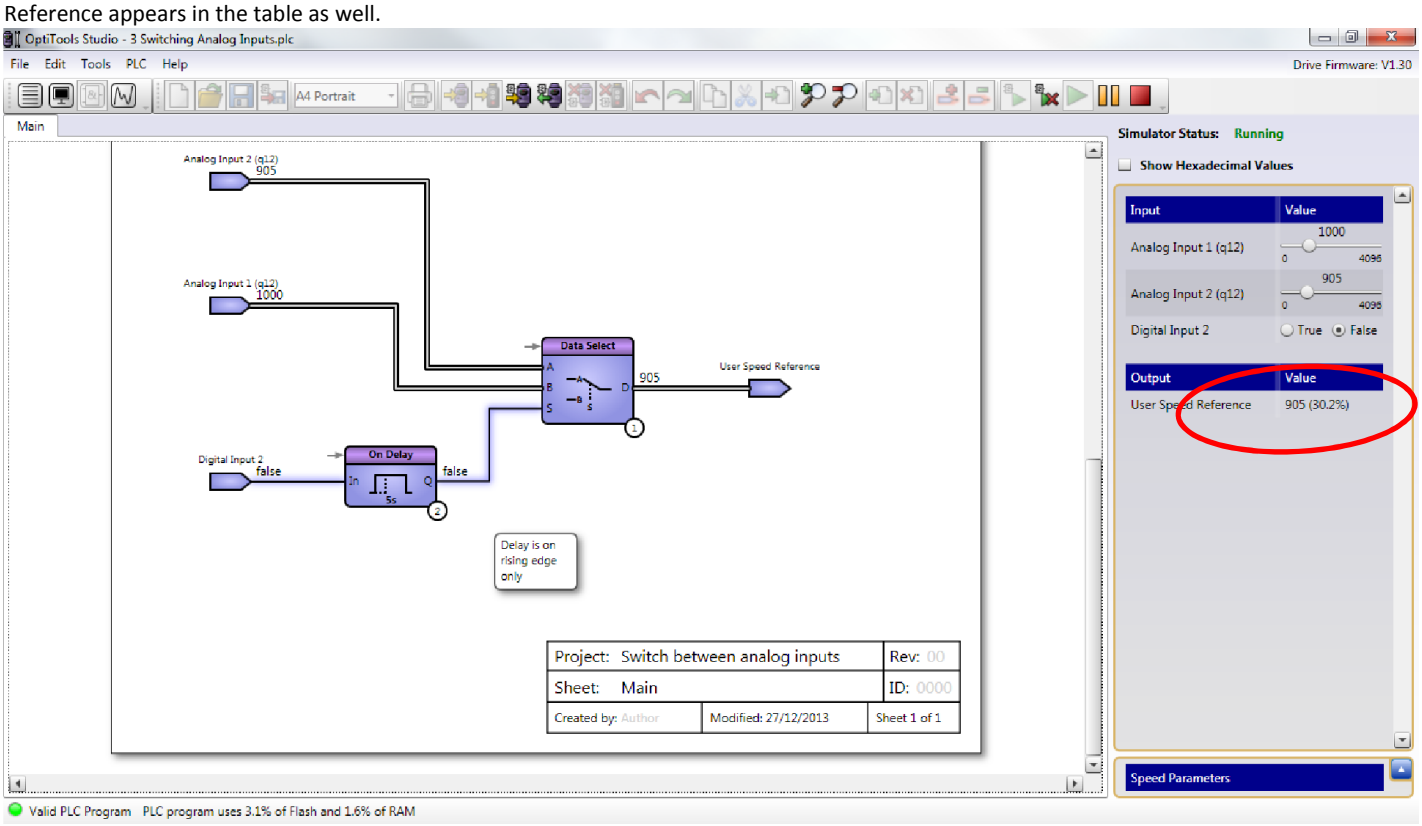
5.13.1. Draw up the programme

Open OTS and select the Function Block Editor. Drag the required function blocks from the groups on the left of the screen. The data select function can be found in the Data Handling group, the On Delay is in the Timer/Counter group, and all the inputs and outputs are in the I/O group.
Connect them up by dragging from one position to the other.
Double click (or right click and select edit properties) on the inputs and outputs to select where they go (see illustration). You can use the data inputs as either percentage or 12 bit numbers. Here I've chosen to use 12 bit numbers. Double click on the On Delay timer to select the delay time of 5 seconds. You can add a comment by right clicking on the worksheet, and add a title at the bottom.
If everything is properly connected, you'll get a green light at the bottom left of the screen indicating a valid programme. If not, try clicking on the 'Highlight errors' button; missing connections will appear in red. In the Illustration we just need to make the last connection.

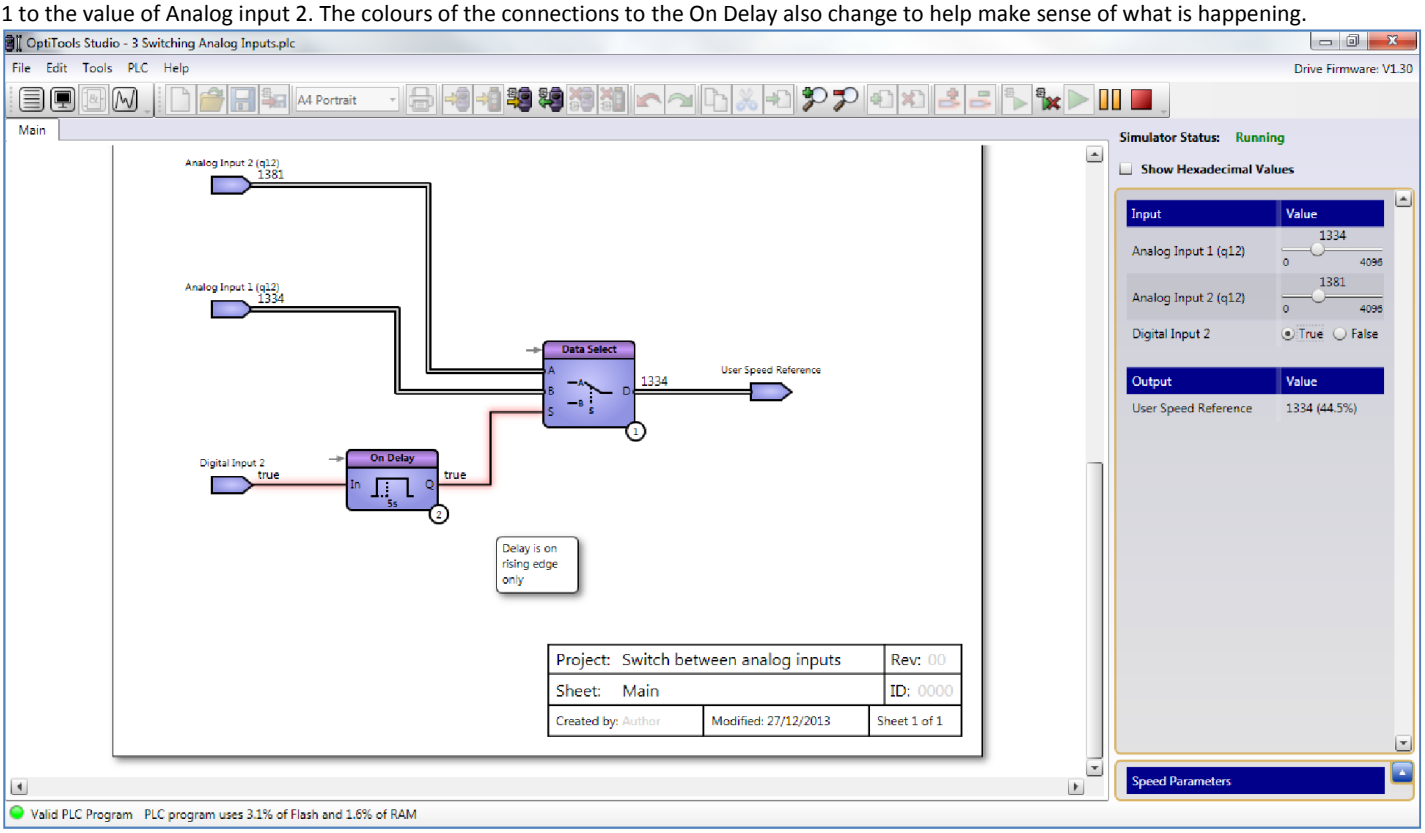


5.13.2. Simulation

Once the programme is valid, enter simulation using the button on the right of the screen. Run the simulation by pressing the green arrow. The simulation status changes to Running. Now use the drag buttons next to the analogue inputs to simulate changes in value. These values appear next to the inputs, as well as in the table, and of course at the output of the Data Select Function Block. The value appearing at the User Speed Reference appears in the table as well.



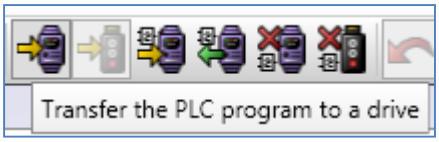
Now change the status of the digital input by pressing the True button. After about 5 seconds (the simulation doesn't track true time exactly) the status of the output of the On Delay block changes and the output of the Data Select block switches from whatever was set in Analog input 1 to the value of Analog input 2. The colours of the connections to the On Delay also change to help make sense of what is happening.



It's probably a good idea to save the programme if you haven't done so already, so stop the simulation, exit the simulation screen and save the file in the usual way. If you haven't got a licence you won't be able to save or upload.

5.13.3.3. Uploading the Programme.

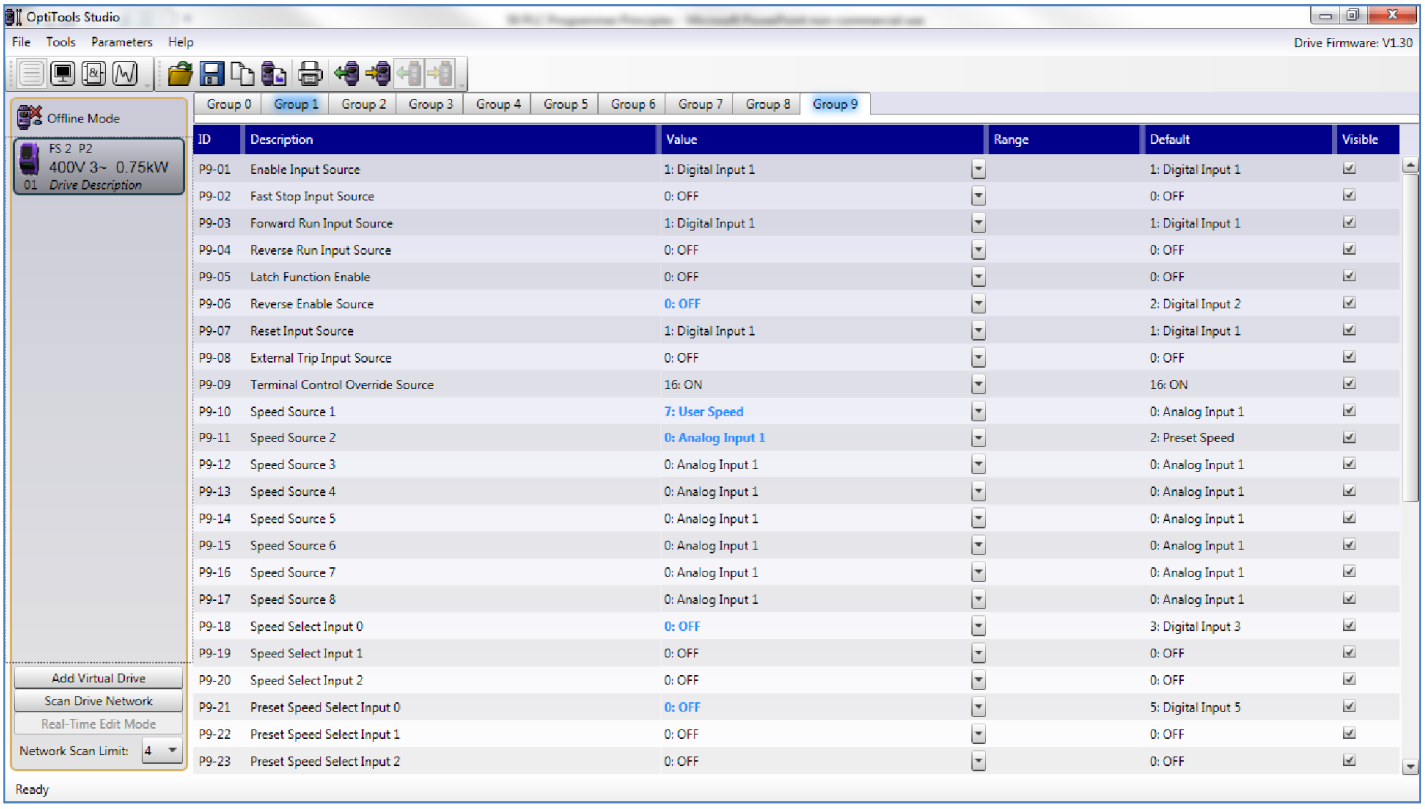
In order to upload the programme, you'll need to set up communication with a drive using OTS and a suitable interface connection. If you haven't used OTS like this before it's a good idea to get used to working with the drive this way by using the parameter editor to change parameters and to upload and download parameter sets.
Now, from within the Function Block Editor, upload the programme in the same way as a parameter set.



Use the buttons at the top of the screen to start the upload. You'll get a prompt to agree the disclaimer and set the security bit. Just check the disclaimer and continue. Remember you'll need a licenced copy to do this.
The programme will take a few seconds to upload, and a prompt will indicate this has happened and been successful. As mentioned before, the drive display will briefly flash **dL – PLC.**

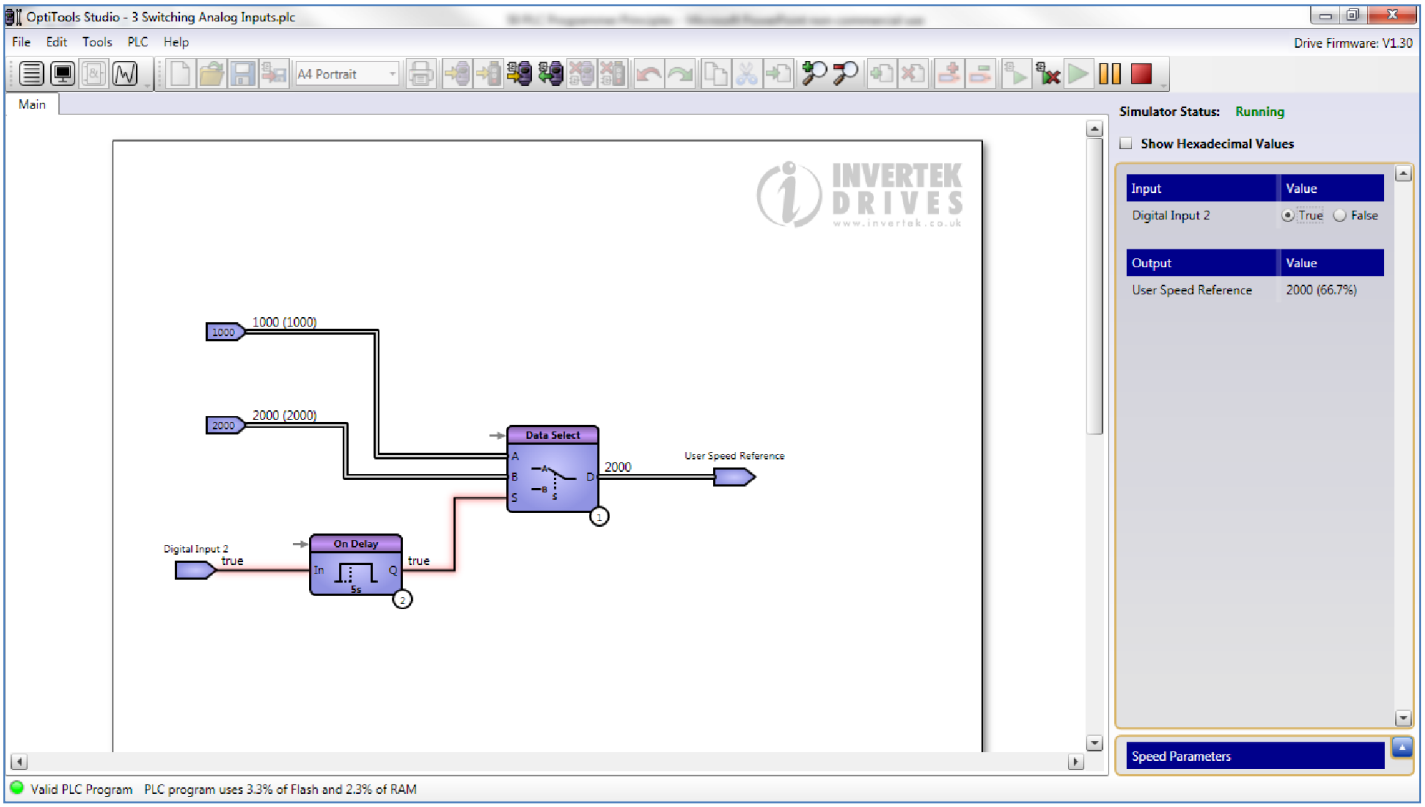
5.13.4. Setting the Necessary Parameters

To make the programme work we need to change a few parameters. Do this from the parameter editor while you are still connected. You'll need to fully access all parameter groups, so set P1-14 to 201 (or whatever you've set the password to if you've changed it)
1. Enable the function block handling in the drive by setting P6-10 = 1 (PLC Function Enable).
2. Free off the connections of the I/O functions by setting P1-13 = 0 (Digital Inputs Function Select).
3. Now look at Group 9 parameters. You'll see the settings (in column 3, changes in blue) have been changed from their default in order to free off functions. For this application we can reconnect most of the functions to their default settings again. So reconnect P9-01, 03, 07, 11. You can check the defaults easily from column 5. Now these functions will work as before. However, digital input 2 was connected to the reverse function, and we'll use it for the switchover function, so we haven't reconnected it to the reverse function P9-06, but left it for the Function Block Programme to use. Finally we should decide where our speed source P9-10 should come from. It was from Analog input 1, but now we'll connect it to User Speed, which is the User Speed Reference in the Function Block programme we uploaded. The picture below shows the necessary settings.



ID	Description	Value	Range	Default	Visible
P9-01	Enable Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-02	Fast Stop Input Source	0: OFF		0: OFF	☑
P9-03	Forward Run Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-04	Reverse Run Input Source	0: OFF		0: OFF	☑
P9-05	Latch Function Enable	0: OFF		0: OFF	☑
P9-06	Reverse Enable Source	0: OFF		2: Digital Input 2	☑
P9-07	Reset Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-08	External Trip Input Source	0: OFF		0: OFF	☑
P9-09	Terminal Control Override Source	16: ON		16: ON	☑
P9-10	Speed Source 1	7: User Speed		0: Analog Input 1	☑
P9-11	Speed Source 2	0: Analog Input 1		2: Preset Speed	☑
P9-12	Speed Source 3	0: Analog Input 1		0: Analog Input 1	☑
P9-13	Speed Source 4	0: Analog Input 1		0: Analog Input 1	☑
P9-14	Speed Source 5	0: Analog Input 1		0: Analog Input 1	☑
P9-15	Speed Source 6	0: Analog Input 1		0: Analog Input 1	☑
P9-16	Speed Source 7	0: Analog Input 1		0: Analog Input 1	☑
P9-17	Speed Source 8	0: Analog Input 1		0: Analog Input 1	☑
P9-18	Speed Select Input 0	0: OFF		3: Digital Input 3	☑
P9-19	Speed Select Input 1	0: OFF		0: OFF	☑
P9-20	Speed Select Input 2	0: OFF		0: OFF	☑
P9-21	Preset Speed Select Input 0	0: OFF		5: Digital Input 5	☑
P9-22	Preset Speed Select Input 1	0: OFF		0: OFF	☑
P9-23	Preset Speed Select Input 2	0: OFF		0: OFF	☑

So now it only remains to run the drive and test the action of digital input 2 to switch between the two analog inputs, not forgetting the delay on the rising edge.
As a variant, you can replace the switching of analog inputs by fixed settings to save connecting potentiometers to the analog inputs. Replace the data inputs with Constants, set to whatever you wish. Here's the simulation:

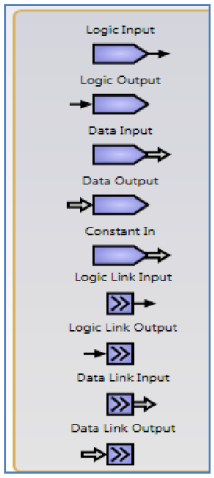


You can see that preparing the programme is pretty simple, and provided you have made the necessary changes to parameters it is quick to upload and check out in the drive.

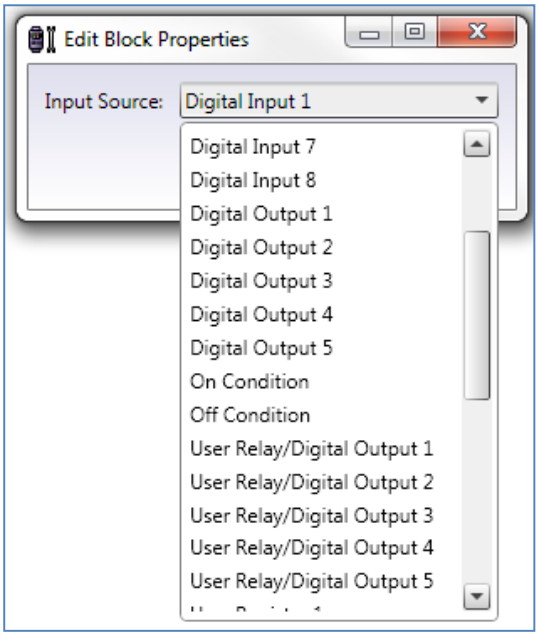
6. Function Block Editor – Function Block Descriptions

The following pages explain how individual function blocks work. Except for the simplest, a functional example showing how each block works is shown. These examples do not represent practical or useful programmes, but are intended to illustrate only the operation of the function block. If the operation is unclear, create the programme yourself and run the simulation. You'll see the values change as expected. Don't forget to press the green arrow to run the simulation. Remember the timing may not be accurate as your PC may be busy with other things. Some examples show more than one block in operation.

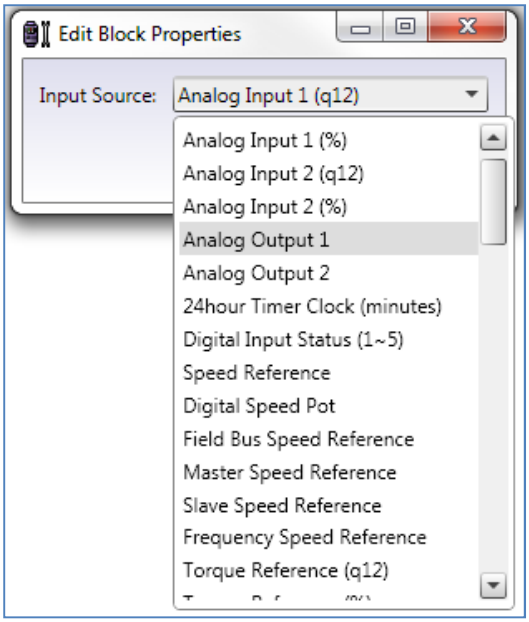
6.1. Input Output Function Groups



These blocks make links to other functions or worksheets within the drive or programme.
The logic inputs can be connected to digital inputs, user registers etc. The outputs can have similar connections, including relay controls. These options are visible in the drop down menus when editing the block properties.



Similarly, the data inputs can have many different connections:



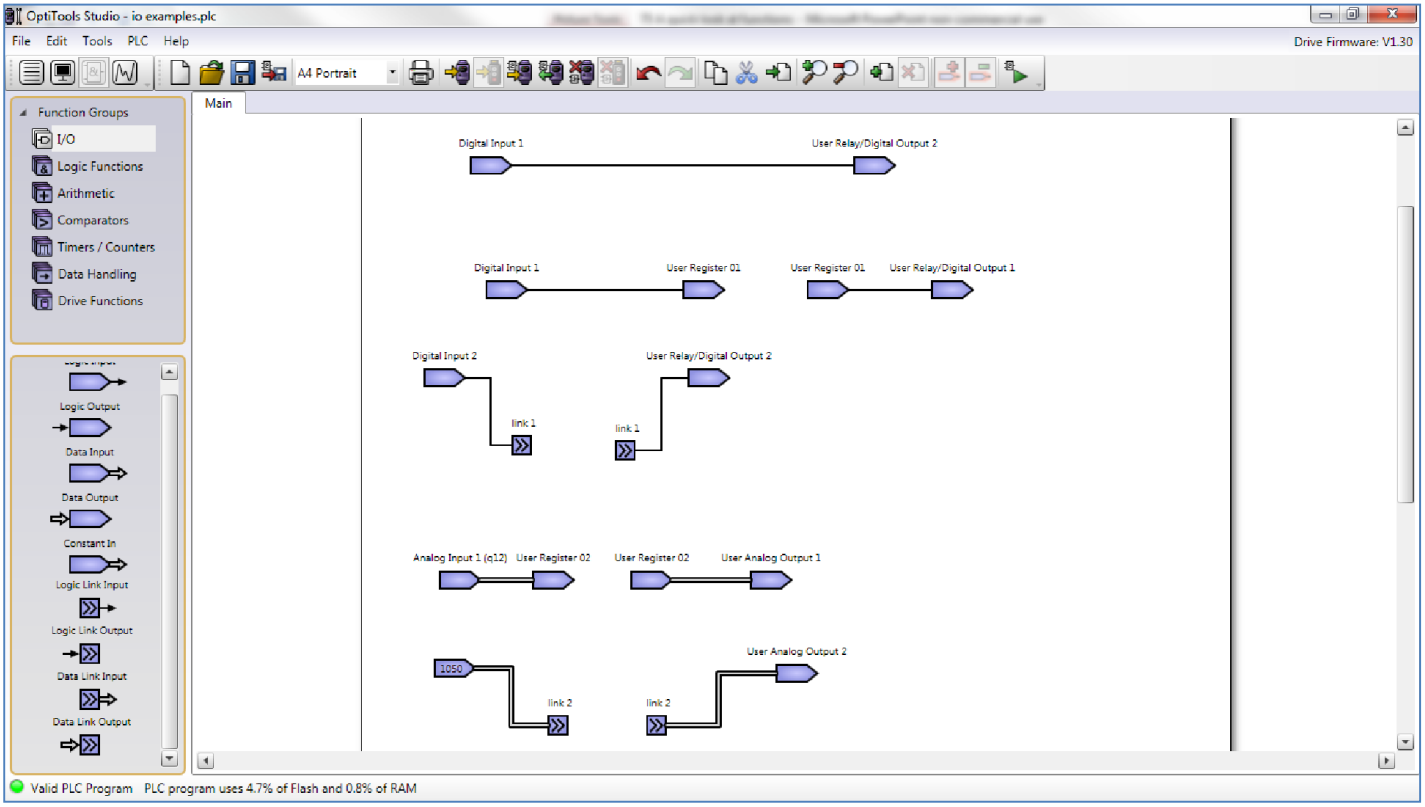
Some of the possible connections are logically possible, but may not appear useful at this stage. These may prove useful as the user gains experience, or may simply be included for completeness.

Link boxes allow connections to be made across multiple sheets, or where connections would confuse the drawing. They can be labelled as required (Pump 1 trip, Low alarm, Link A etc.) The links connect to links of the same name. When preparing the link, existing names are offered as an option to aid correct connection.

In the trivial examples below, the first connection is simply between digital input 1 and relay output 2 with no intervening logic. The second example shows a connection from the same digital input to relay output 1, but via a register. Of course, in a real programme the registers would be connected to other function blocks.

The third example shows the use of a link (labelled link1). Again, the link inputs and outputs would normally be on separate sheets.

Similar examples are shown for data connections. The final example uses a constant input to an analogue output via link 2.



If you prepare this programme yourself and run the simulator you can observe the effect of the connections.

Note that some inputs and outputs will be limited to 12bit or percentage values. Take care when connecting these to larger, 16 bit numbers or to signed numbers, where overflow or underflow will result in confusing results. Always check by simulation.

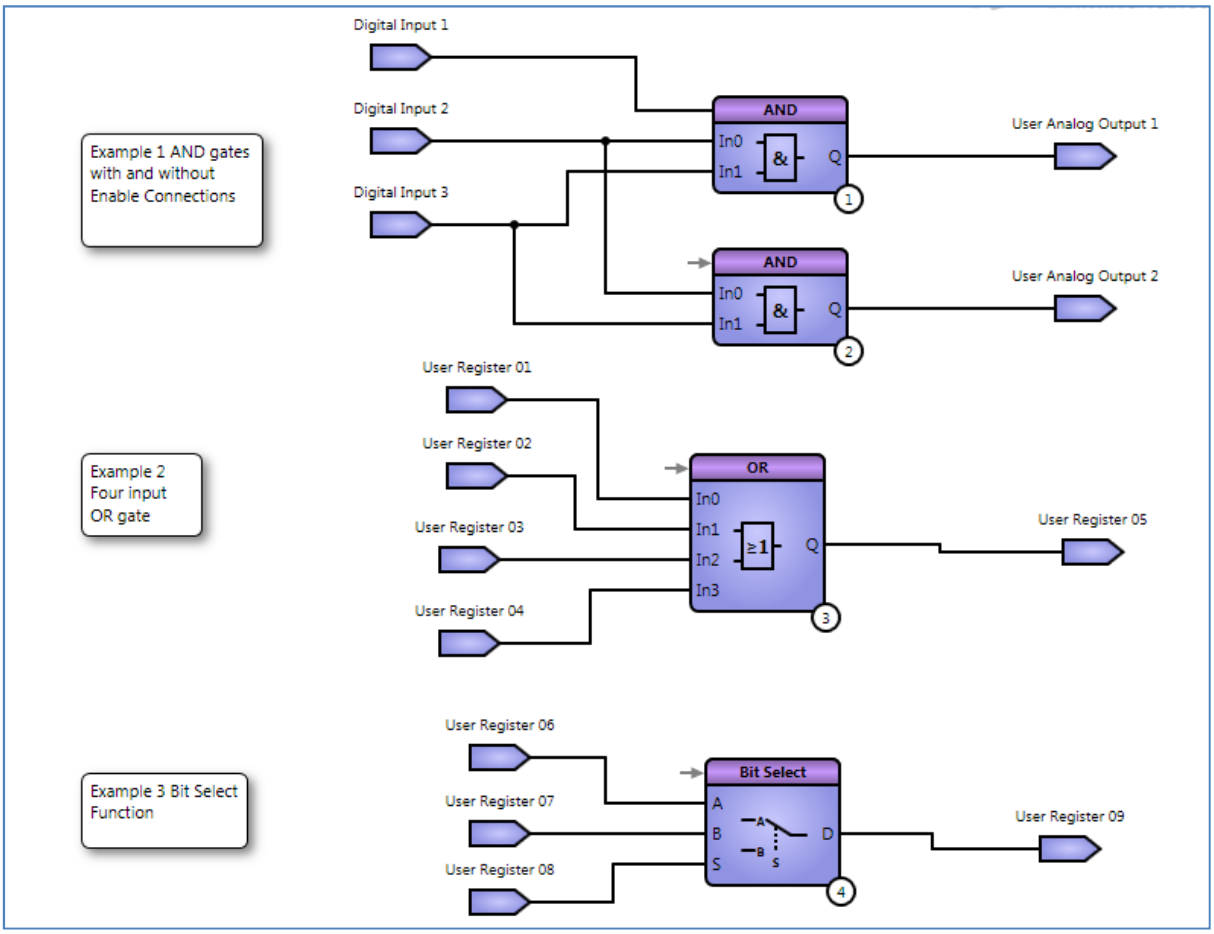
6.2. Logic Functions

These functions operate as expected, and carry out logical Boolean tasks. Where it makes sense, the number of inputs on a logic block can be increased up to 16 be editing the block properties.

In the first example two digital inputs are connected to two AND gates. A third digital input enables the first AND gate. Simulating this will show that, if used, the enable connection at the top of the gate, must be TRUE for the gate to function (First AND gate). If it isn't connected, the gate is automatically enabled (second AND gate).

In the second example, the OR gate is configured with four inputs from User registers, and a fifth register is connected to the output. If this is simulated, the input registers can be set, and the output register 05 monitored.

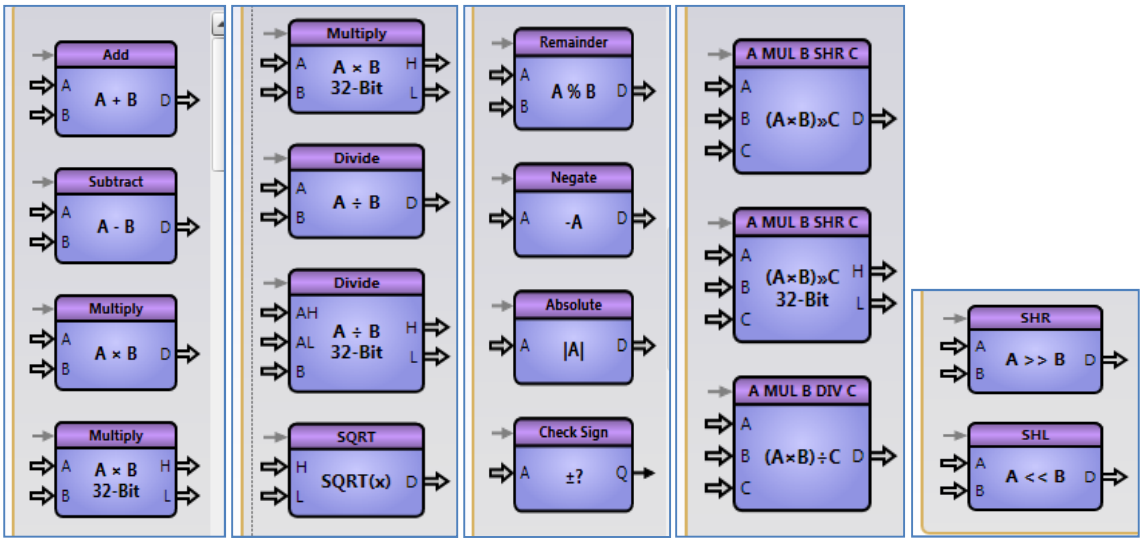
In the third example, the bit select function is demonstrated. This function simplifies selecting logic signals to one function block. Simulating will show that the when user register 8 is FALSE, the value of user register 6 is transmitted to user register 9. When user register 8 is TRUE, user register 7 is transmitted.



In summary, the logic functions work as expected.

6.3. Arithmetic Functions

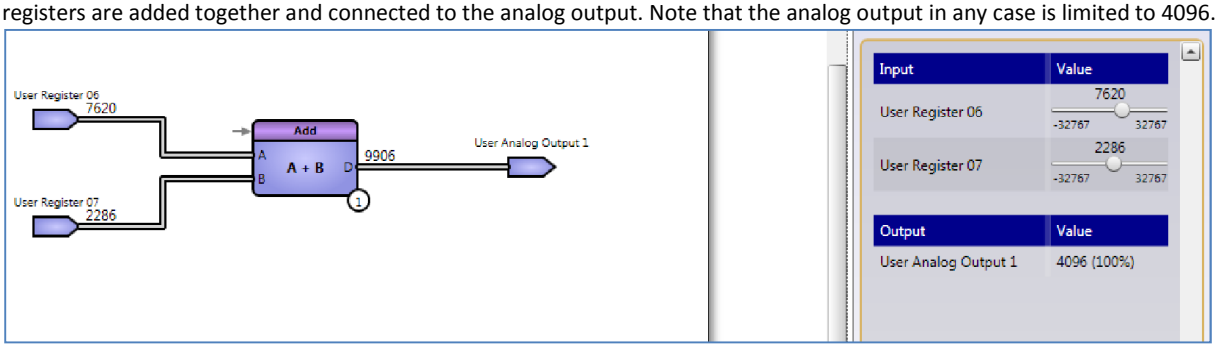
This group of functions provide a full range of arithmetic operations including some 32 bit functions.



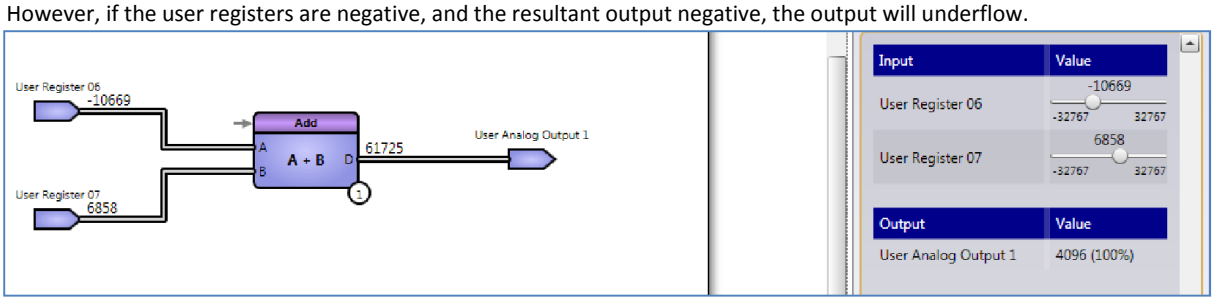
The function blocks work with data inputs, but can be enabled or disabled using the logic enable connection at the top left of the block.

6.3.1. Add and Subtract

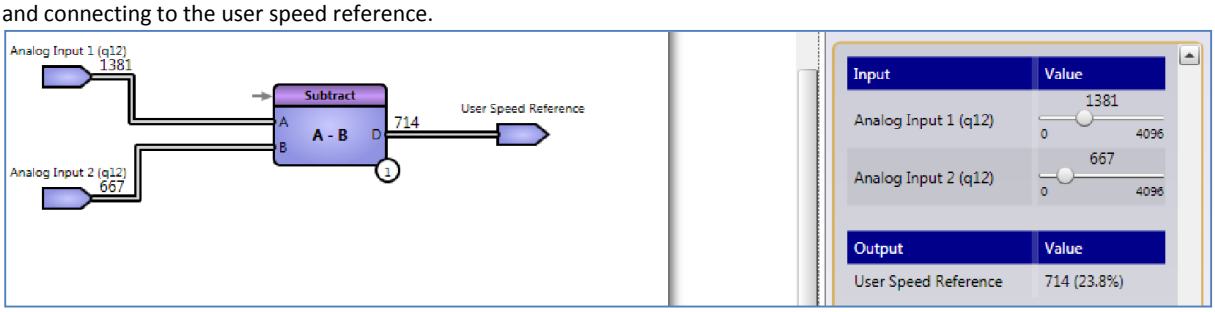
These functions add 16 bit numbers with a 16 bit result. Be very careful when working with signed negative numbers, or subtracting to produce a negative number as these will underflow or overflow and produce unexpected results. So in the example shown below, the two user registers are added together and connected to the analog output. Note that the analog output in any case is limited to 4096.



However, if the user registers are negative, and the resultant output negative, the output will underflow.



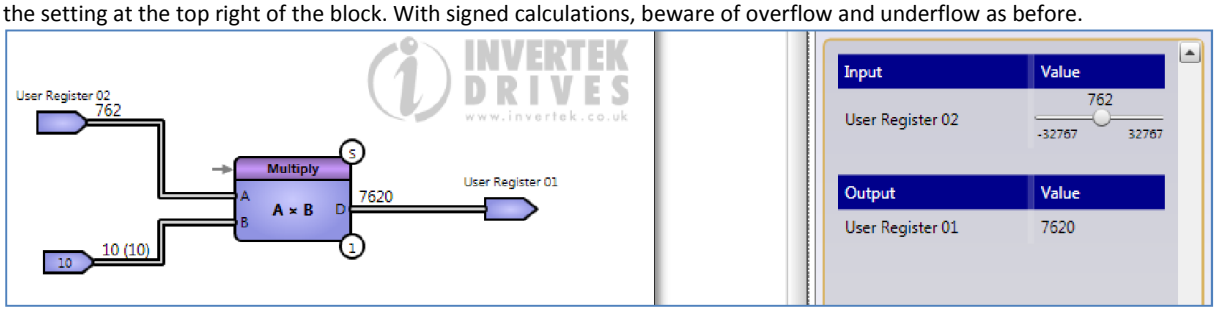
Otherwise, the addition and subtraction functions are straightforward. Here's the subtraction function block, working with two analog inputs and connecting to the user speed reference.



Again, a negative result would result in an underflow.

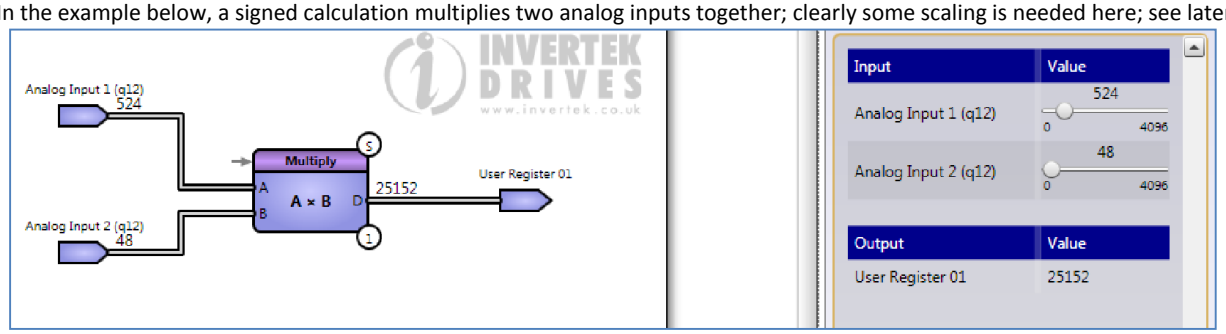
6.3.2. Multiplication and Division

The 16 bit multiplier is the simplest function block here. Note the block can be used as a signed (default) or unsigned multiplier, depending on the setting at the top right of the block. With signed calculations, beware of overflow and underflow as before.

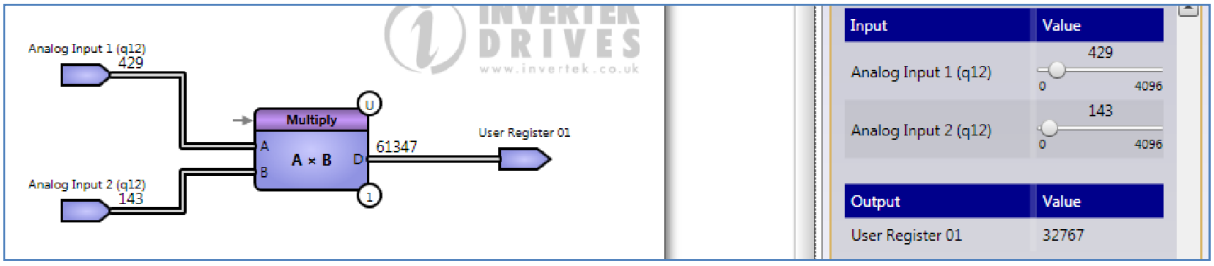


In this example above, the user register 02 is multiplied by a constant in a signed calculation.

In the example below, a signed calculation multiplies two analog inputs together; clearly some scaling is needed here; see later.

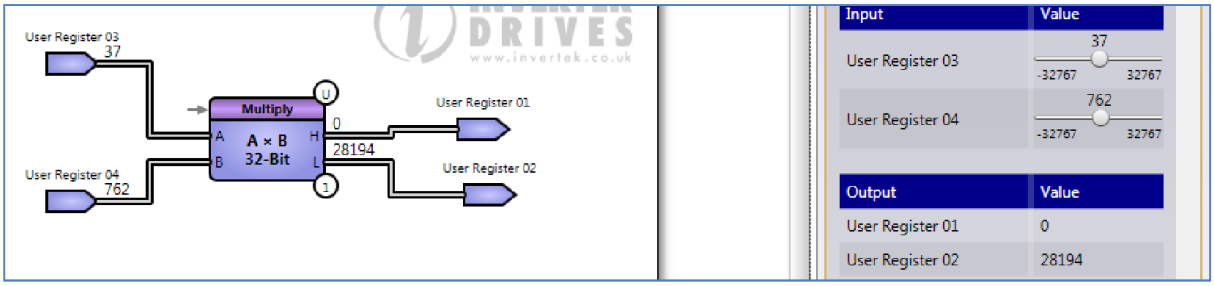


With an unsigned calculation, the output range of the multiplier is increased, but the user register limits its value.

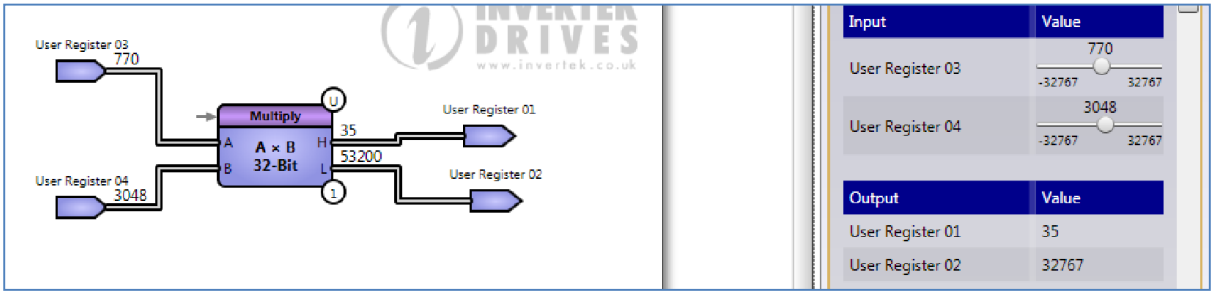


6.3.3. 32 bit Multiplication.

With this function block, two 16 bit numbers are multiplied together to produce a full 32 but result. In the example below, the result is only 16 bit, so the High output is 0.

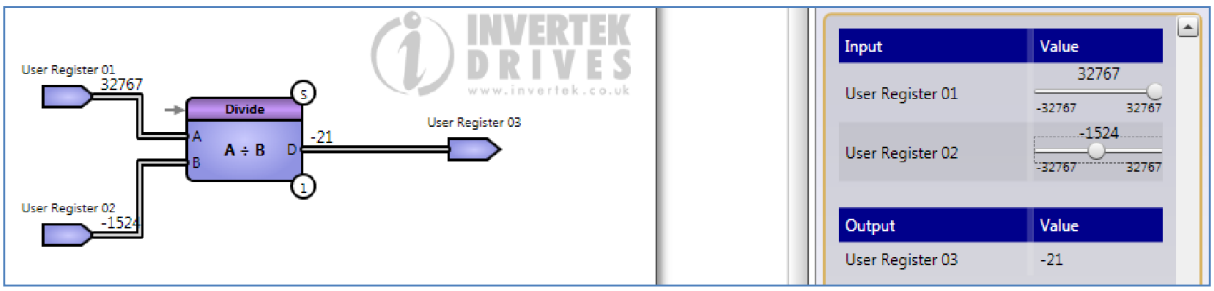


With increased inputs on register 03 and 04, the High output increases. Notice that User Register 02 has limited to its 15 bit value.

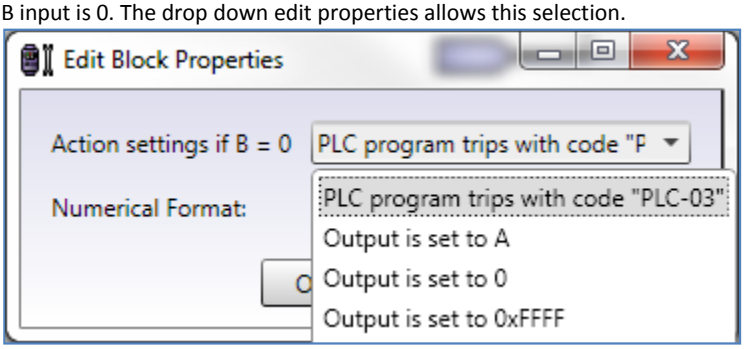


6.3.4. Division

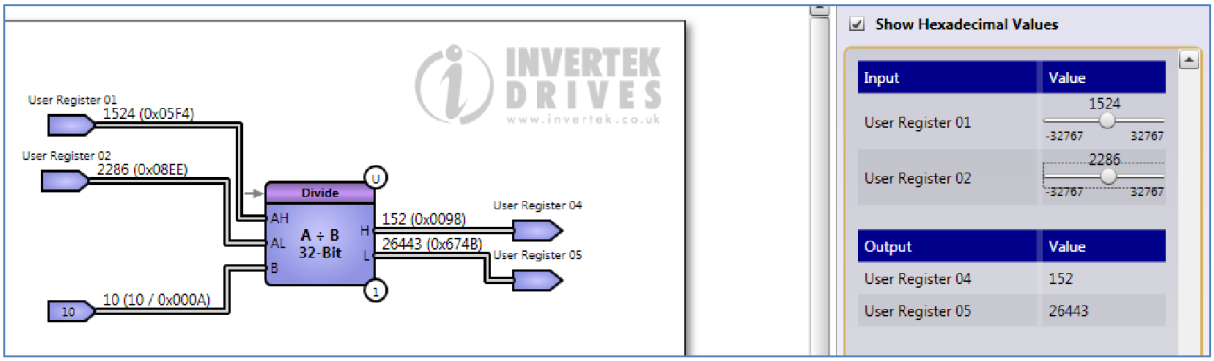
Division function blocks are available in 16 and 32 bit resolution.



In this 16 bit calculation the signed numbers result in a negative result. All the Divide function blocks offer the option of selecting actions if the B input is 0. The drop down edit properties allows this selection.



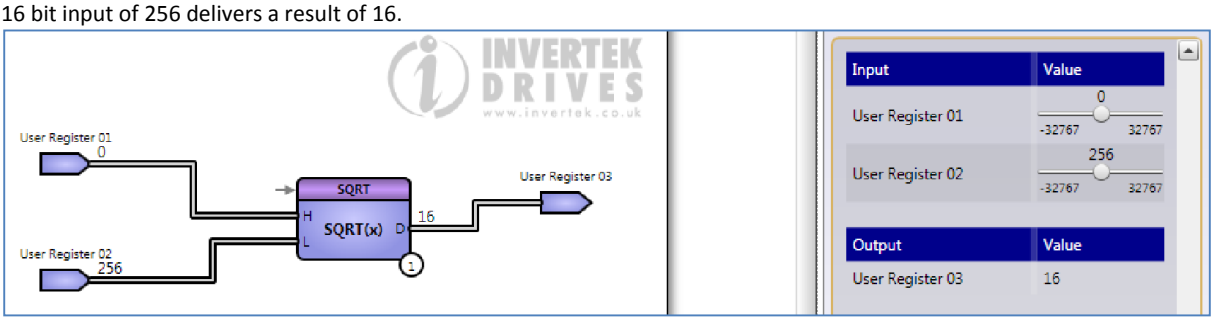
A full 32 bit division function block is also available.



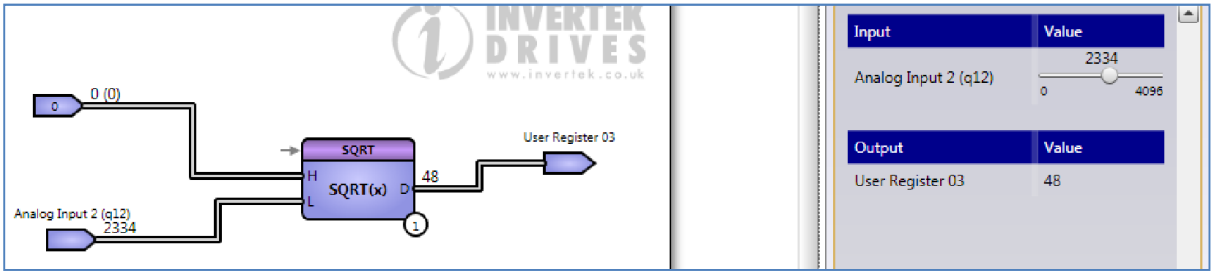
In this example, we've checked the box in the simulation to show hex values to add to the confusion. The 32 bit number is divided by 10, and the high and low outputs connected to User registers.

6.3.5. Square Root

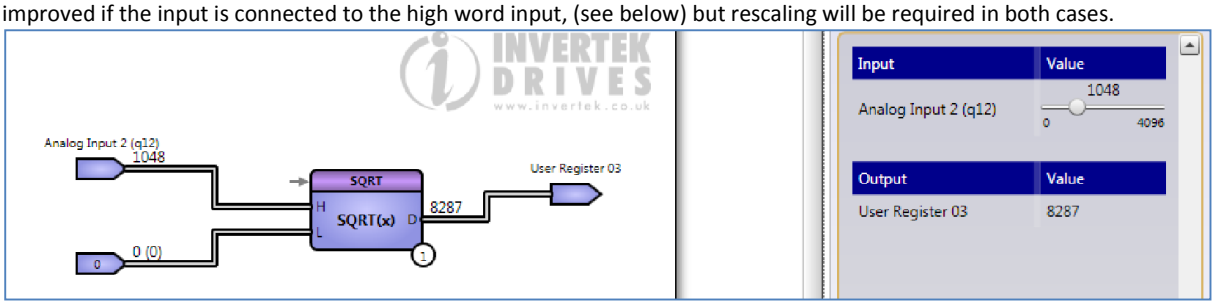
This function takes the root of 32 bit number and produces a 16 bit result. In the trivial example below, the High 16 bit number is 0, so the low 16 bit input of 256 delivers a result of 16.



A more practical example shows the function used to produce the square root of an analog input; maybe to linearize a transducer for example.



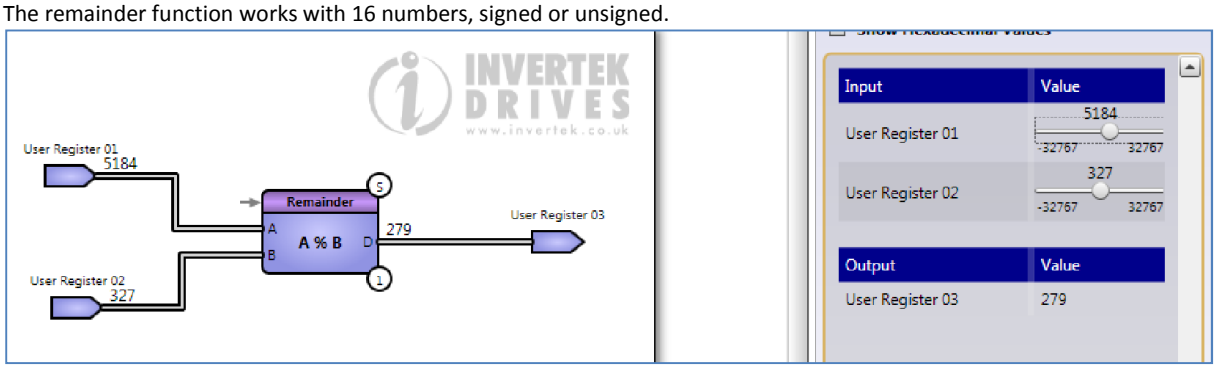
We've used a constant input of 0 and the 12 bit input from Analog input 2 produces a result connected to User Register 3. Resolution is improved if the input is connected to the high word input, (see below) but rescaling will be required in both cases.



Negative inputs will produce results caused by underflow and should be avoided.

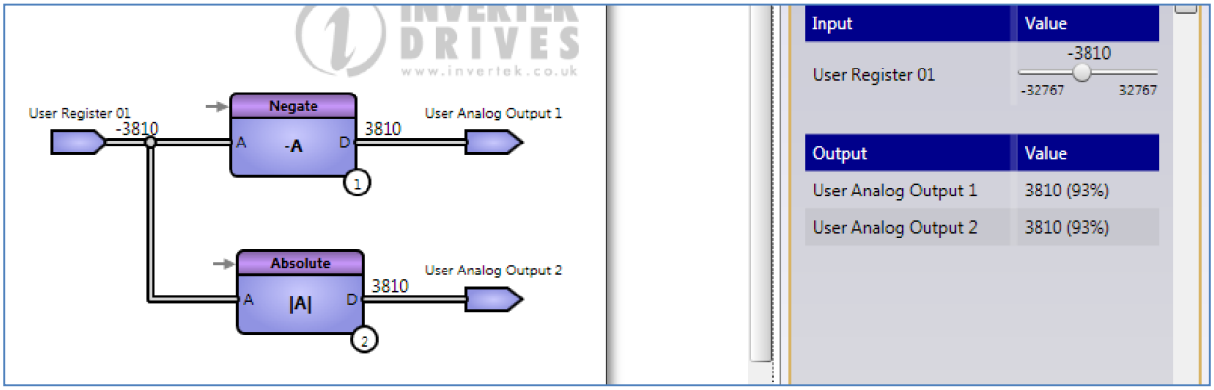
6.3.6. Remainder Function

The remainder function works with 16 numbers, signed or unsigned.

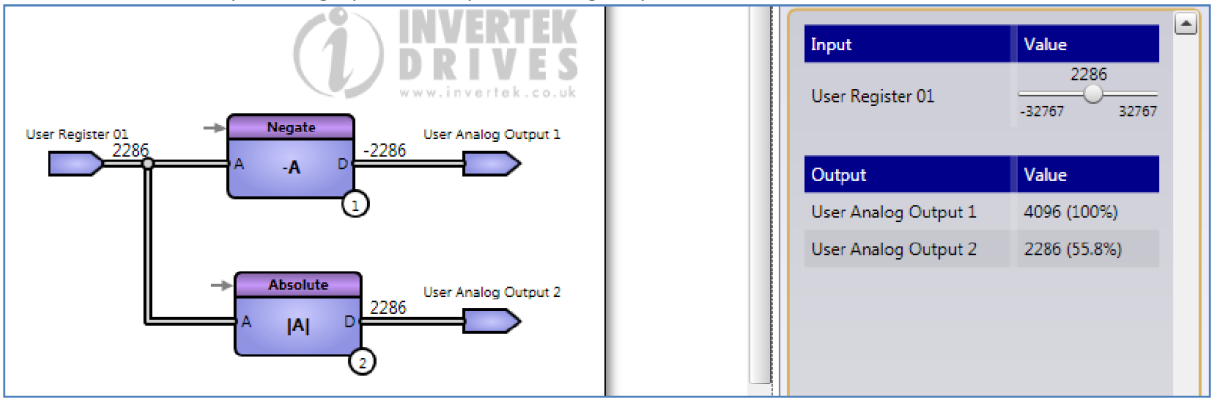


6.3.7. Negate and Absolute Functions

These functions both perform as expected. In the example below, a negative value becomes positive in both cases, delivering a positive value to the analog outputs.

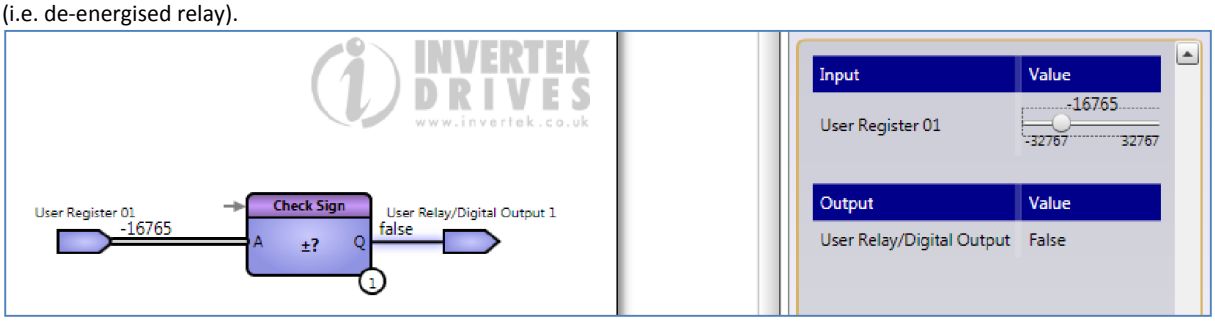


However, a positive input produces a negative result from the Negate function, which underflows at Analog output 1. Using the Absolute function is better here, producing a positive output at Analog Output 2.

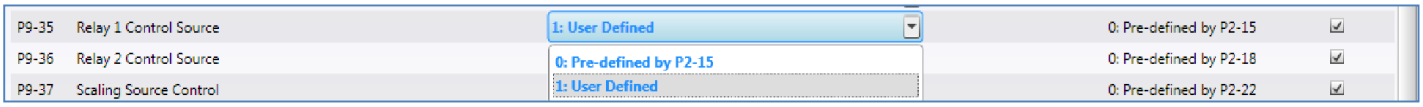


6.3.8. Check Sign

Clearly a useful function to prevent these problems is the check sign function. In the example below, the result is connected to an output relay, which may be used to switch on a pump for example if the result of an internal calculation is negative. The negative value results in a FALSE (i.e. de-energised relay).

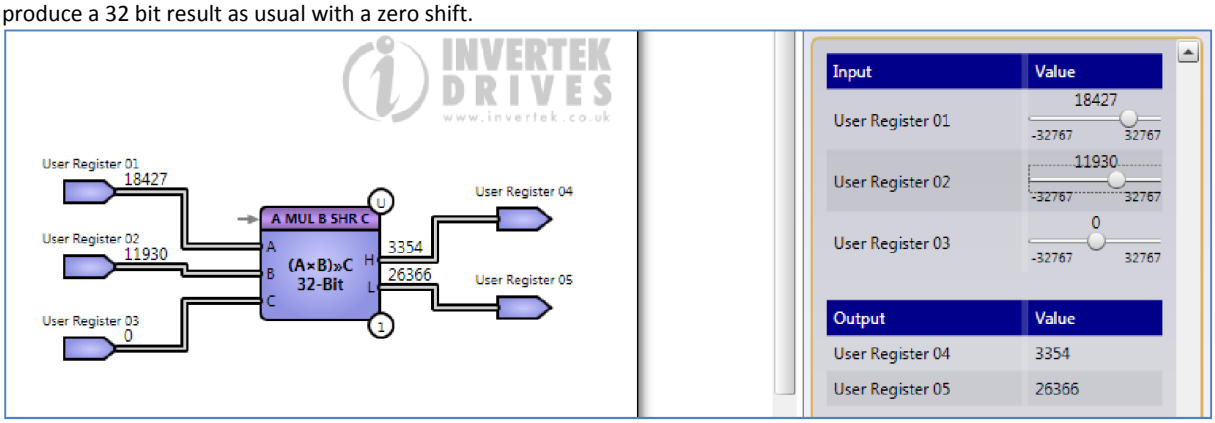


Remember that as well as other changes, it will be necessary to set Parameter P9-35 (Relay 1 Control Source) to 'user defined' using the drop down box.

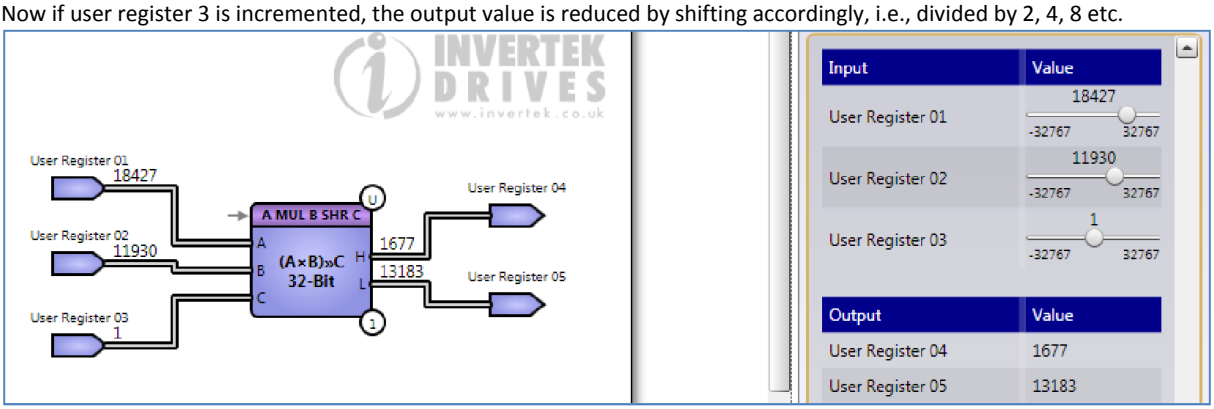


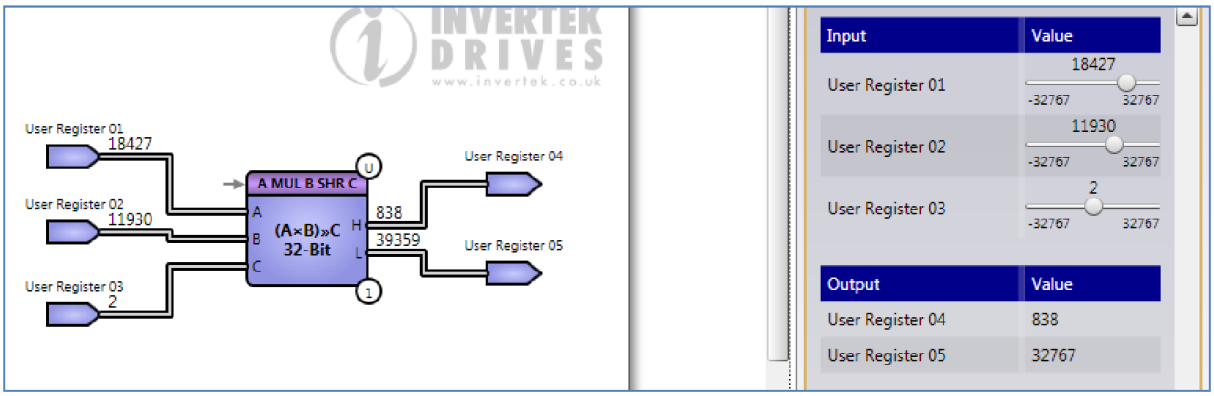
6.3.9. Multiplication with Right Shift.

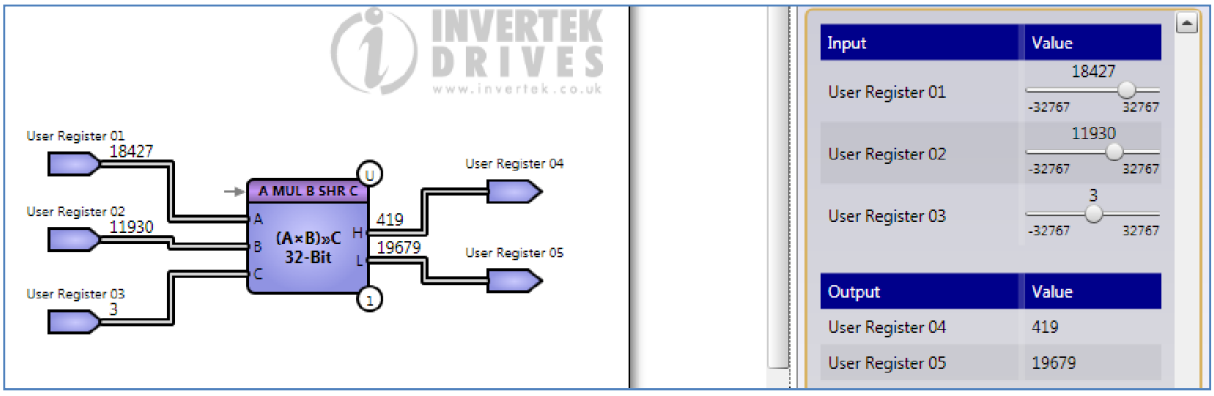
This function multiplies two 16 bit numbers, but includes the ability to right shift the result. In the example below, the unsigned 16 bit inputs produce a 32 bit result as usual with a zero shift.



Now if user register 3 is incremented, the output value is reduced by shifting accordingly, i.e., divided by 2, 4, 8 etc.

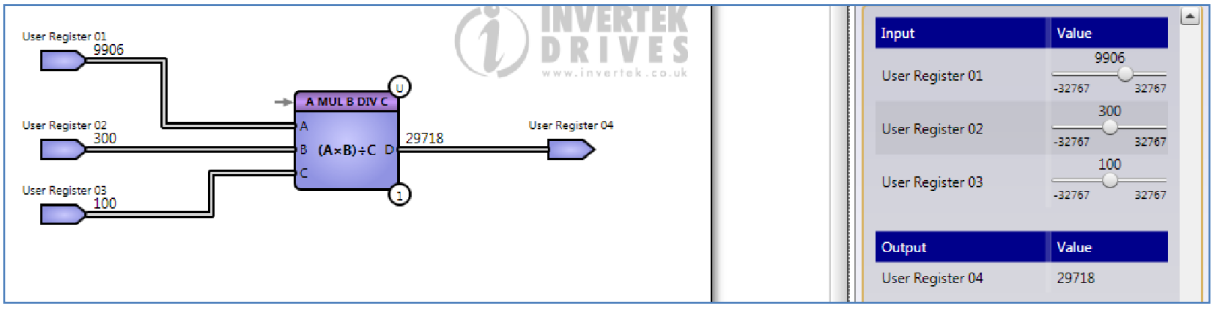




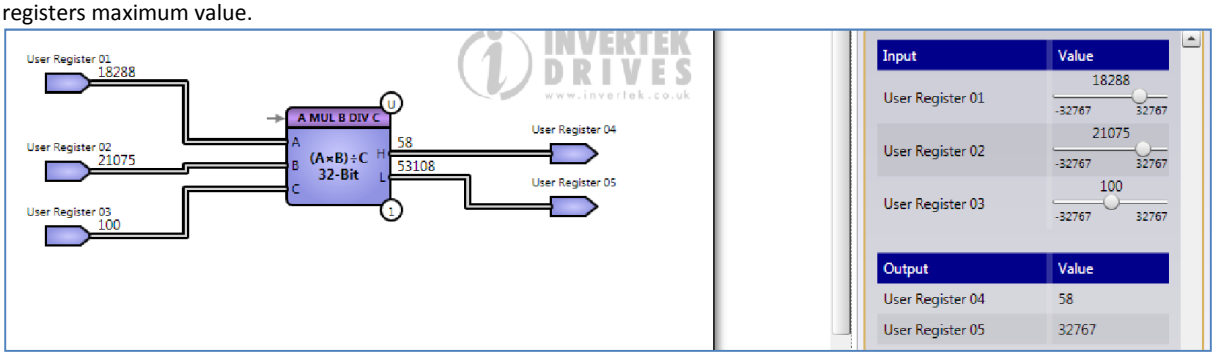


6.3.10. Multiplication and Division blocks.

These blocks carry out multiplication followed by division in one block. Because the internal result of the multiplication is 32 bit, this result retains accuracy compared with using separate 16 bit multiplication and division function blocks. Blocks with a 16 bit and 32 bit results are available. In the example below, using the 16 bit block, the initial multiplication by 300 has been handling internally as a 32 bit number, so there is no loss of accuracy in the result.



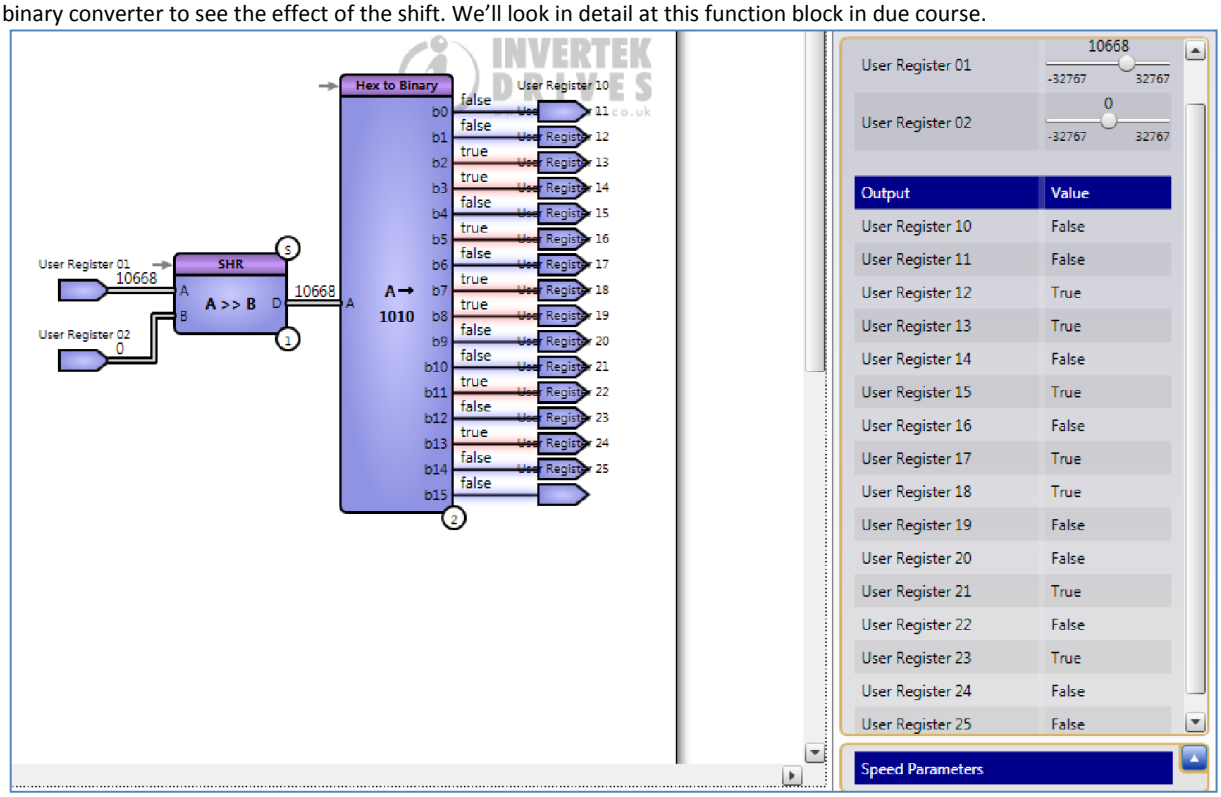
With the 32 bit output block higher numbers can, of course be handled. Notice once again that the lower result output has been limited to the registers maximum value.



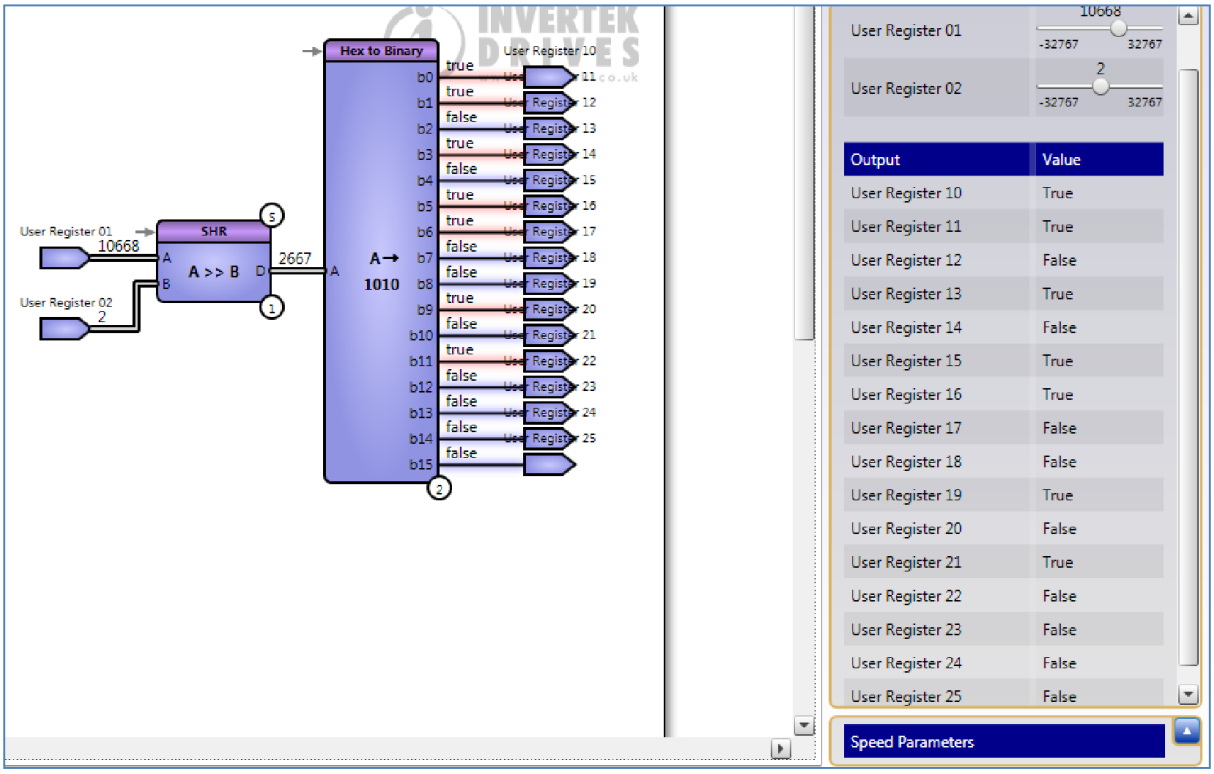
Be very careful working with signed numbers, as results can become confusing.

6.3.11. Shift right and Shift left functions

The final Arithmetic blocks carry out shift functions on 16 bit numbers. We can see the result in the functions below. We've added a Hex to binary converter to see the effect of the shift. We'll look in detail at this function block in due course.

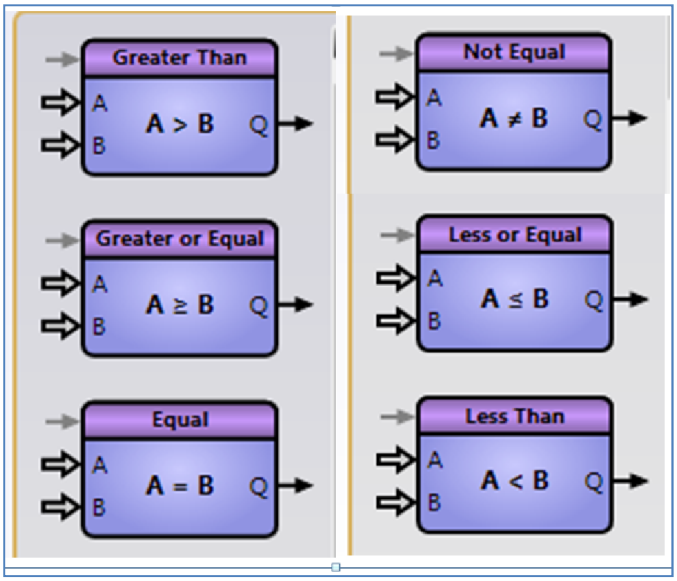


With the value of register 02 changed to 2, we can see the binary values have all moved right (up the table) two places. In effect, the value of the input 10668 is divided by 4 to 2667.

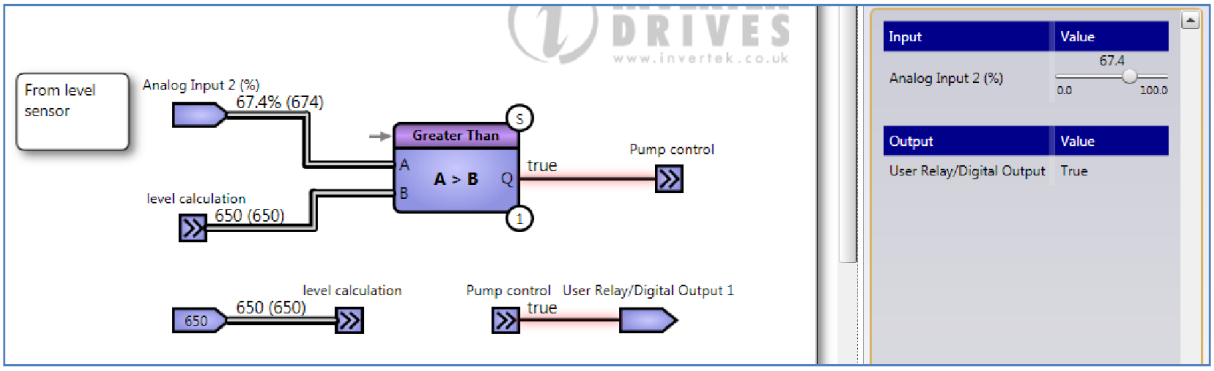


The final block will operate in the same way of course, but shift to the left.

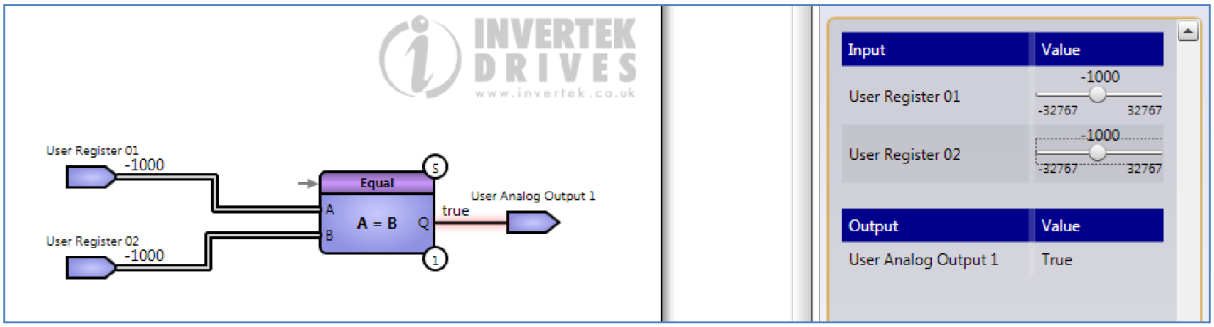
6.4. Comparators



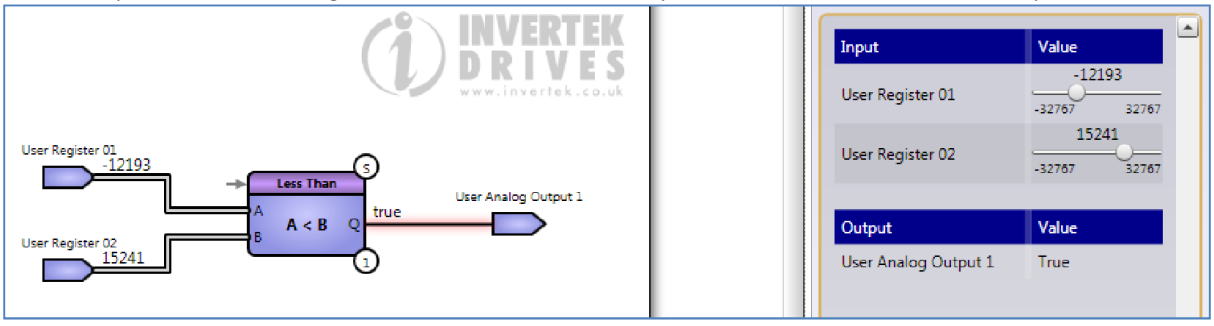
The various comparators produce a logic result (TRUE or FALSE) dependent on the result of the comparison of the two 16 bit inputs.
Comparators are useful for not only monitoring values within programmes, but also for measuring inputs from sensors etc.
In the example below, the comparator may be part of a larger programme. The level detector input is compared with a level calculation, linked in this example simply to a constant, but in a larger programme to another page of logic and calculation. When the level is exceeded, the output may also go for further processing; in this case, it is simply linked to the relay output.



Notice the Analog input has been chosen to operate in percentage, giving an output of 0 to 1000. So the comparator changes state at 650, the value of B via the link from the constant.
The other comparators work as expected; their inputs may be signed or unsigned

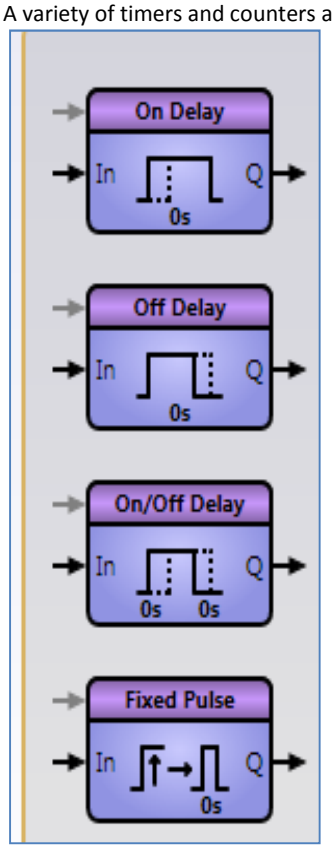


In the example above, the two negative numbers are detected as equal. Below, A is less than B, so the output is TRUE.

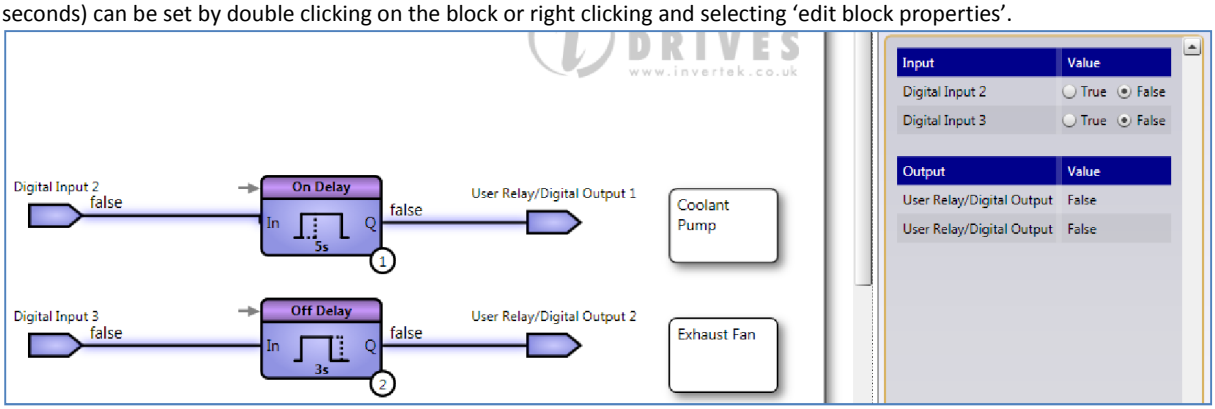


6.5. Timers and Counters

A variety of timers and counters are included here. The simpler function blocks produce a delay in a rising or falling edge.

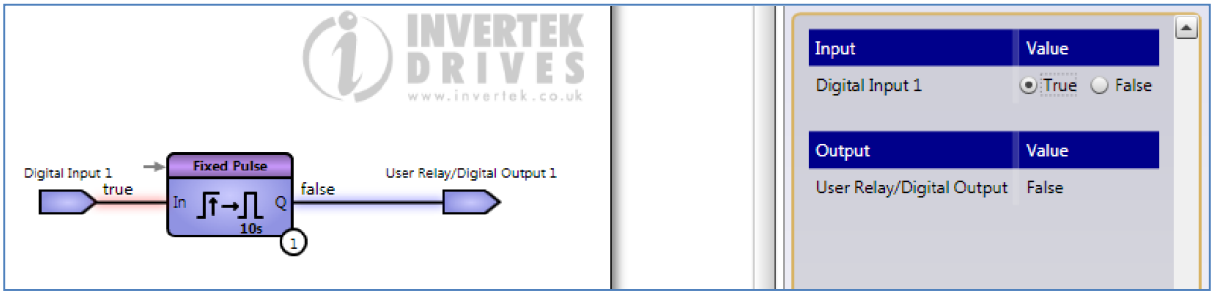


In the example below, the digital inputs, relays and programming capability of the drive have been 'borrowed' to carry out an external function controlling a pump and a fan, independent of the drive operation. The coolant pump is switched on by digital input 2 after a 5 second delay, and the fan is switched off after a 3 second delay from the falling edge of digital input 3. As usual, the properties (in this case the time delay in seconds) can be set by double clicking on the block or right clicking and selecting 'edit block properties'.

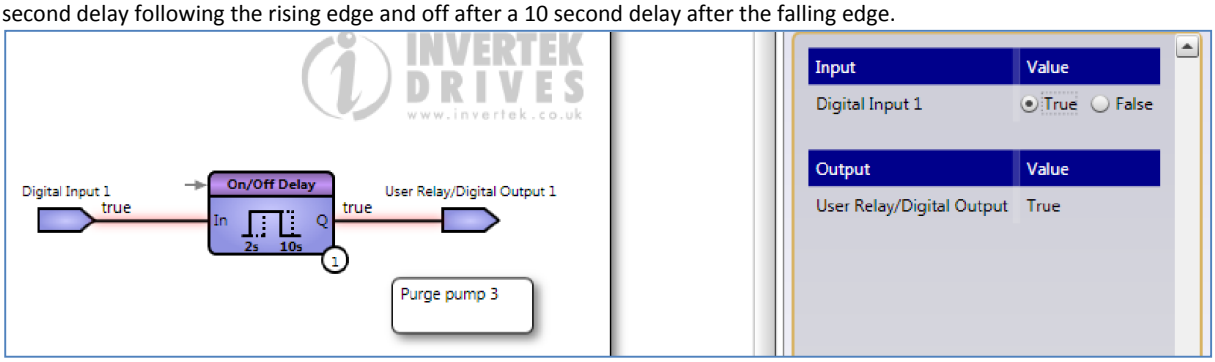


In fact, the programming function of the drive can replace many external components, and can be used independently of the drive itself, providing these features aren't needed to control the drive.

In the example below, the output relay is enabled for 10 seconds whenever digital input 1 has generated a rising edge. The timer has timed out in this simulation.



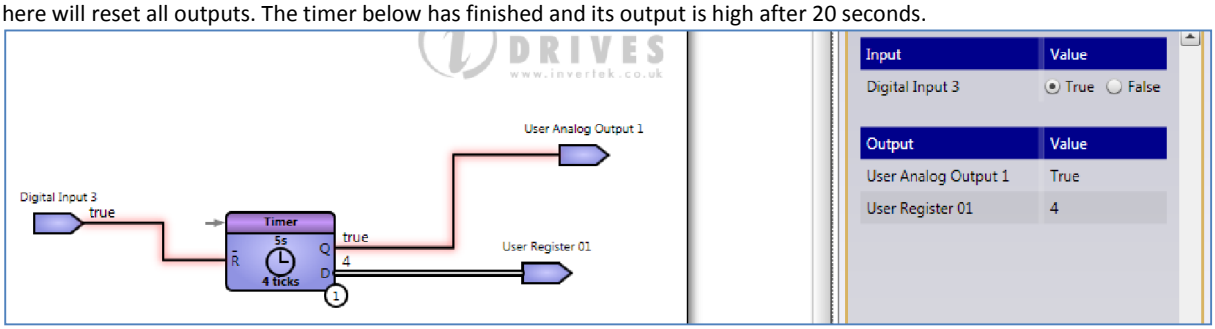
Maybe the On/Off delay function could be used to produce a similar (but not the same) function. Now the purge pump switches on after a two second delay following the rising edge and off after a 10 second delay after the falling edge.



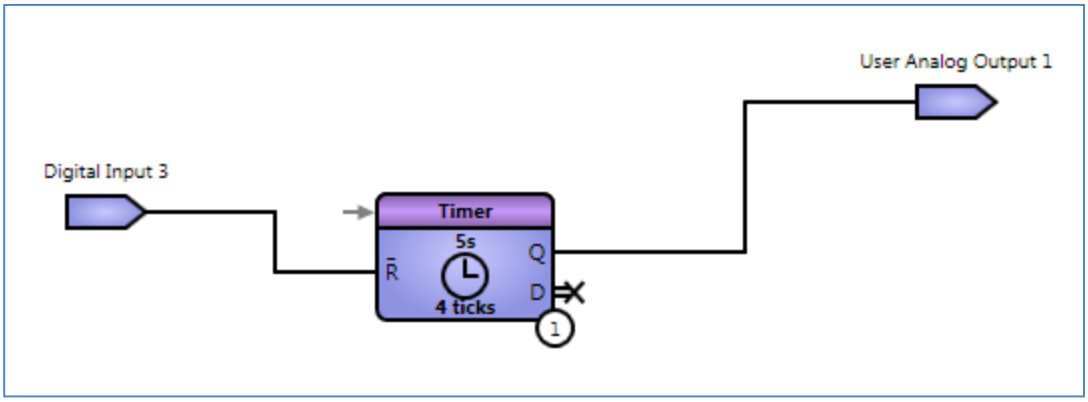
Incidentally, if the TRUE input signal become FALSE before the output has changed state, the output will remain FALSE; this applies to the on delay function block as well, and to falling edges of OFF delays. So it may be necessary to use the fixed pulse function if short pulses are expected and need to be recognised.

6.6. Timers

The timers provided by the function block programme are a little more complex. The timer here has a 'tick' number as well as a time setting. Here we've set a tick of 4 and a time of 5 seconds. That means after 5 seconds (i.e. one 'tick') the data output is incremented, and after 20 seconds (4 times 5 seconds) the Q output changes state from FALSE to TRUE. The input needs to be high of course for the timer to run; a low here will reset all outputs. The timer below has finished and its output is high after 20 seconds.

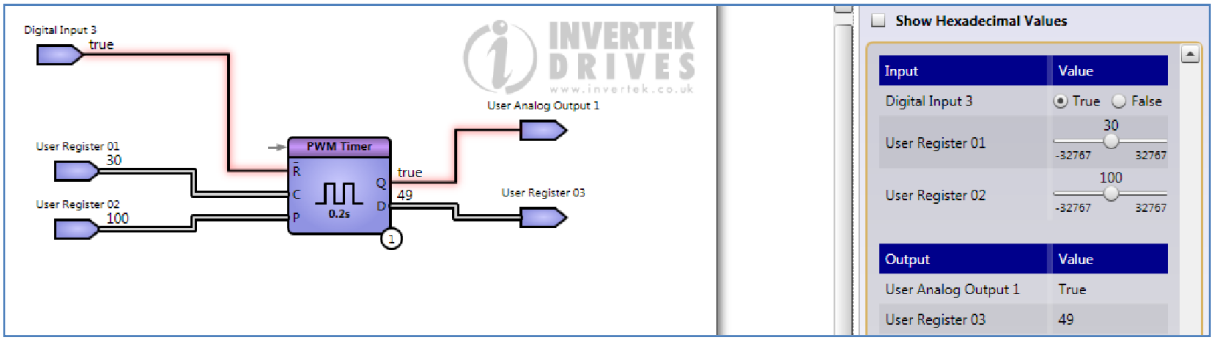


Don't forget if you don't want to connect the data output, you can right click on it and then you won't have to connect anything.



6.6.1. Pulse width Modulation Timer

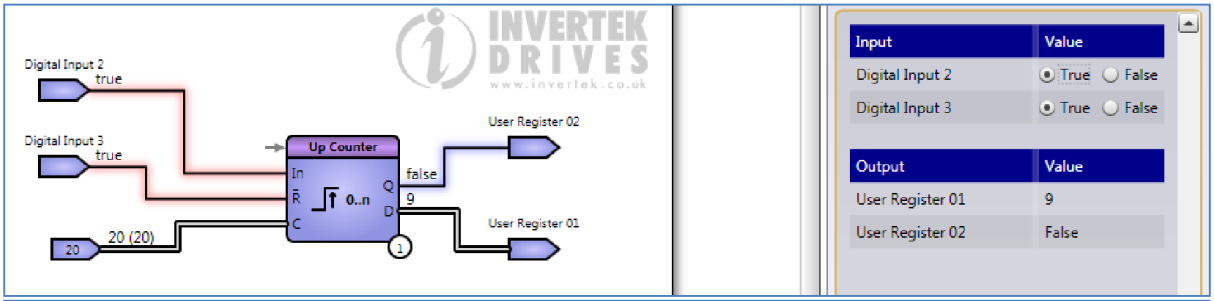
The PWM timer is shown below. The NOT RESET input is connected to digital input 3 to start and reset the timer. The basic clock rate is set within the PWM timer and is set to 0.2 seconds in this case. Input P sets the overall period – in this case, 100 ticks or 20 seconds. D continuously counts from 0 to 100. The output Q changes state when D exceeds C. Therefore the value of C between 0 and 100 controls the width (that is, duration) of the pulse on Q.

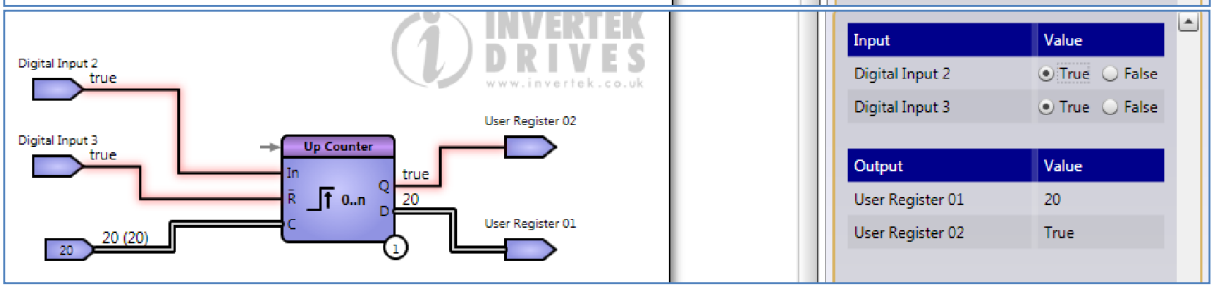


In the example above, D has exceeded 30, so Q is TRUE. Q will be FALSE for 6 seconds and TRUE for 14.
We'll see later that a PWM counter is available as well.

6.6.2. Up Counter

Counter operation is pretty straightforward. In the example below, the counter has been enabled by digital input 3, and counts rising edges on Digital input 2. When the value at input C (20) is reached, output Q changes from FALSE to TRUE. The counter then stops and may be reset using the NOT RESET input.

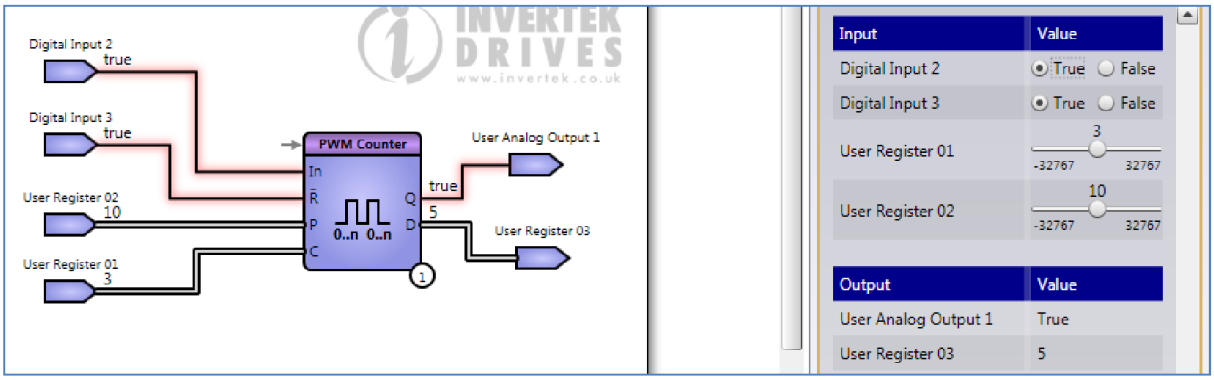




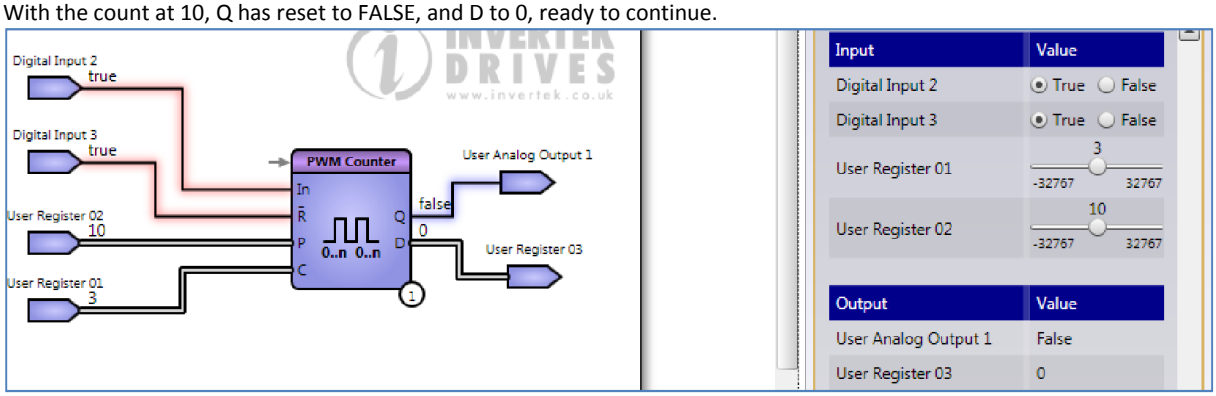
Input C is connect to a constant, but could be connected to variable in a programme.
The down counter works in a similar way, but will count from C to zero and then stop.

6.6.3. PWM Counter

The PWM counter operates in a similar way as the PWM timer, but instead of having a preset internal clock, it is incremented by rising edge input. In the example below, we've got similar settings to the PWM timer, and after 3 rising edges from digital input 2, Q changed to TRUE. Now the count has reached 5.



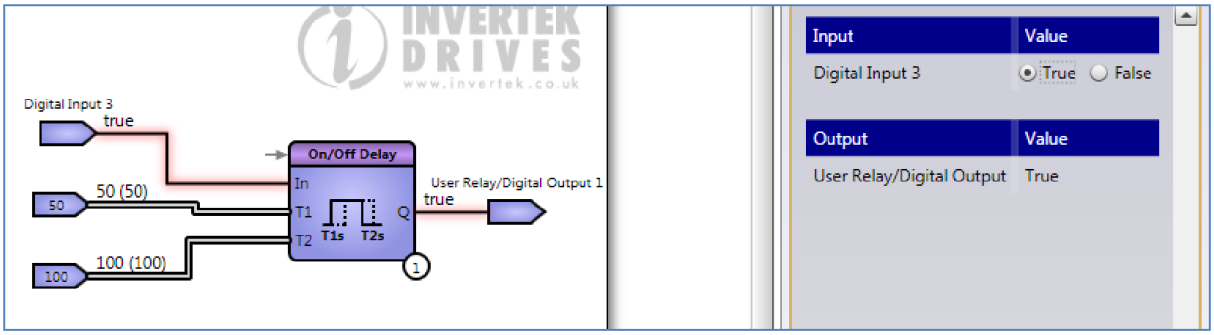
With the count at 10, Q has reset to FALSE, and D to 0, ready to continue.



Incidentally, the physical positions of P and C in the block are exchanged in comparison with the timer. Don't get confused!

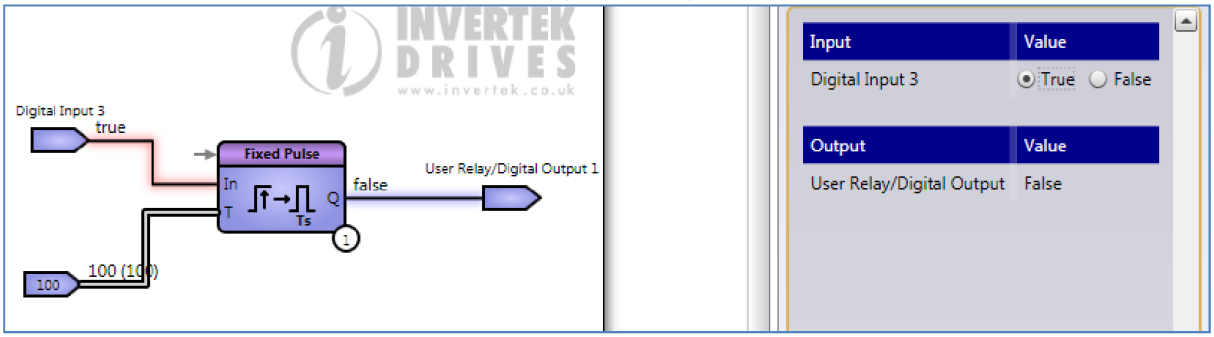
6.6.4. On Off Delay Timer

A variation of the on off delay timer, this timer uses data inputs to set the on and off delay. T1 sets the on delay, and T2 the off. The value of T1 and T2 are in $1/10^{th}$ seconds, so 100 is 10 seconds. To show this simply we have two constants applied to T1 and T2; there is a 5 second delay to the rising edge, and 10 second delay to the falling edge.



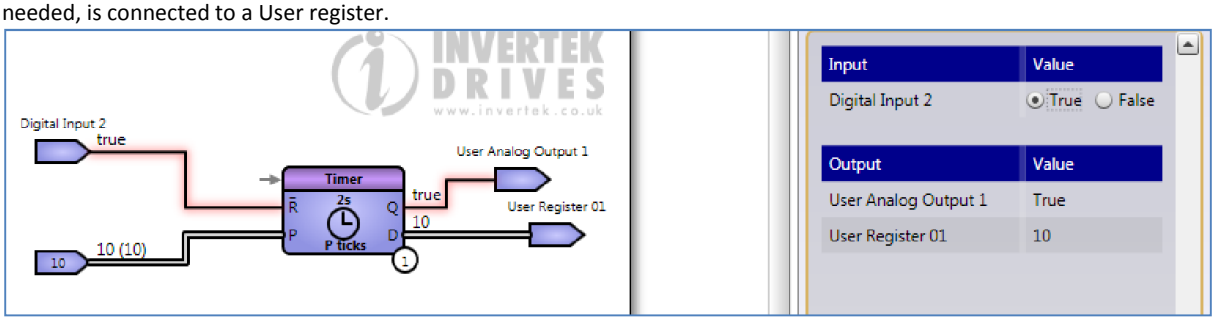
6.6.5. *Fixed Pulse Timer*

The same function is available with the Fixed Pulse Timer (Monostable)



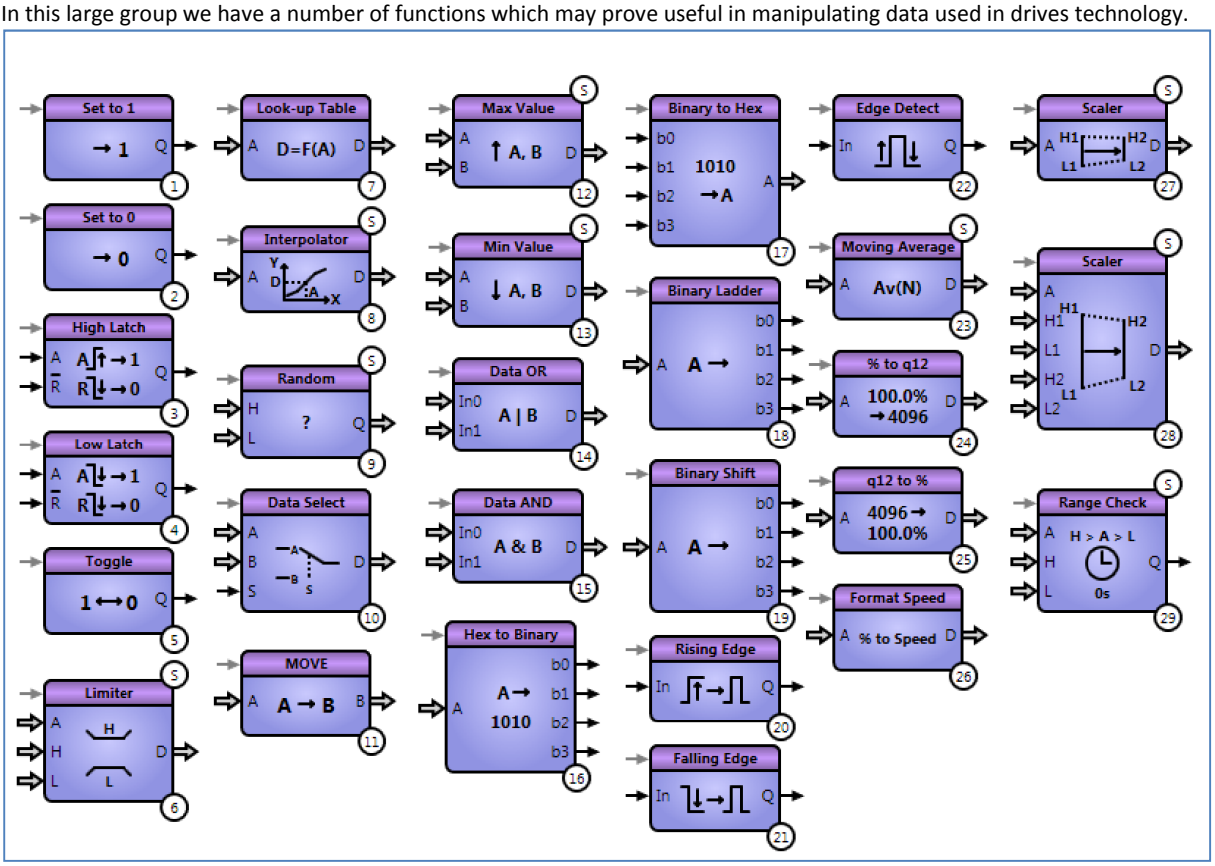
6.6.6. *General Purpose Timer*

Again, this timer is similar to the previous one, but the duration is adjustable at data input P. To simplify the example, P has been set to 10, and the tick rate to 2 seconds, so the output will be TRUE after 20 Seconds. The timer is enabled from Digital Input 2, and the data output, if needed, is connected to a User register.



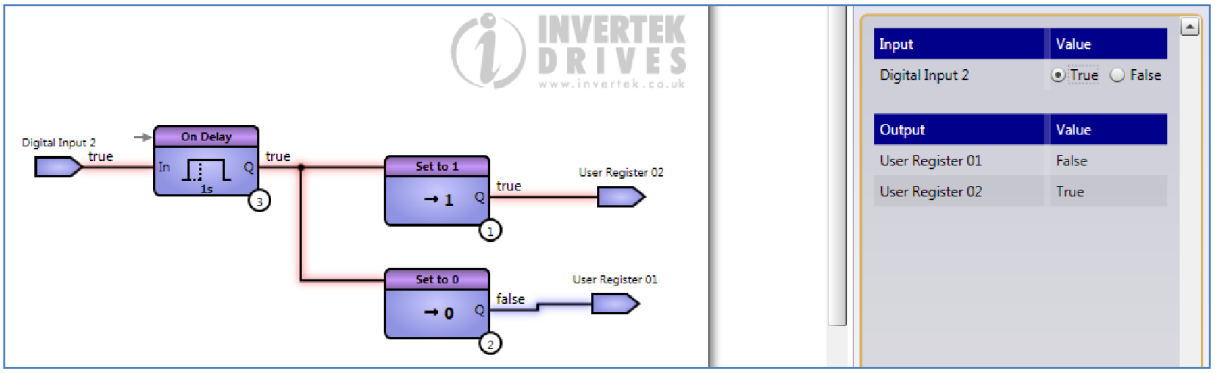
6.7. Data Handling

In this large group we have a number of functions which may prove useful in manipulating data used in drives technology.



6.7.1. *Set to 1, Set to 0*

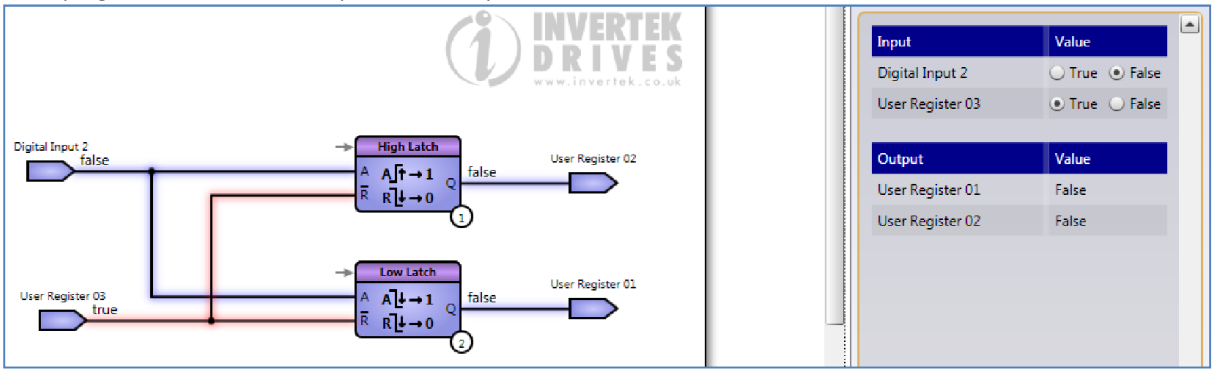
These functions are useful to preset conditions or to set conditions that shouldn't change. They have no inputs, but could be activated using the enable input if necessary. In the example below, digital input sets User registers 2 and 1 after a short delay.



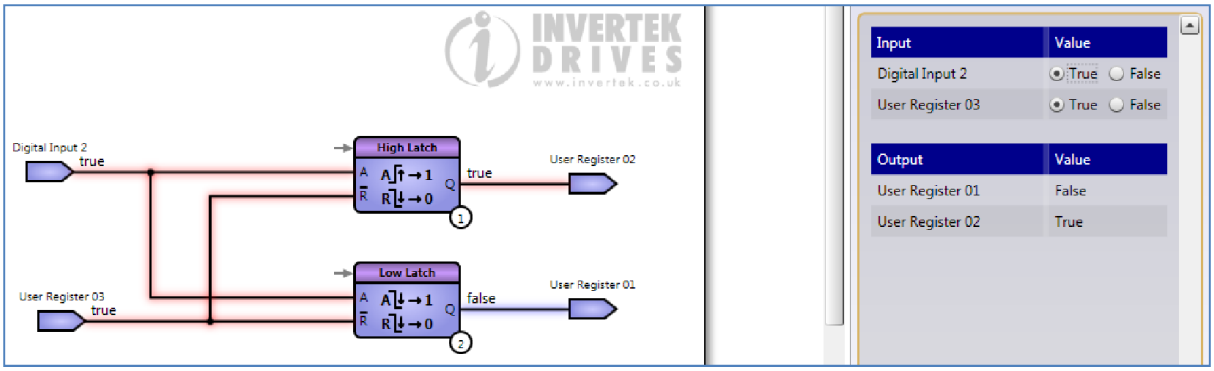
6.7.2. *High and Low Latches*

Latches are edge sensitive; that is, the latch responds to a change of state, not what follows or precedes the change. The simple latches here respond to a rising edge (FALSE to TRUE) and falling edge (TRUE to FALSE) respectively. A FALSE signal on the NOT RESET input will always force the outputs to FALSE, so for latch operation this should be TRUE.

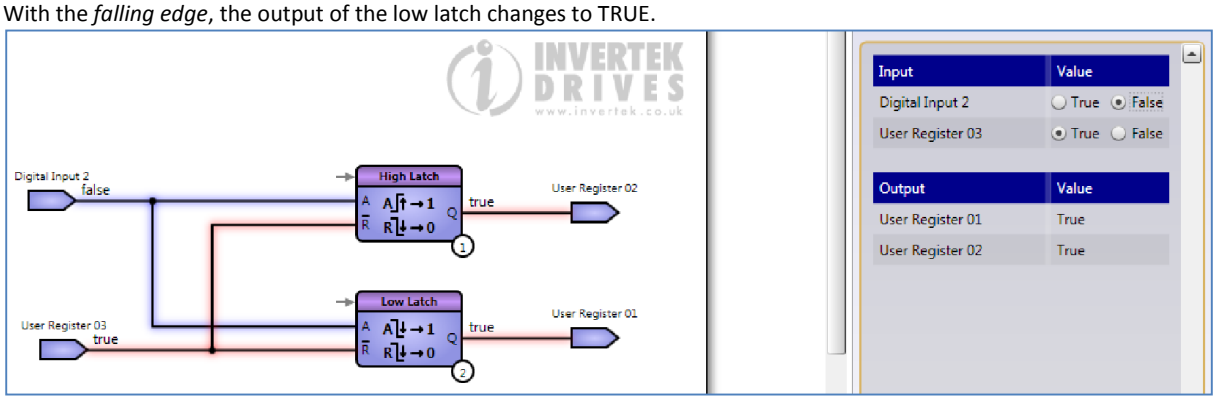
In the programme below, both outputs are initially FALSE.



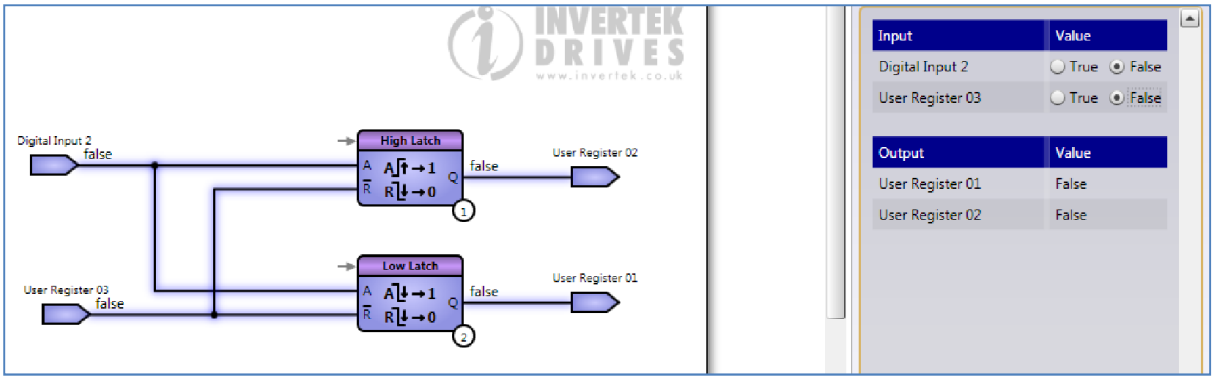
With the *rising edge* of digital input 2, the output of the high latch changes to TRUE.



With the *falling edge*, the output of the low latch changes to TRUE.



Both latches hold those conditions until reset by the low (FALSE) state of the NOT RESET inputs

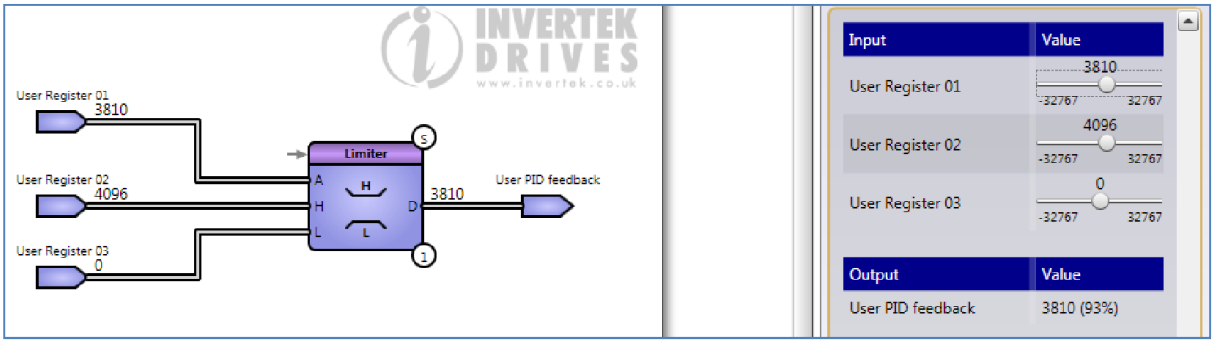


6.7.3. Toggle

The Toggle function switches between TRUE and FALSE continuously when enabled.

6.7.4. Limiter

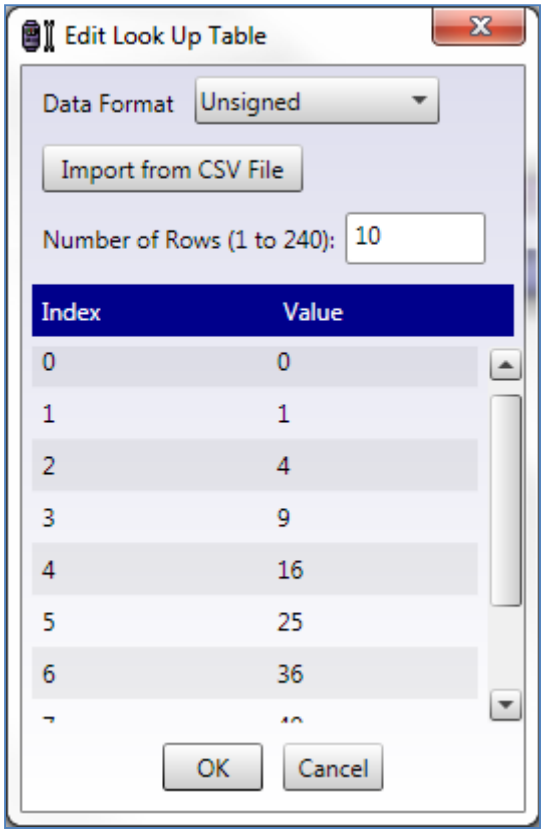
The limiter ensures that the output value D follows the input value A, but limits the output between the high input H and the low input L. If the input H is, for some reason, less than L, there are various options available, such as tripping the drive, outputting A etc. In the example below, we've limited the value from User Register 01 to between 0 and 4096 so the output is suitable for use in the User PID Feedback.



The Limiter can also be set to handle unsigned numbers.

6.7.5. Look Up Table

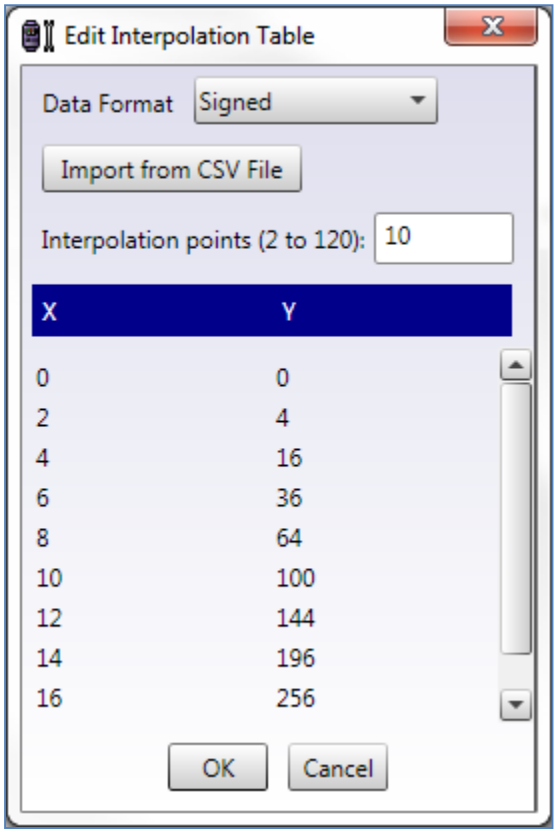
There are several functions available for linearizing and rescaling data. This block allows a table of values to be stored for looking up. The table can also be imported from a CSV file. In the table below, the output is the square of the input.



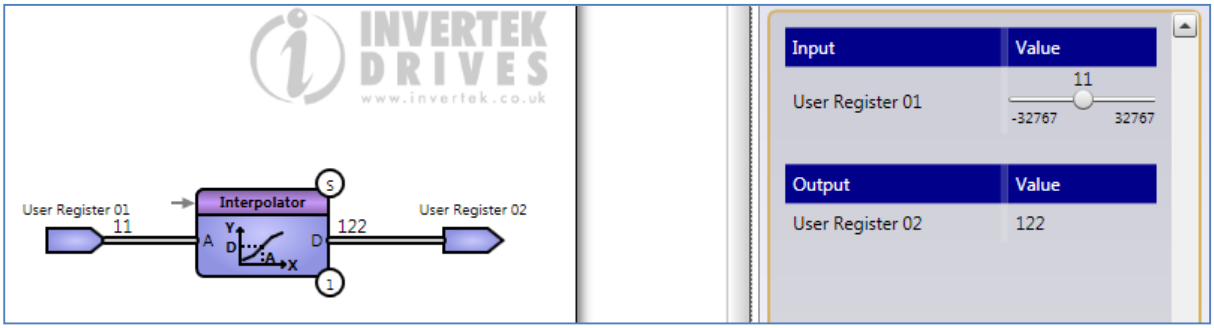
The result is shown below. The table can have up to 120 rows. The function will return the maximum or minimum value if the input to the table is out of the index range.

6.7.6. Interpolator

The interpolator function operates in a similar way to the look-up table, but allows intermediate values to be handled. In the table below we have squares of even numbers:

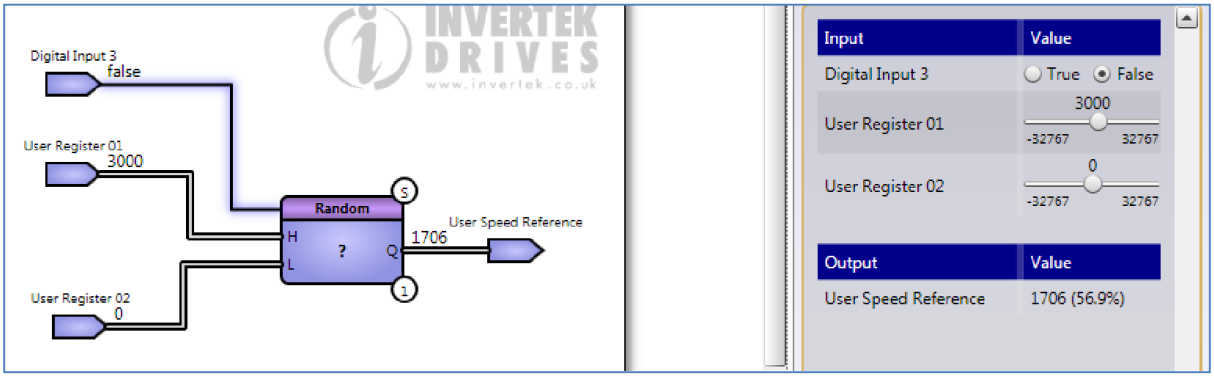


Tables of more complex functions can be similarly constructed. The interpolation result shows a slight calculation error:



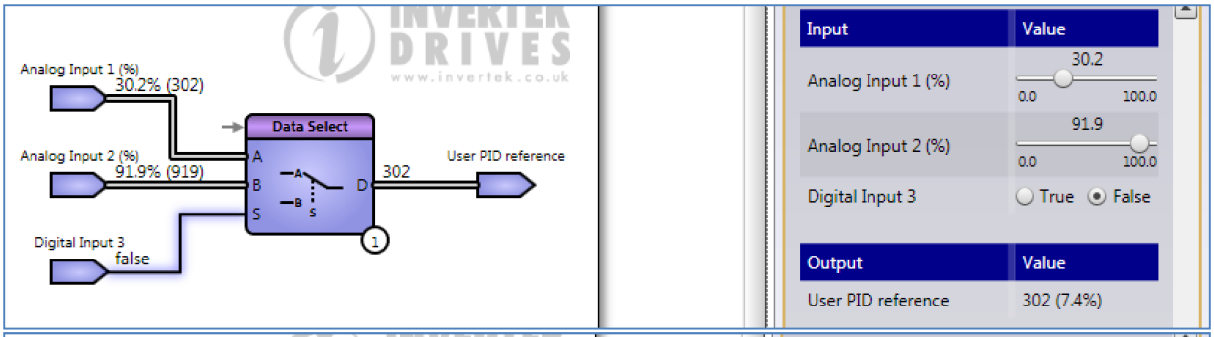
6.7.7. Random Number Generator

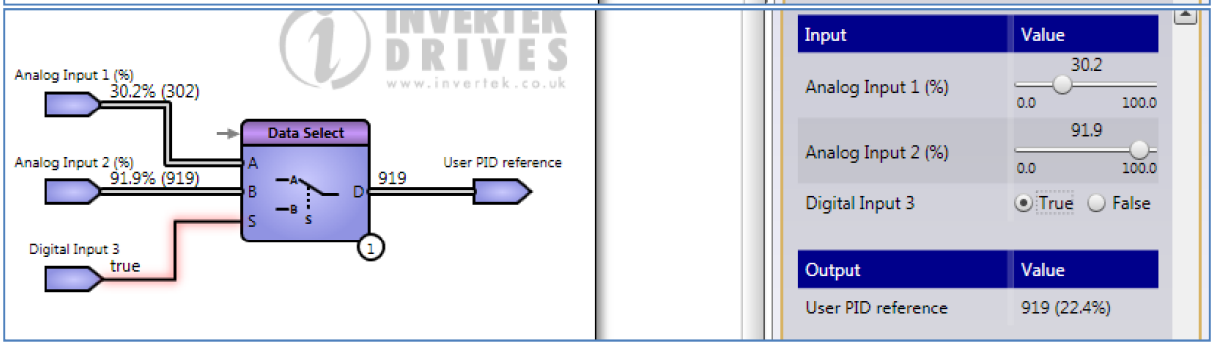
The Random number function block returns a random value between the high and low limits. In the example below, a random speed reference between 0 and 3000 is generated. The block runs continuously, so if a single value is required, the enable function can be used to start and then stop the generator, as here.



6.7.8. Data Select

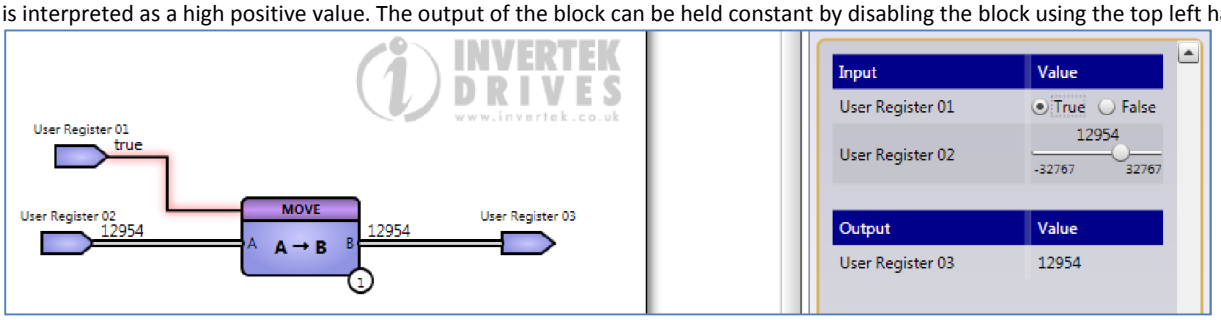
This block makes switching between two data signals easy. In this case we are switching between two analog inputs. Notice that the Analog inputs in this case are percentage, and therefore scale between 0 and 1000. The PID reference is 0 to 4096, so some additional scaling would normally be required (see later)





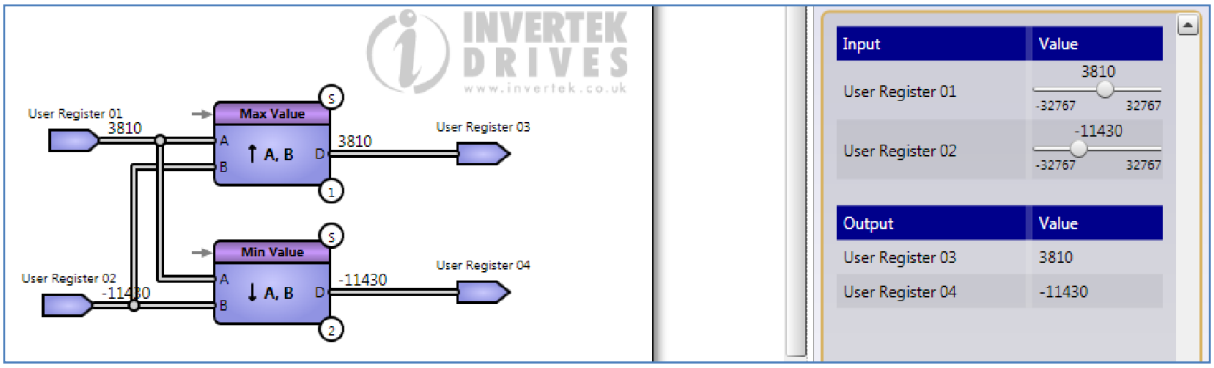
6.7.9. *Move*

With this function block a data value is simply transmitted when the block is enabled. The Move function is unsigned, so a negative value input is interpreted as a high positive value. The output of the block can be held constant by disabling the block using the top left hand input.

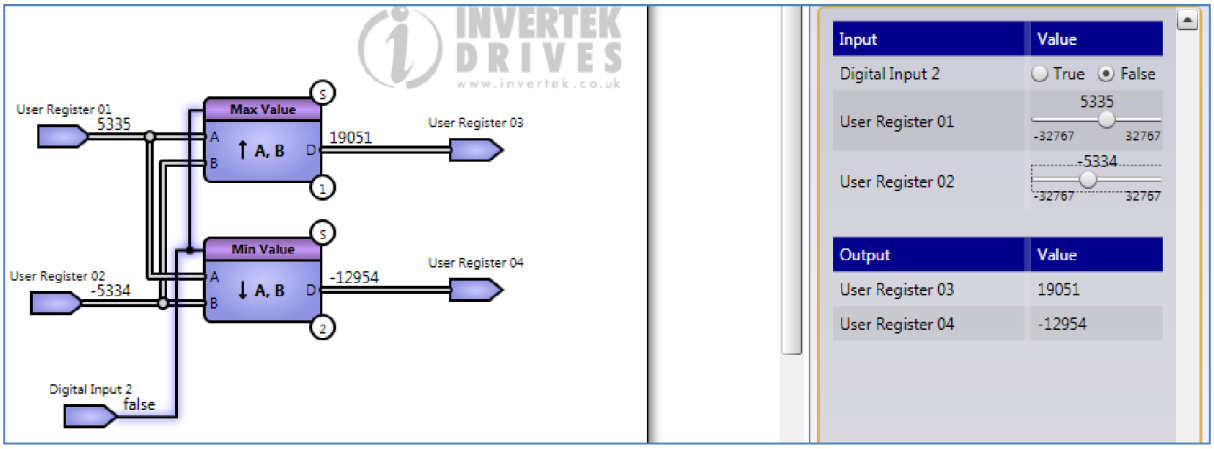


6.7.10. *Maximum and Minimum Values*

These blocks deliver the minimum and maximum data values as shown below.

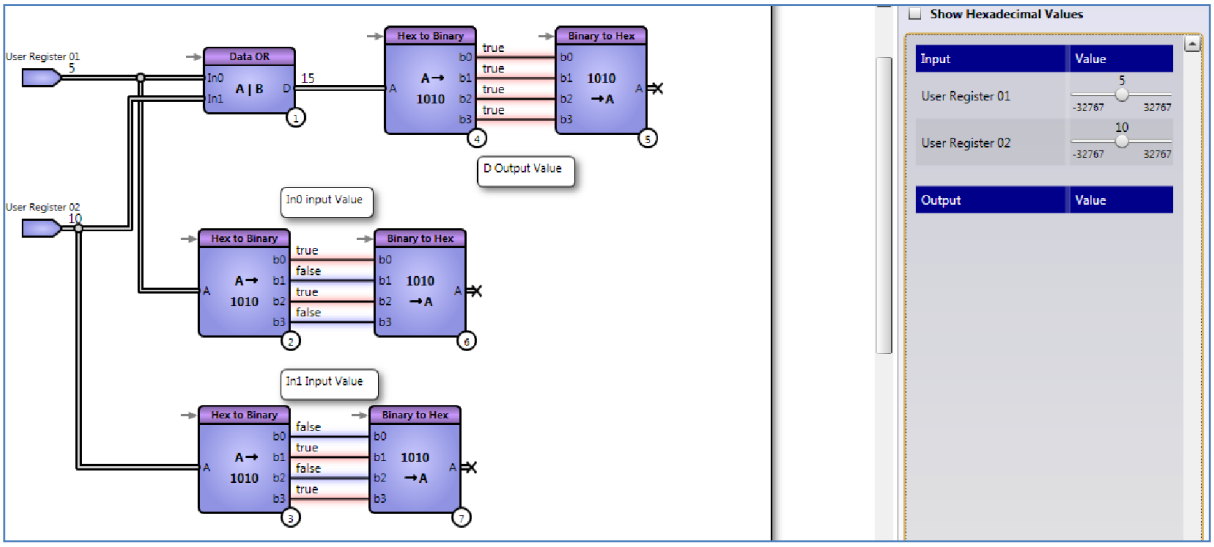


Again, the data values can be held using an enable input. In the example below, the values were sampled using the enable input, and the input value have now changed, while the outputs are fixed.



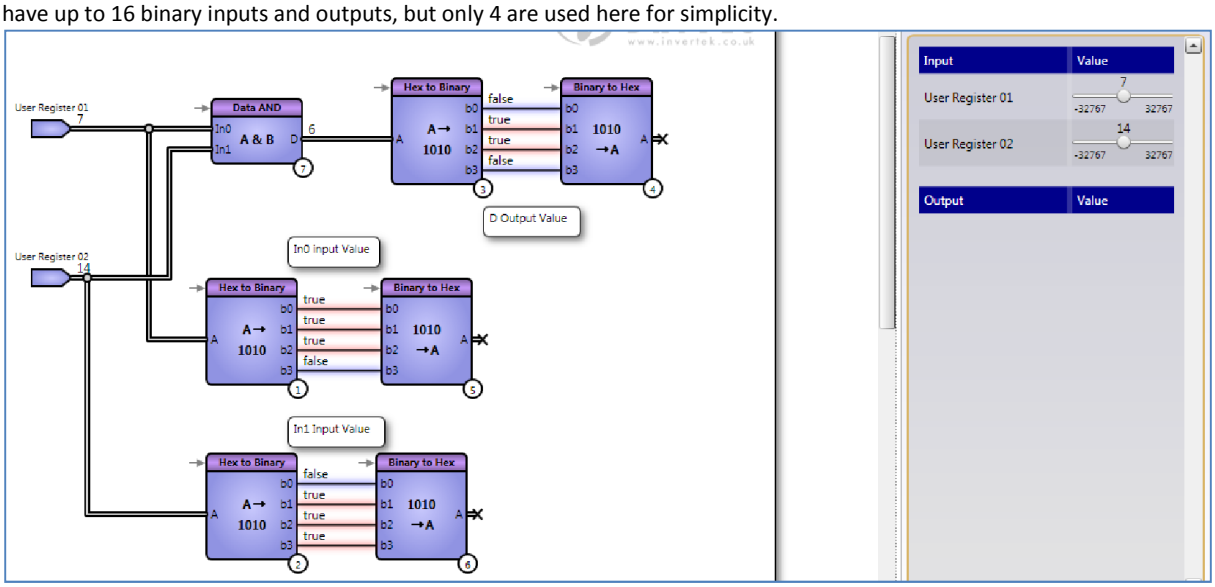
6.7.11. Data OR and Hex to Binary Conversions

The Data OR function is demonstrated below, and Hex to Binary Conversion functions are used to clarify. The two inputs and the output of the Data OR block have Hex to Binary conversion blocks attached so we can see the actual binary bits. The outputs of the Conversion blocks are not used (by right clicking on the output they can be terminated). So when input In0 value 0101 (5) is OR'ed with input In1 value 1010 (10) the result at the output D is 1111 (15).



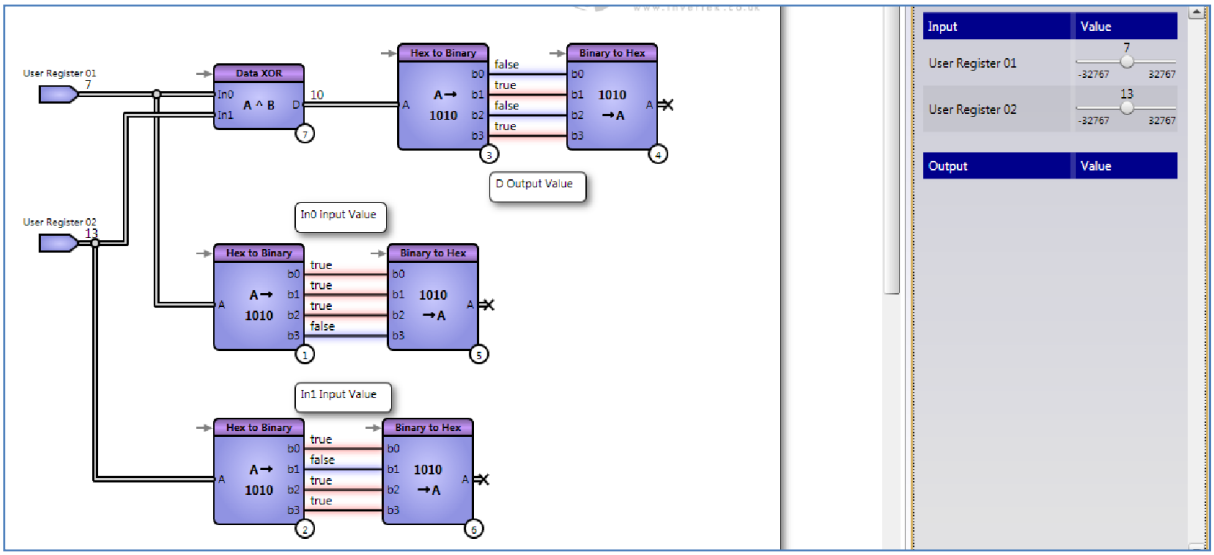
6.7.12. Data AND Function

This operates in a similar way. In this example 0111 (7) is AND'ed with 1110 (14) with a result of 0110 (6). The Hex to binary converters can have up to 16 binary inputs and outputs, but only 4 are used here for simplicity.



6.7.13. Data EXOR Function

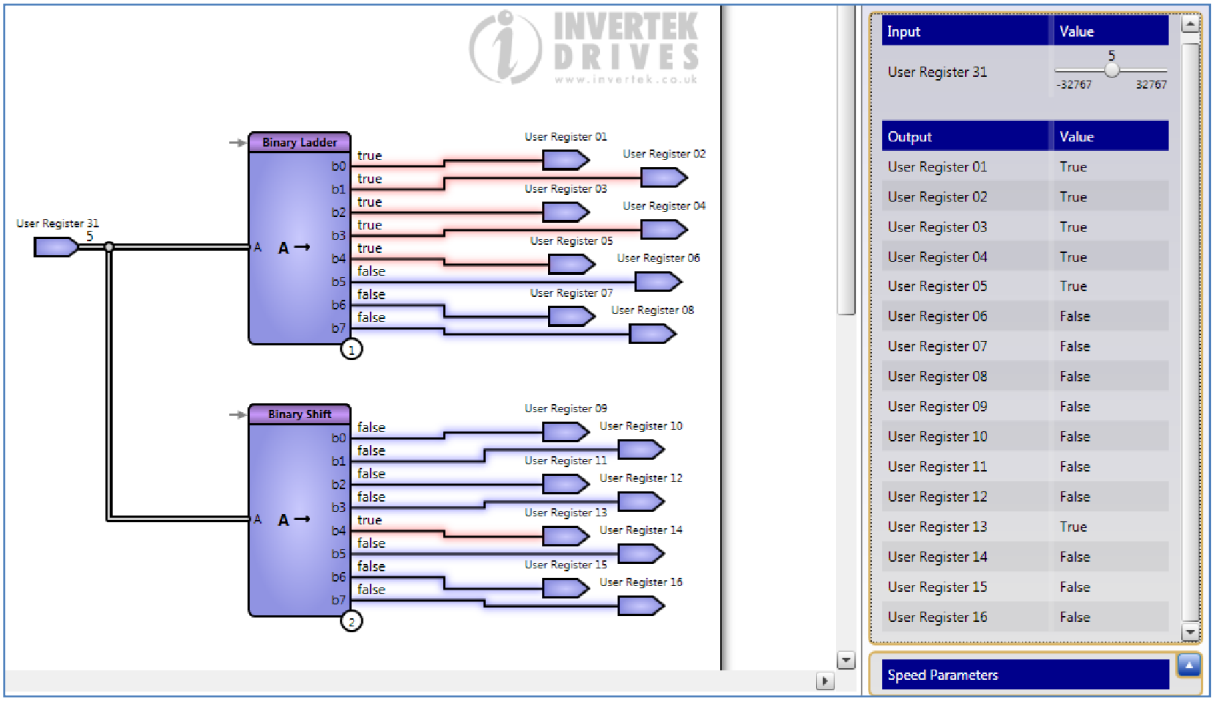
The data EXOR function is shown below.



Where the inputs are the same, the output is FALSE, where they are different, the output is TRUE.

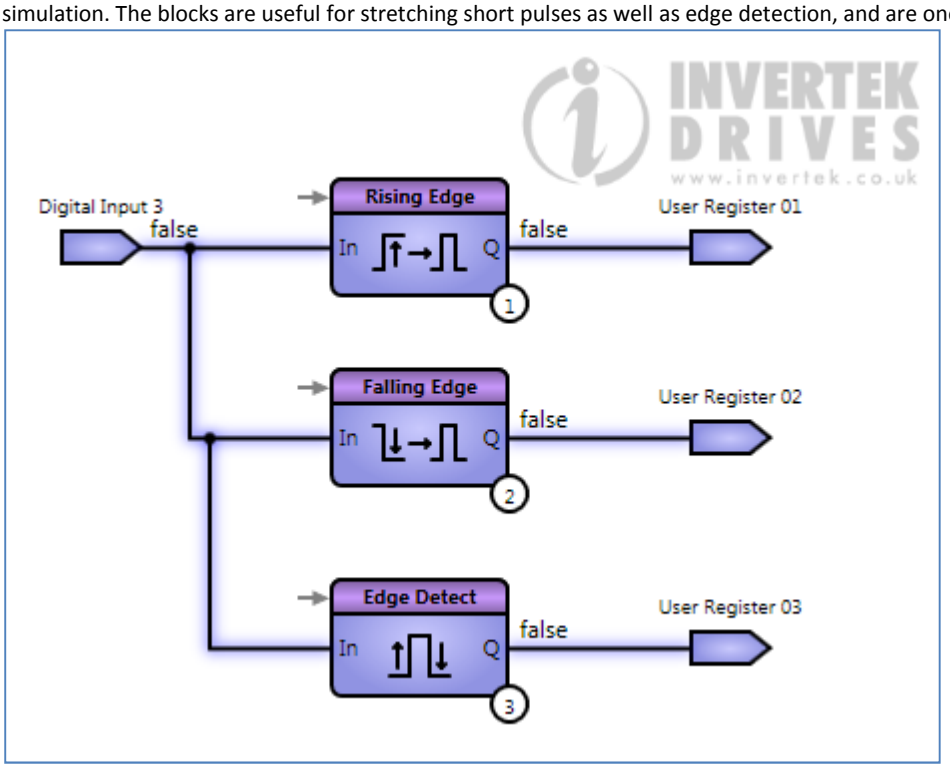
6.7.14. Binary Ladder and Binary Shift

These function blocks will convert the data input into either a binary ladder, where the outputs become progressively all TRUE, or a binary shift, where only one output is TRUE which is shifted depending on the value of input A. The simulation below, using 8 bit blocks (up to 16 are possible) shows this for and input of 5.



6.7.15. Rising, Falling and Edge Detect

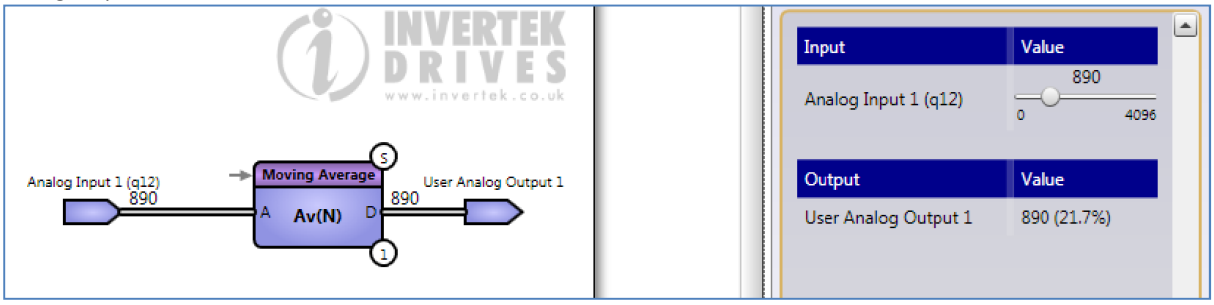
These three function blocks are difficult to simulate as they produce short pulses from edges. The short pulses don't really show on the simulation. The blocks are useful for stretching short pulses as well as edge detection, and are one PLC cycle long.



6.7.16. Moving Average

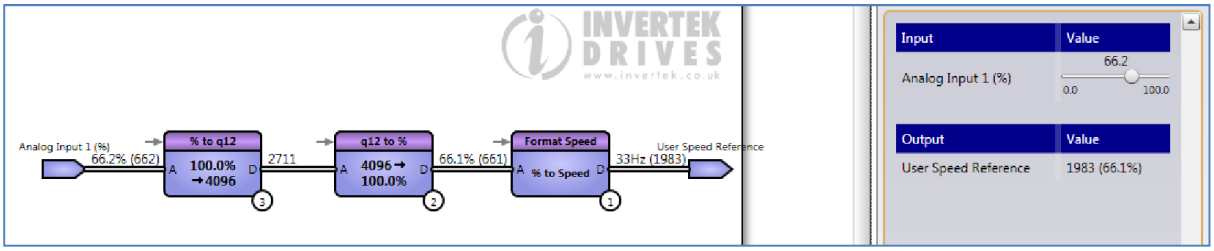
The function calculates the average of a changing data value based on a variable sample number which can be set between 2 and 16.

The sample rate is based on the PLC cycle time. So during simulation the output will track the input quickly, a short lag showing only when the sample number is set to the maximum of 16. However, in real applications it is a useful block for removing unwanted sudden changes in signals from a transducer or similar input. In the example below the signal from the analog input is filtered by the block and connected directly to the analog output.



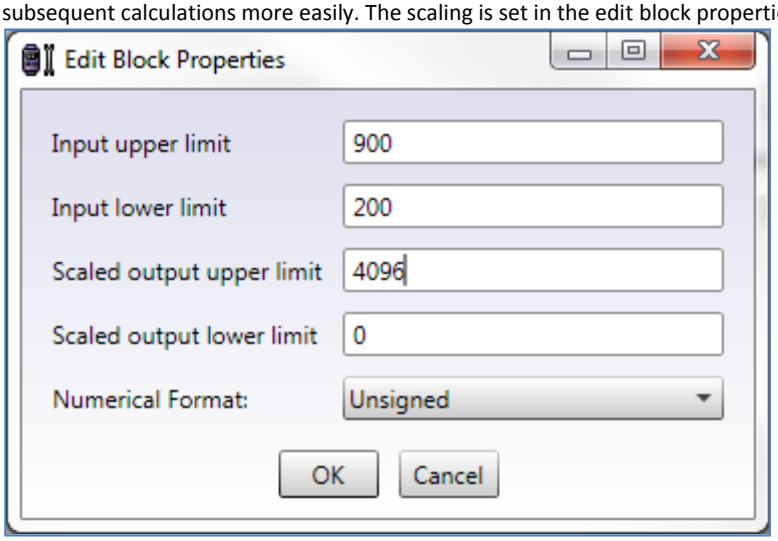
6.7.17. Data Conversion Blocks

There are three blocks available to facilitate conversion from percentage, 12 bit numbers (0 – 4096) and frequency (0 – 3000). As previously mentioned, many signals derived from inputs are in 12 bit format, and frequency is scaled between 0 and 3000 (this makes working with 50 and 60Hz easier). Data conversion between these is carried out by the conversion blocks shown in the example below. The output from Analog input 1 is chosen to be in percentage. The first block converts this to a 12 bit value. The second block converts this back to percentage, and the third block converts from percentage to 0 – 3000, which corresponds to 0 to 50 or 60Hz with default drive settings. So the input of 66.2 is converted to 33Hz via all three conversion blocks.

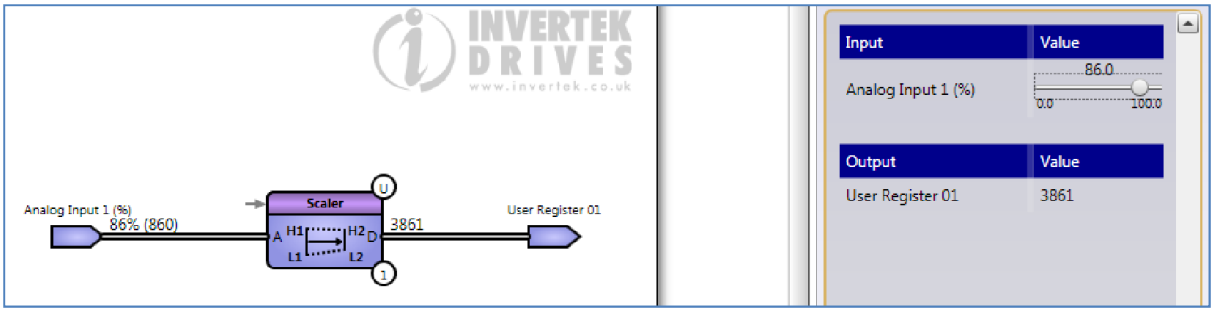


6.7.18. Scaler

This block is a general purpose scaler. Suppose we have a transducer that has a range of 2 to 9 Volts and is connected to analog input 1. By setting the scaling so that 200 to 900 (remember the programme processes 0 – 100% as 0 – 1000) is rescaled to 0 to 4096 we can then handle subsequent calculations more easily. The scaling is set in the edit block properties box:



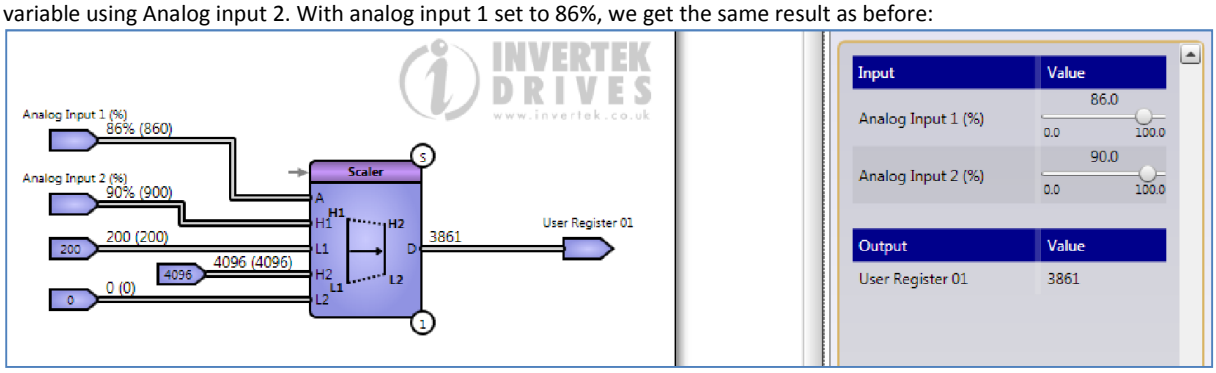
And the resultant scaling can be seen in the simulation. This solution is quicker for linear scaling than the interpolation blocks and look up tables described earlier.



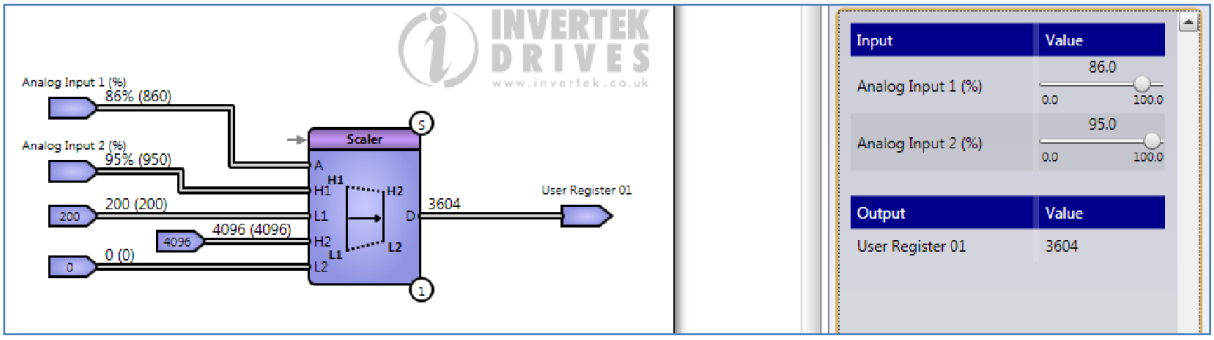
6.7.19. Adjustable Scaler

This scaler operates in a similar way to the previous block, but the scaling limits that were fixed in the block properties are now variables and therefore inputs to the block. This means the scaling can be adjusted as part of the programme, for instance for automatic scaling and calibration.

In the example below, we've kept the output scaling the same by using constant inputs. The lower limit is also fixed, but the upper limit is now variable using Analog input 2. With analog input 1 set to 86%, we get the same result as before:

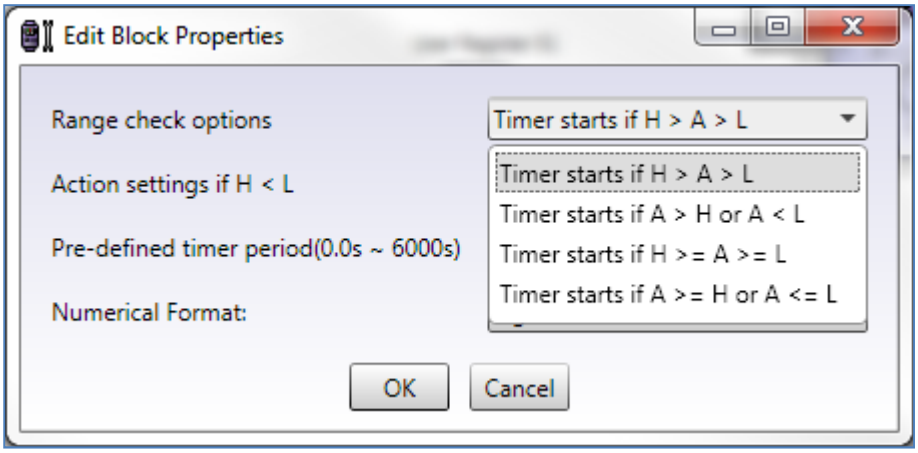


However, changing analog input 2 will change the scaling accordingly:

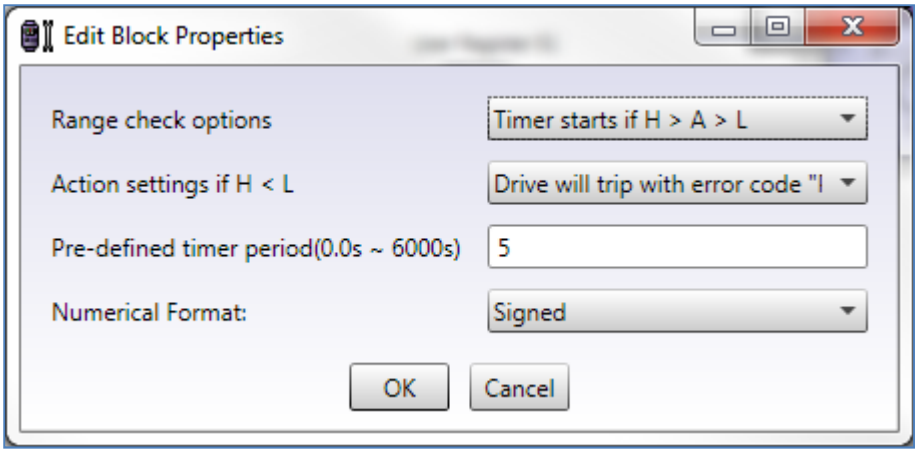


6.7.20. Range Check

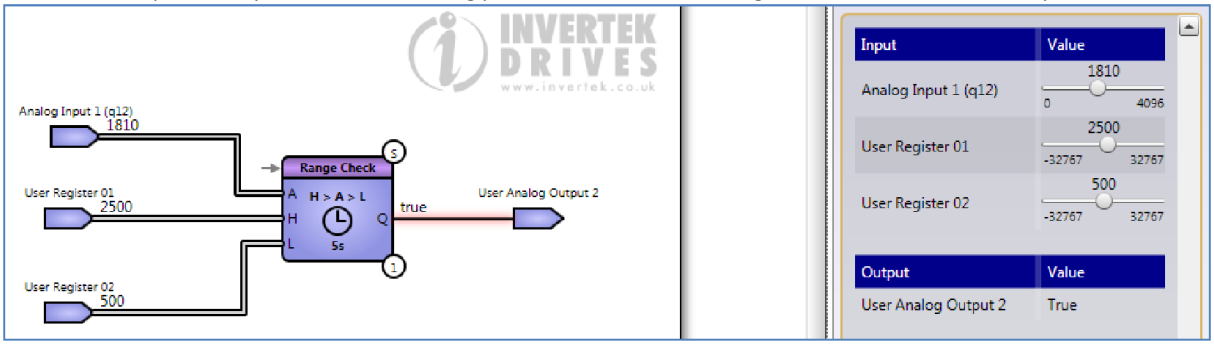
This function block will monitor a variable and Compare it with high and low range. Various range option settings are available. Here we choose the simplest option, monitoring if the input A lies between H and L.



The output will change from FALSE to TRUE if A is in the limits and has remained so for the time set in the timer.

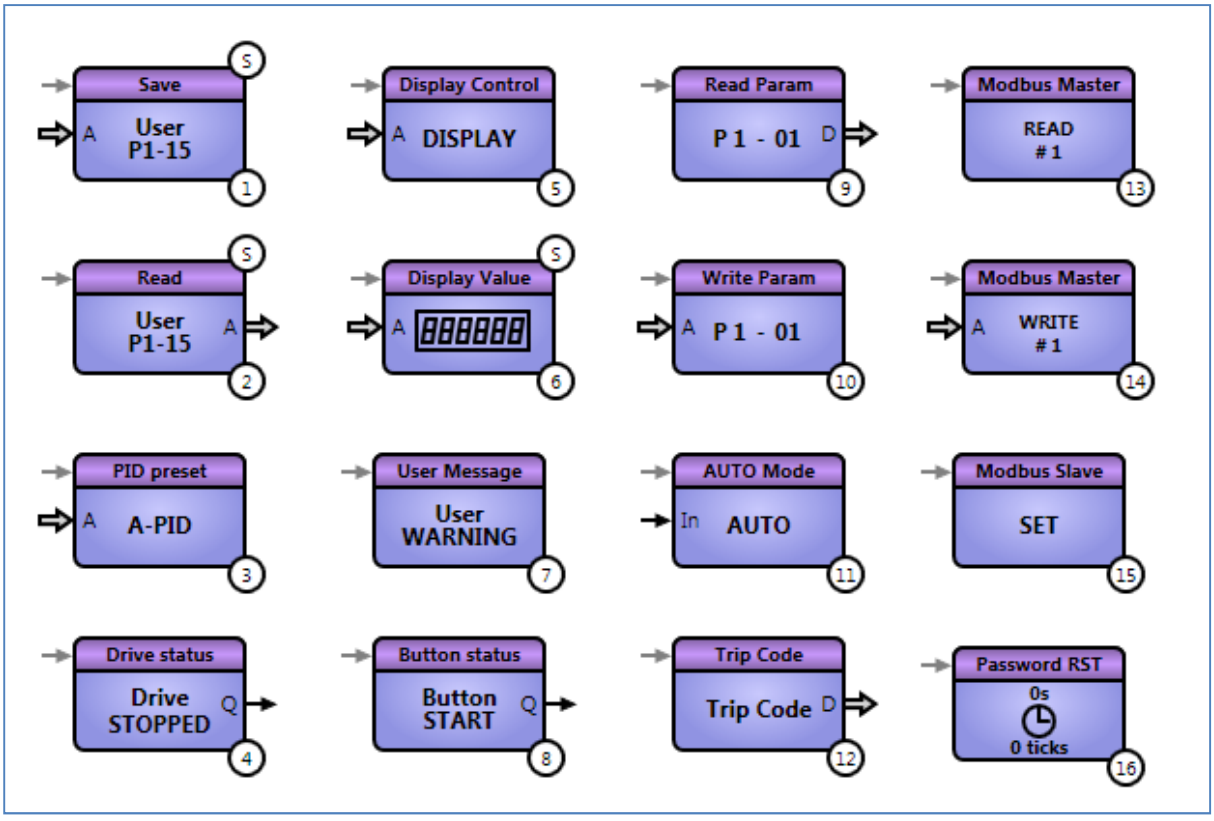


In this example the timer is set to 5 seconds (possible settings 0 – 6000 seconds in 0.1 increments). So in the simulation below, the output Q changed to TRUE after analog input 1 came within the range 500 to 2500 for 5 seconds.
This function is particularly useful for eliminating problems with start or changeover conditions in control systems.



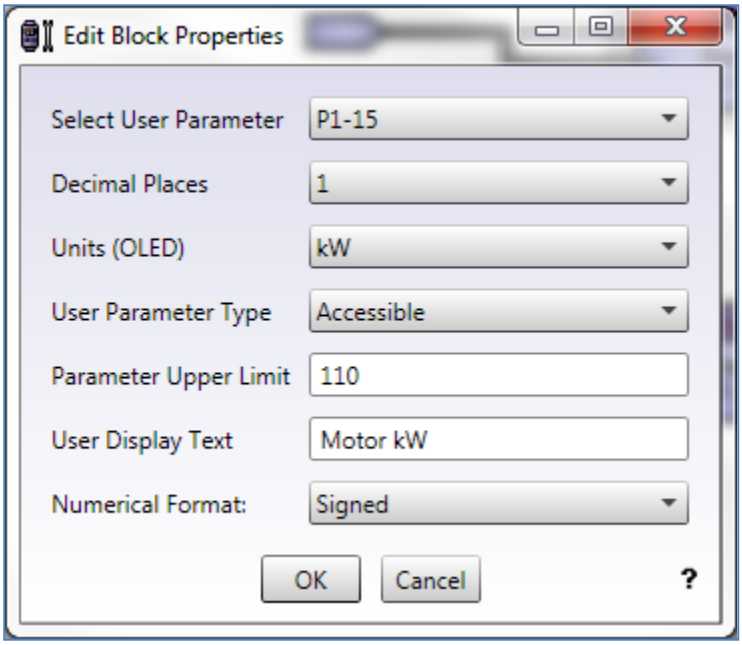
6.8. Drive Functions

This group of functions is specific to the drive and allows parameters to be created, adjusted as well as the display to be customised. These functions can be harder to simulate as they are related to drive features.

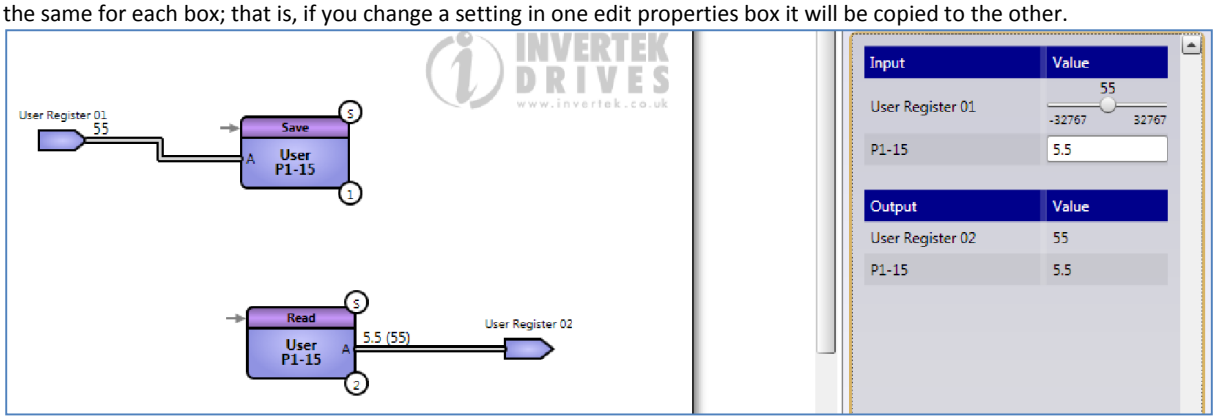


6.8.1. Save and Read User Parameters

These two function blocks allow user parameters to be created. The parameters can then be used within the programme and, unlike other registers, can be read and written to by the operator using the display and panel buttons. Either of the first two function blocks can be used to create the new parameter, which can be defined using the edit block properties menu.



In this example, we have set the maximum parameter value to 110 and set one decimal point for the display of the motor kW. The value of parameter will be displayed with a decimal point, and if the drive has an OLED display (which can display text and units) then “kW” will appear in the display. So now the user can select parameter 1-15 and choose (for example) the power of the connected motor. Presumably this would be used in other parts of the programme.
In the example below, because the two function blocks have the same parameter numbers, the settings in the Edit Properties box as above are the same for each box; that is, if you change a setting in one edit properties box it will be copied to the other.



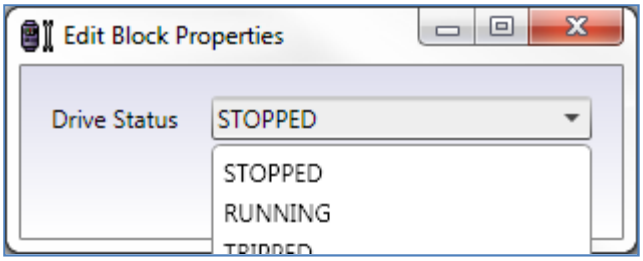
Note that although the display will show 5.5, the value in Register 01 is 55, and this is the value that will be used in other parts of the programme. The maximum value of the parameter has been set to 110, that is, 11kW in the display. The simulation also conveniently shows the parameter display value.
The parameter has been selected as accessible from the Edit Properties menu. This means the user can view (and therefore change) the parameter. All parameters created will be accessible up to and including the highest parameter set as accessible. So if P1-15,16,17, 19 are set as accessible, and P1-18 and P1-20 are set as hidden, P1-15 to P1-19 – including P1-18 – are accessible. Otherwise there would be gaps in the list. This is explained in the text that appears if the mouse hovers over the "?" in the edit properties menu shown above.

6.8.2. PID Preset

This function block sets the value of the PID Integrator to the value of input A on the block. When using closed loop control the integrator value may cause unexpected response, especially when starting up or adapting the control. If the function block editor is used as part of a closed loop controller pre-setting the integrator value may be useful.

6.8.3. Drive Status.

This block will return a TRUE value if the selected drive condition occurs. The following conditions may be selected by editing the block properties:
STOPPED
RUNNING
TRIPPED
HEALTHY
AUTO MODE
HAND MODE
INHIBIT
External 24V
FIRE MODE
STANDBY
U/O TORQUE
PUMP STIR



This function is particularly useful when a control system expects a certain response, but the status of the drive prevents this.

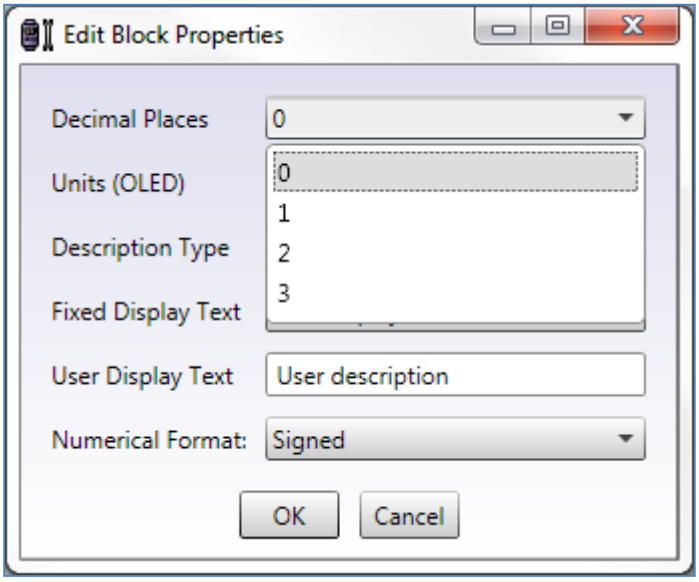
6.8.4. Display Control

The drive can display several different drive values; these are often selected by the front panel push buttons. This function block allows the values to be displayed dependent on the value of the data input A.

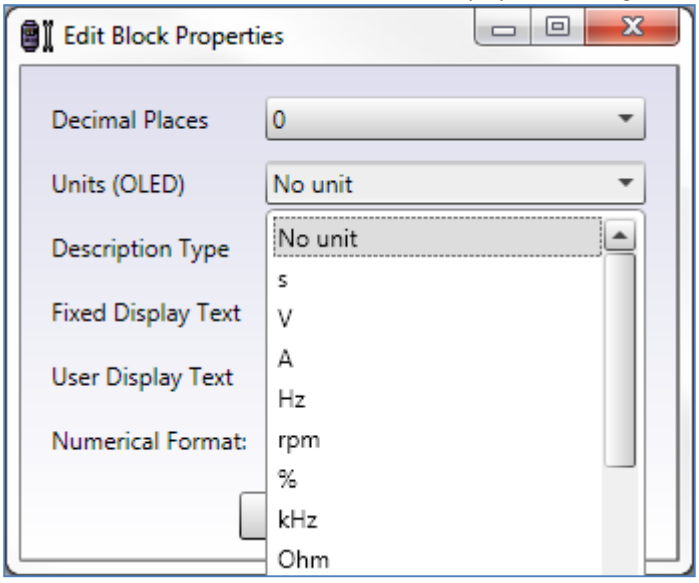
A Value	Display shows:
0	Speed in Hz
1	Motor Current
2	Motor Power
3	Scaled Display 1
4	Scaled Display 2
5	Speed in RPM

6.8.5. Display Value

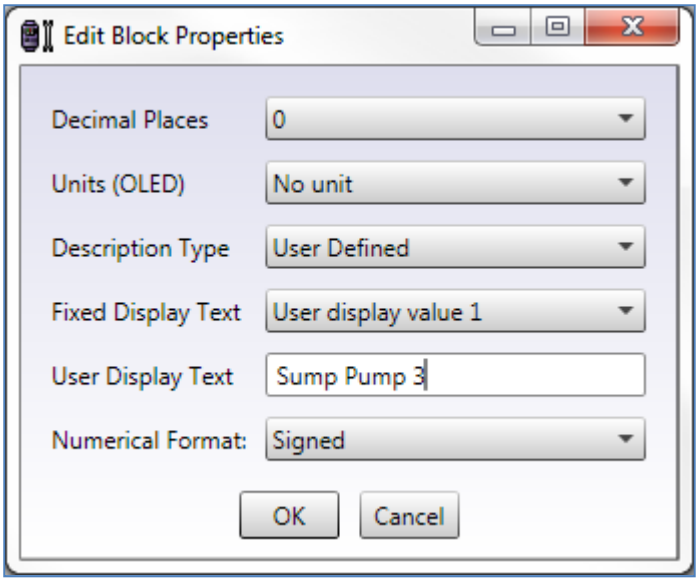
This function allows the drive to display messages, values and units in the OLED display fitted to a drive. This function does not work with an OLED display that is remote from the drive, or with the seven segment display.
The edit box allows the display to be customised. The number of decimal places can be selected:



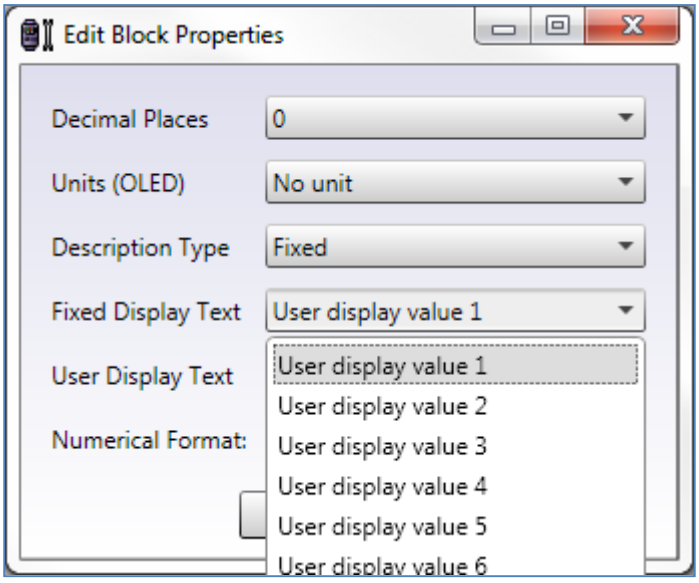
There is a wide selection of units that can be displayed, including bar, Celsius, CFM etc.:



If a user defined text is selected, the text can be defined:



This text will appear in the OLED display. If the description type is set as 'fixed', the OLED display will show 'PLC User Value X', where X can be selected as 1 to 8:



6.8.6. User Message

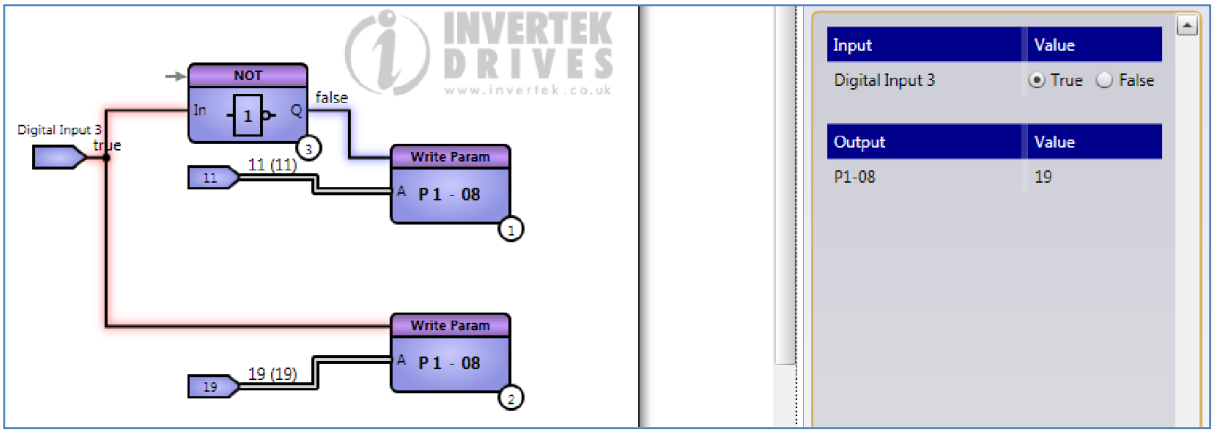
This function simply writes the different Text options to the OLED display. The following text appears in the centre of the display:
WARNING
ALARM
TRIP
BLANK (i.e. no display)

6.8.7. Button Status

This function block will monitor a button and return TRUE if the button is pressed. Clearly this can enhance the function of buttons and should be used with care.
The following buttons can be monitored
START
STOP
NAVIGATE
UP
DOWN
HAND
AUTO
The buttons' normal functions are unchanged.

6.8.8. Read and Write Parameter Values

These function blocks allow parameters to be written to and read as part of the programme. This is could be used to load different parameter settings for different applications or different motors. In the example below, the digital input is switching between two values of P1-08 that set the current limit of the drive.



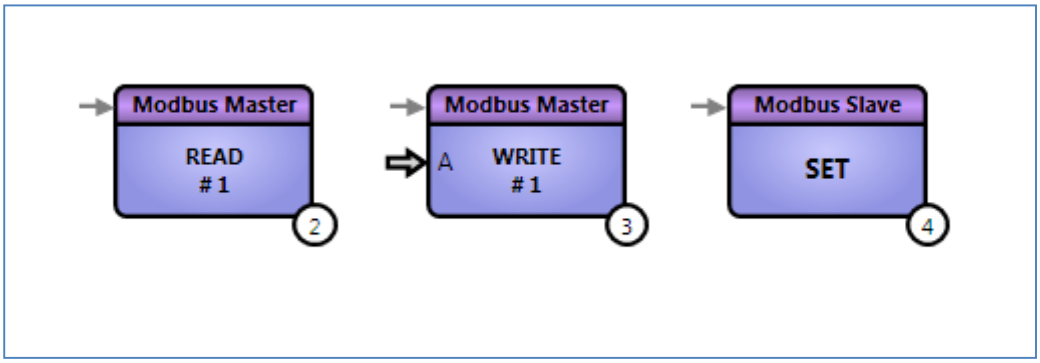
6.8.9. AUTO Mode

This function will put the drive into AUTO mode when a TRUE signal is applied to the input of the block. Switching between HAND and AUTO is a frequent requirement in HVAC control systems.

6.8.10. Trip Code

This function block will return the value of the last trip code. A description of the faults and the corresponding numbers is given in the manual, and also by hovering the mouse over the function block.

6.8.11. Modbus Functions – Read, Write, set as Slave



These functions allow the drive to operate as a Modbus Master or Slave. Modbus is an open communications system that allows the drive and other equipment to be controlled and monitored over an RS 485 (or other serial) interface. The drive has Modbus built into its communication system. The detailed functioning of Modbus is beyond the scope of this document.

In a Modbus system, a Master device will control the Slaves (i.e. other drives or devices) by sending appropriate messages; Slaves can generally only respond to commands and cannot initiate messages. Normally the drive operates as a slave and is controlled by an external Modbus Master.

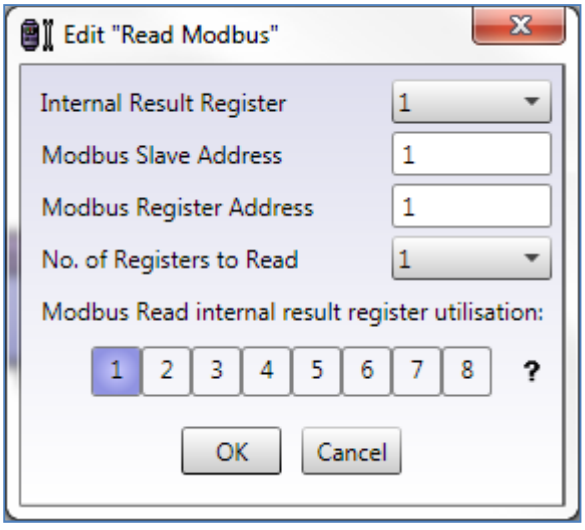
As Modbus works by reading from and writing to registers within the equipment (the drive in this case) it is possible, using these function blocks, to set up the drive to act as a Master. When operating as a Modbus master, the drive has available eight read and eight write registers. The Read function block will use the Modbus interface to read from a slave address and register and write the value to an internal read register. This can then be processed and acted upon by the program.

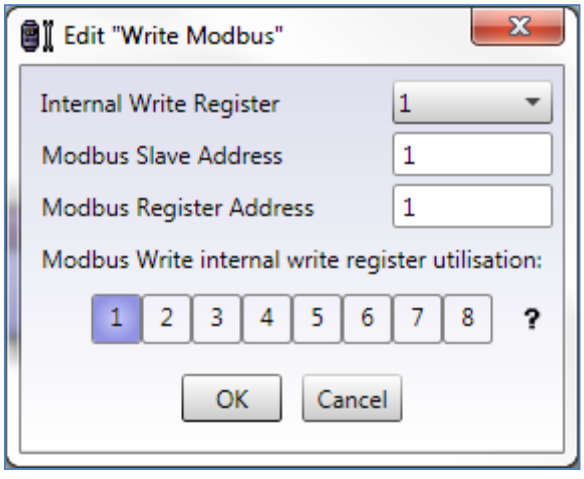
Similarly, the Write function block will take a value from an internal write register and transmit it via the interface to a Slave at the designated address and register, the slave reacting accordingly.

When either of these function blocks run, the drive is configured as a Modbus master, and will remain so until the Modbus Slave function runs.

There is more information available by hovering the mouse over the function block as well as over the "?" on the edit menu.

The Edit menus for the Read Modbus and Write Modbus function blocks are shown below.

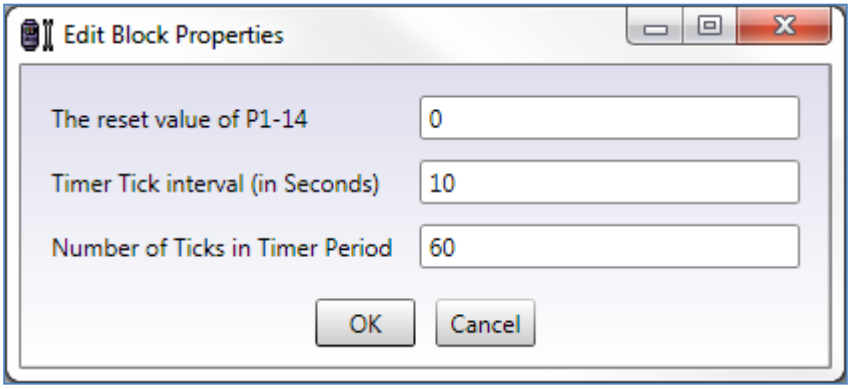




6.8.12. Password Reset

This function block will automatically reset the value of the password for parameter P1-14, which allows access to some or all advanced parameters. The block can be used to enable access by writing the access value after a delay, or more usually to ensure the access is *not* left enabled after the user has set the access code and left the drive.

As with other timers, the duration of a 'tick' is set, and the number of ticks before the password is reset is also entered in the edit properties box. The reset value of P1-14 can also be selected here. In the example below the value of 0 (the default value) will be written after 600 seconds (10 minutes).

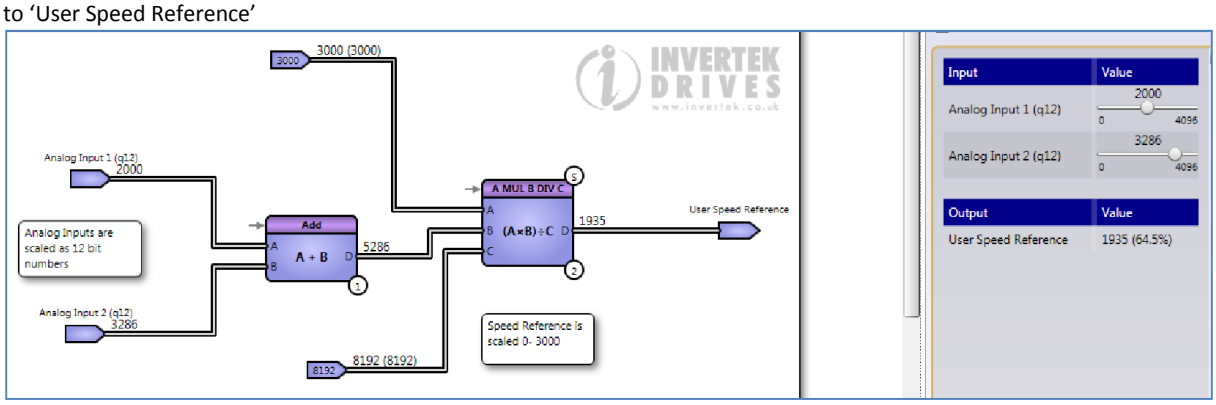


7. Practical Program Examples

The following programmes have been prepared to demonstrate how some of the function blocks connect together to create practical programmes. The necessary parameter settings are also included. They are not intended to represent full and complete solutions; it will always be necessary to consider all possible operating conditions and adjust the programme accordingly. However, the programmes can be simulated and uploaded to drives to demonstrate their features.

7.1. Analogue Input addition with Scaling

In this example, the two analog inputs are added together and are scaled using the multiply divide block. As each analog input is a 12 bit number, the 16 bit multiply divide block will handle the result of the addition (maximum value 8192) without overflow. The result is used to control the output frequency of the drive and therefore the speed of the motor.
The scaling is carried out by dividing the input by 8192 and multiplying by 3000 to match the scaling of the speed reference, where 3000 represents the maximum frequency – often 50 or 60Hz. By using a multiply divide block there is no loss of accuracy as the internal result is 32 bit. As the numbers 8192 and 3000 are fixed, we simply use constant data inputs set to those values. The output connects to a data output set to 'User Speed Reference'



7.1.1. Parameter Settings

The parameter settings for this programme are straightforward.
As always, **P6 - 10** is set to 1 to enable the programme. To access this and other parameters, **P1 – 14** should be set to 201 (or to the appropriate password value if this has been changed).
In order to be able to change the settings in group 9 parameters (the internal connections for the different drive functions) it is necessary to set **P1 – 13** to 0.
The Group 9 parameters will have to be changed ready to accept new settings:

Group 0 | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 | Group 8 | Group 9

ID	Description	Value	Range	Default	Visible
P9-01	Enable Input Source	0: Safety Input		1: Digital Input 1	☑
P9-02	Fast Stop Input Source	0: OFF		0: OFF	☑
P9-03	Forward Run Input Source	0: OFF		1: Digital Input 1	☑
P9-04	Reverse Run Input Source	0: OFF		0: OFF	☑
P9-05	Latch Function Enable	0: OFF		0: OFF	☑
P9-06	Reverse Enable Source	0: OFF		2: Digital Input 2	☑
P9-07	Reset Input Source	0: OFF		1: Digital Input 1	☑
P9-08	External Trip Input Source	0: OFF		0: OFF	☑
P9-09	Terminal Control Override Source	16: ON		16: ON	☑
P9-10	Speed Source 1	0: Analog Input 1		0: Analog Input 1	☑
P9-11	Speed Source 2	0: Analog Input 1		2: Preset Speed	☑
P9-12	Speed Source 3	0: Analog Input 1		0: Analog Input 1	☑
P9-13	Speed Source 4	0: Analog Input 1		0: Analog Input 1	☑
P9-14	Speed Source 5	0: Analog Input 1		0: Analog Input 1	☑
P9-15	Speed Source 6	0: Analog Input 1		0: Analog Input 1	☑
P9-16	Speed Source 7	0: Analog Input 1		0: Analog Input 1	☑
P9-17	Speed Source 8	0: Analog Input 1		0: Analog Input 1	☑
P9-18	Speed Select Input 0	0: OFF		3: Digital Input 3	☑
P9-19	Speed Select Input 1	0: OFF		0: OFF	☑
P9-20	Speed Select Input 2	0: OFF		0: OFF	☑
P9-21	Preset Speed Select Input 0	0: OFF		5: Digital Input 5	☑

In particular, the digital input functions have been disconnected. For the drive to use these as before, these must be reconnected; that is, in this example they are set back to their defaults:

Group 0 | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 | Group 8 | Group 9

ID	Description	Value	Range	Default	Visible
P9-01	Enable Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-02	Fast Stop Input Source	0: OFF		0: OFF	☑
P9-03	Forward Run Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-04	Reverse Run Input Source	0: OFF		0: OFF	☑
P9-05	Latch Function Enable	0: OFF		0: OFF	☑
P9-06	Reverse Enable Source	2: Digital Input 2		2: Digital Input 2	☑
P9-07	Reset Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-08	External Trip Input Source	0: OFF		0: OFF	☑
P9-09	Terminal Control Override Source	16: ON		16: ON	☑
P9-10	Speed Source 1	0: Analog Input 1		0: Analog Input 1	☑
P9-11	Speed Source 2	0: Analog Input 1		2: Preset Speed	☑
P9-12	Speed Source 3	0: Analog Input 1		0: Analog Input 1	☑
P9-13	Speed Source 4	0: Analog Input 1		0: Analog Input 1	☑
P9-14	Speed Source 5	0: Analog Input 1		0: Analog Input 1	☑
P9-15	Speed Source 6	0: Analog Input 1		0: Analog Input 1	☑
P9-16	Speed Source 7	0: Analog Input 1		0: Analog Input 1	☑
P9-17	Speed Source 8	0: Analog Input 1		0: Analog Input 1	☑
P9-18	Speed Select Input 0	3: Digital Input 3		3: Digital Input 3	☑
P9-19	Speed Select Input 1	0: OFF		0: OFF	☑
P9-20	Speed Select Input 2	0: OFF		0: OFF	☑
P9-21	Preset Speed Select Input 0	0: OFF		5: Digital Input 5	☑
P9-22	Preset Speed Select Input 1	0: OFF		0: OFF	☑
P9-23	Preset Speed Select Input 2	0: OFF		0: OFF	☑

As mentioned earlier, notice that digital input 1 has three functions: enabling the drive, running it and resetting it (stopping), so all these must be reconnected.

Finally, Speed Source1 must be connected to the programme's 'User Speed Reference', so the speed is set by the result of the programme:

Group 0 | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 | Group 8 | Group 9

ID	Description	Value	Range	Default	Visible
P9-01	Enable Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-02	Fast Stop Input Source	0: OFF		0: OFF	☑
P9-03	Forward Run Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-04	Reverse Run Input Source	0: OFF		0: OFF	☑
P9-05	Latch Function Enable	0: OFF		0: OFF	☑
P9-06	Reverse Enable Source	2: Digital Input 2		2: Digital Input 2	☑
P9-07	Reset Input Source	1: Digital Input 1		1: Digital Input 1	☑
P9-08	External Trip Input Source	0: OFF		0: OFF	☑
P9-09	Terminal Control Override Source	16: ON		16: ON	☑
P9-10	Speed Source 1	7: User Speed		0: Analog Input 1	☑
P9-11	Speed Source 2			2: Preset Speed	☑
P9-12	Speed Source 3			0: Analog Input 1	☑
P9-13	Speed Source 4			0: Analog Input 1	☑
P9-14	Speed Source 5			0: Analog Input 1	☑
P9-15	Speed Source 6			0: Analog Input 1	☑
P9-16	Speed Source 7			0: Analog Input 1	☑
P9-17	Speed Source 8			0: Analog Input 1	☑
P9-18	Speed Select Input 0			3: Digital Input 3	☑
P9-19	Speed Select Input 1			0: OFF	☑
P9-20	Speed Select Input 2			0: OFF	☑
P9-21	Preset Speed Select Input 0			5: Digital Input 5	☑
P9-22	Preset Speed Select Input 1			0: OFF	☑

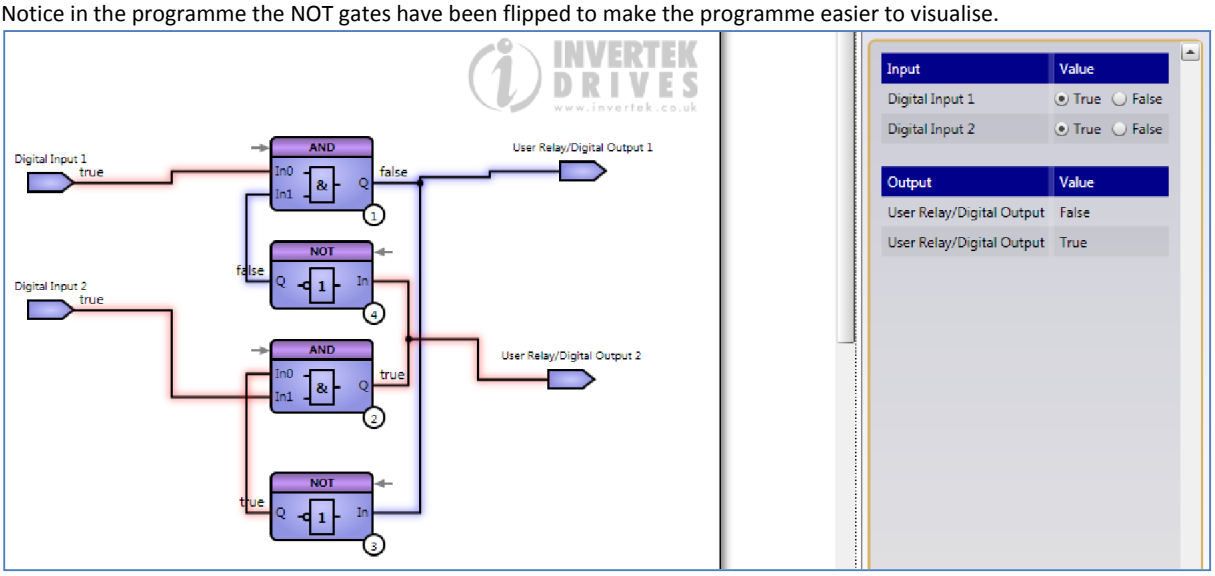
0: Analog Input 1
1: Analog Input 2
2: Preset Speed
3: Keypad Speed
4: User PID Speed
5: Master Speed
6: Fieldbus Speed
7: User Speed
8: Frequency Input
9: Preset Speed 1
10: Preset Speed 2
11: Preset Speed 3
12: Preset Speed 4
13: Preset Speed 5

So we've selected 'User Speed' from the drop down menu.

You can see from the simulation above that the scaling suits the speed reference as required.

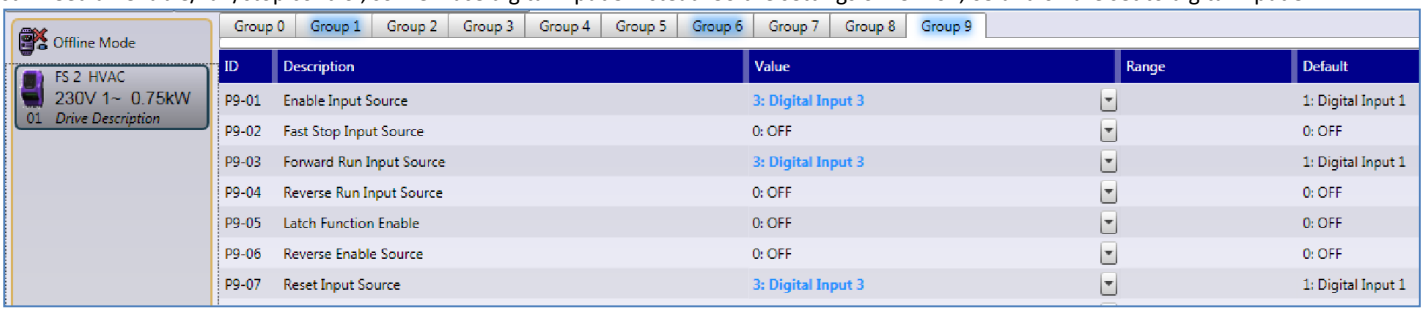
7.2. Priority Control of Relays

This is an example of a programme which has little effect on the drive, and is possibly 'borrowing' the spare inputs and outputs for another purpose, thus saving the cost of additional control equipment. In this case, the logic gates act as a latch, such that once one input goes high, the corresponding relay is energised and the other input and relay are locked out.
In the simulation shown below, Digital input 2 was TRUE, so Relay 2 is also TRUE. Later, Digital input 1 has gone TRUE, but with no effect as it has been 'locked' by the action of Digital input 2 via AND gate 2 and NOT gate 4.
Notice in the programme the NOT gates have been flipped to make the programme easier to visualise.

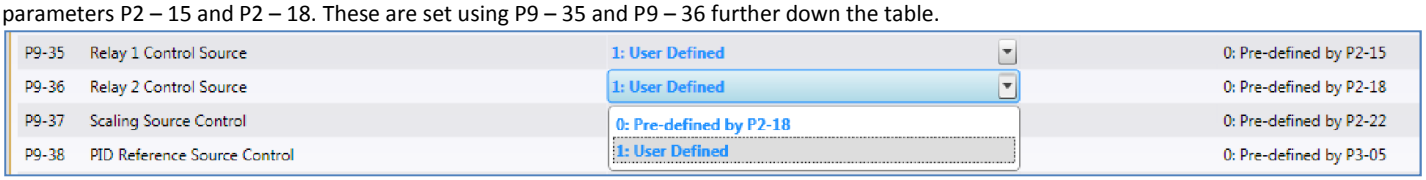


7.2.1. *Parameter Settings*

Once again, it is necessary to set P6 – 10 to 1 to enable the programme, and P1 -13 to 0 to release the settings of the group 9 parameters. This drive is an HVAC drive, so the default settings are a little different for digital inputs 2 and 3. The settings for the digital inputs have been disconnected ready for reallocation, so as we are using digital inputs 1 and 2 we will not reconnect them to any internal setting. However, we still need an enable/run/stop control, so we'll use digital input 3 instead. So the settings of P9 – 01, 03 and 07 are set to digital input 3.

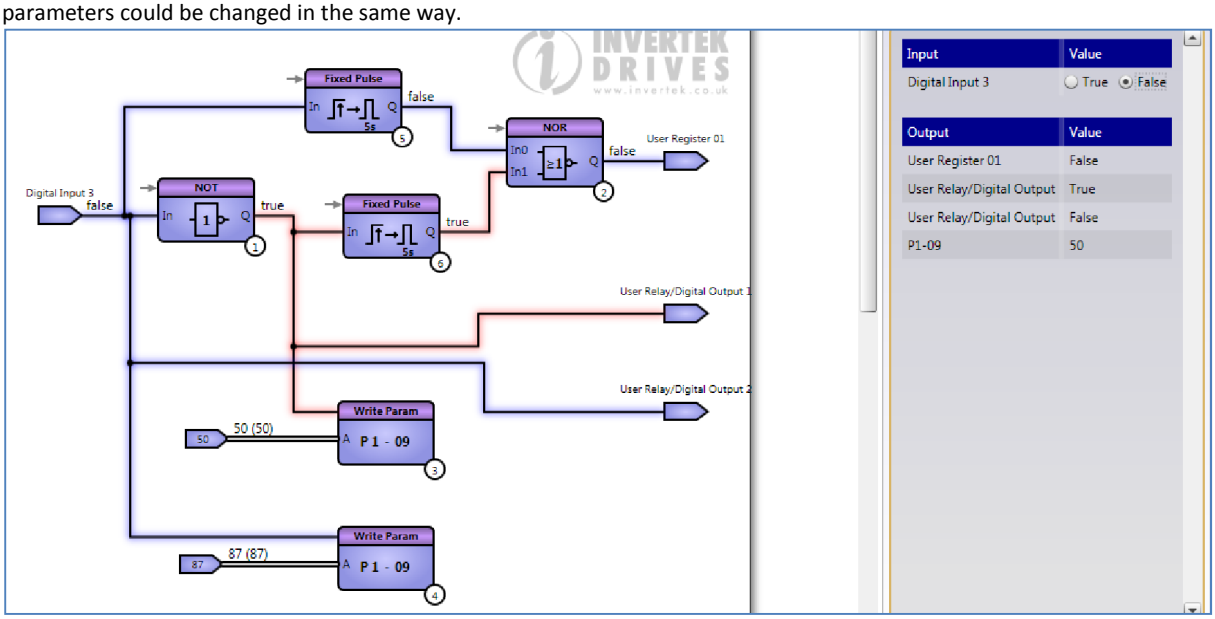


We are using the two relays, so we need to set these to User defined, instead of their default settings which links them to the settings of parameters P2 – 15 and P2 – 18. These are set using P9 – 35 and P9 – 36 further down the table.



7.3. 50 to 87Hz Switchover

It is sometimes advantageous to use a motor connected for 230V operation on a 400V supply (assuming the motor is designed with 400V insulation) and increase the maximum frequency to 87Hz. This extends the Voltage to frequency curve to the higher speed, retaining full torque. Therefore more power may be extracted from the motor.
In this design, digital input 3, when FALSE, enables relay 1 via the NOT gate, and enables the Write parameter block to write 50 (Hz) to parameter P1 – 09. When TRUE, relay 2 is energised, and another Write parameter block is used to set P1 – 09 to 87.
The digital input also generates a 5 second pulse when it changes state (i.e. changing from 50 to 87 and also when changing from 87 to 50), using two fixed pulse blocks and a NOR gate to combine and invert them. The purpose here is to provide a short time when the drive is inhibited during the changeover. The output is connected to User register 1 for this purpose.
In the example below, digital input 3 has just gone from TRUE to FALSE, so the output of the NOR gate is FALSE (inhibiting the drive - see below), and 50Hz is selected for P1 – 09 via the write parameter block. If the drive were changing between two different motors, other parameters could be changed in the same way.



7.3.1. *Parameter Settings*

P6 – 10 to 1, P1 – 13 to 0 as usual. In group 9, we have re-enabled many of the functions, but we have connected the Forward Run function P9 – 03, and the reset function P9 - 07 to the output of User Register 1, so the drive will be stopped by the action of the 5 second pulses as described above. Note that digital input 1 still provides a 'master' enable signal at P9 – 01(and must therefore be connected to something), and that digital input 3 remains unconnected at P9 – 18 so it can be used by the programme.

Group 0 | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 | Group 8 | Group 9

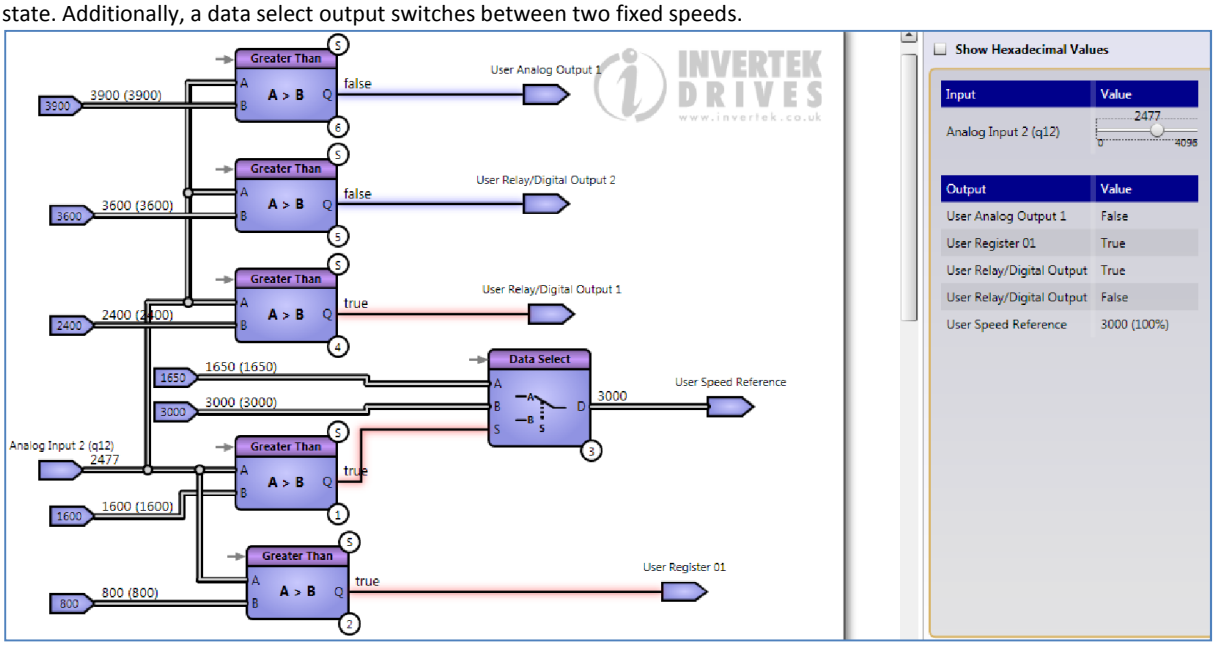
ID	Description	Value	Range	Default
P9-01	Enable Input Source	1: Digital Input 1		1: Digital Input 1
P9-02	Fast Stop Input Source	0: OFF		0: OFF
P9-03	Forward Run Input Source	17: User Register 1		1: Digital Input 1
P9-04	Reverse Run Input Source	0: OFF		0: OFF
P9-05	Latch Function Enable	0: OFF		0: OFF
P9-06	Reverse Enable Source	0: OFF		0: OFF
P9-07	Reset Input Source	17: User Register 1		1: Digital Input 1
P9-08	External Trip Input Source	0: OFF		0: OFF
P9-09	Terminal Control Override Source	16: ON		16: ON
P9-10	Speed Source 1	0: Analog Input 1		0: Analog Input 1
P9-11	Speed Source 2	1: Analog Input 2		1: Analog Input 2
P9-12	Speed Source 3	2: Preset Speed		2: Preset Speed
P9-13	Speed Source 4	2: Preset Speed		2: Preset Speed
P9-14	Speed Source 5	0: Analog Input 1		0: Analog Input 1
P9-15	Speed Source 6	0: Analog Input 1		0: Analog Input 1
P9-16	Speed Source 7	0: Analog Input 1		0: Analog Input 1
P9-17	Speed Source 8	0: Analog Input 1		0: Analog Input 1
P9-18	Speed Select Input 0	0: OFF		3: Digital Input 3

Finally, the relays are set for user defined as in the previous example.

P9-35	Relay 1 Control Source	1: User Defined	0: Pre-defined by P2-15
P9-36	Relay 2 Control Source	1: User Defined	0: Pre-defined by P2-18
P9-37	Scaling Source Control	0: Pre-defined by P2-18	0: Pre-defined by P2-22
P9-38	PID Reference Source Control	1: User Defined	0: Pre-defined by P3-05

7.4. Multistage Comparators

In this example, several greater than comparators are connected to Analog input 2. As the value of the input changes, various outputs change state. Additionally, a data select output switches between two fixed speeds.



So as analog input 2 increases from 0, user register 01 changes state at 800 (this starts the drive – see below), the data select switches at 1600, user relay 1 is energised at 2400, User relay 2 at 3600, and Analog output 1 at 3900. In the simulation above, analog input 2 has the value 2477.

7.4.1. Parameter Settings

As always, P6 – 10 to 1, P1 – 13 to 0.

In group 9, we'll connect the run (P9-03) and reset (P9-07) functions to User register 1, so the first comparator will start the drive. We'll connect the User speed to speed source 1 (P9-10) as shown below.

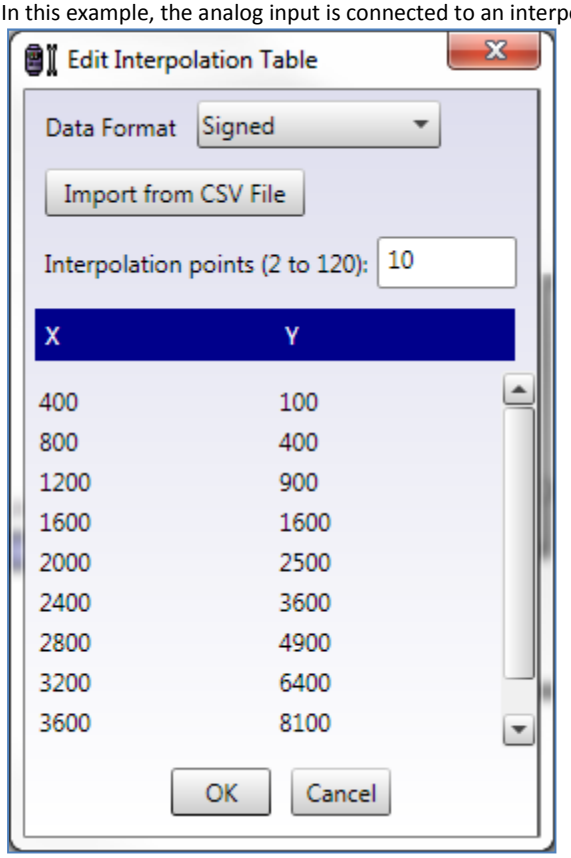
P9-01	Enable Input Source	1: Digital Input 1		1: Digital Input 1
P9-02	Fast Stop Input Source	0: OFF		0: OFF
P9-03	Forward Run Input Source	17: User Register 1		1: Digital Input 1
P9-04	Reverse Run Input Source	0: OFF		0: OFF
P9-05	Latch Function Enable	0: OFF		0: OFF
P9-06	Reverse Enable Source	0: OFF		0: OFF
P9-07	Reset Input Source	17: User Register 1		1: Digital Input 1
P9-08	External Trip Input Source	0: OFF		0: OFF
P9-09	Terminal Control Override Source	16: ON		16: ON
P9-10	Speed Source 1	7: User Speed		0: Analog Input 1
P9-11	Speed Source 2	1: Analog Input 2		1: Analog Input 2
P9-12	Speed Source 3	2: Preset Speed		2: Preset Speed

Digital input 1 (P9-01) is still an overall enable, and must therefore be linked or connected to a master enable contact. Finally, we'll change P9-33 (analog output 1 source) P9-35 and P9-36 Relay controls to User connections, as shown below.

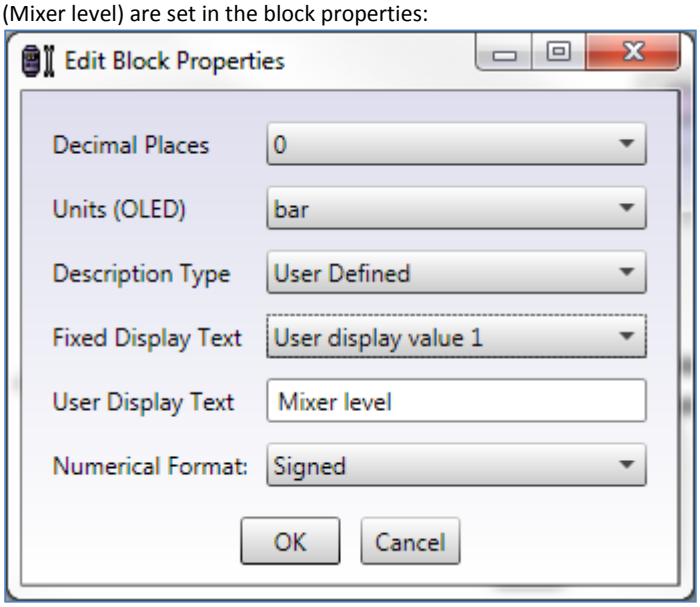
P9-33	Analog Output 1 Control Source	1: User Analog Output 1 (Digital)		0: Pre-defined by P2-11
P9-34	Analog Output 2 Control Source	0: Pre-defined by P2-13		0: Pre-defined by P2-13
P9-35	Relay 1 Control Source	1: User Defined		0: Pre-defined by P2-15
P9-36	Relay 2 Control Source	1: User Defined		0: Pre-defined by P2-18

7.5. 5 Interpolator and Display

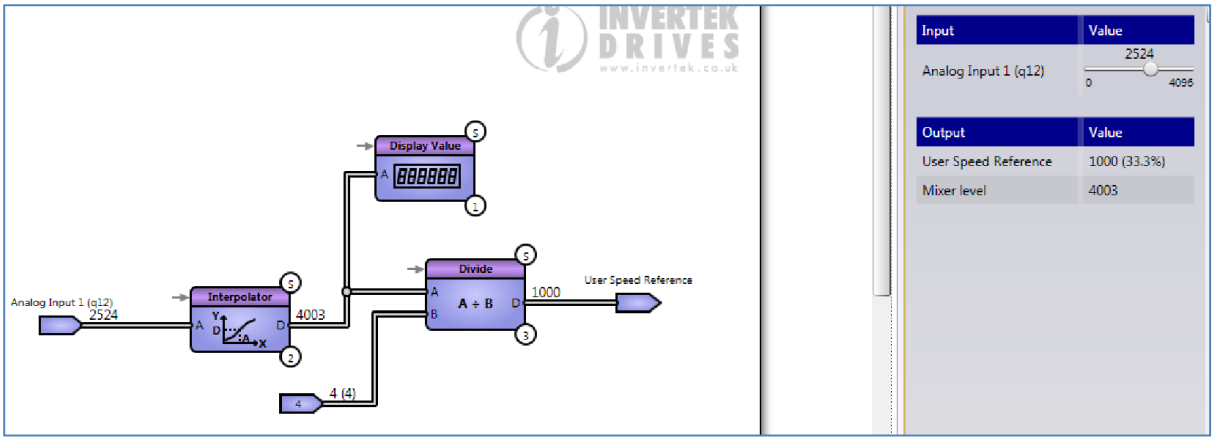
In this example, the analog input is connected to an interpolator function that changes the value according to the interpolator look up table:



The result is used as a speed reference (scaled simply by dividing by 4) and is also connected to the display. The units (Bar) and display text (Mixer level) are set in the block properties:



This information will only appear if a drive with a built in OLED display is used.
The simulation shows that the interpolating table is step less in operation, unlike a look up table.



7.5.1. Parameter Settings

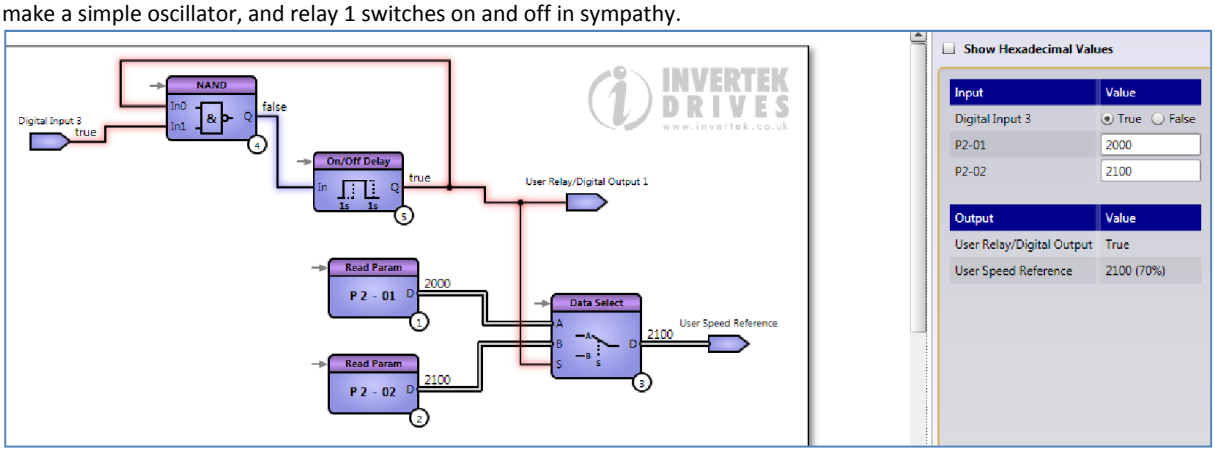
As always, P6 – 10 to 1, P1 – 13 to 0.

In group 9, it is only necessary to connect the User speed to speed source 1 (P9-10) as in previous examples.

7.6. Wobbulation

In some applications such as textile winding, it is useful to be able to slightly vary the speed of the winder to prevent pattern build up on the bobbin.

This programme uses digital input 3 to switch between two fixed speeds in parameters P2-01 and P2-02. The NAND gate and the on/off delay make a simple oscillator, and relay 1 switches on and off in sympathy.



The fixed speeds are selected in the simulation as 2000 (that is 66% of 3000, or 33Hz with a 50Hz maximum) and 2100 (35Hz). The oscillator will run at 1Hz.

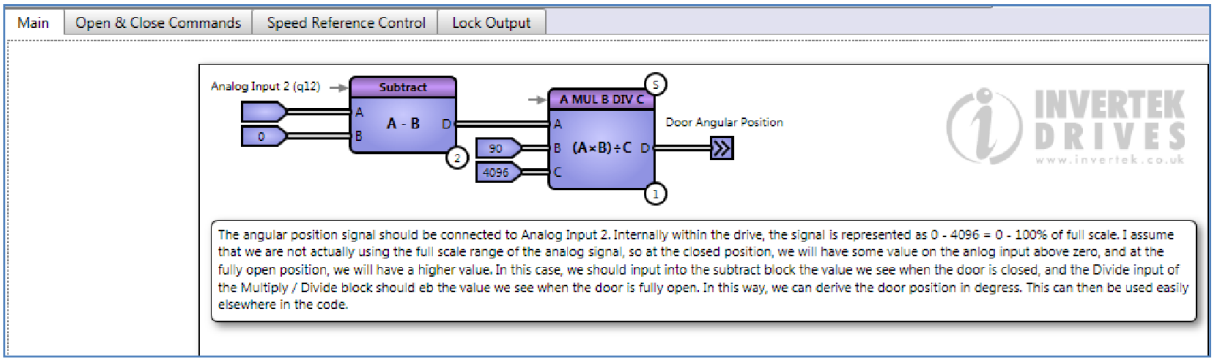
7.6.1. Parameter Settings

As always, P6 – 10 to 1, P1 – 13 to 0.

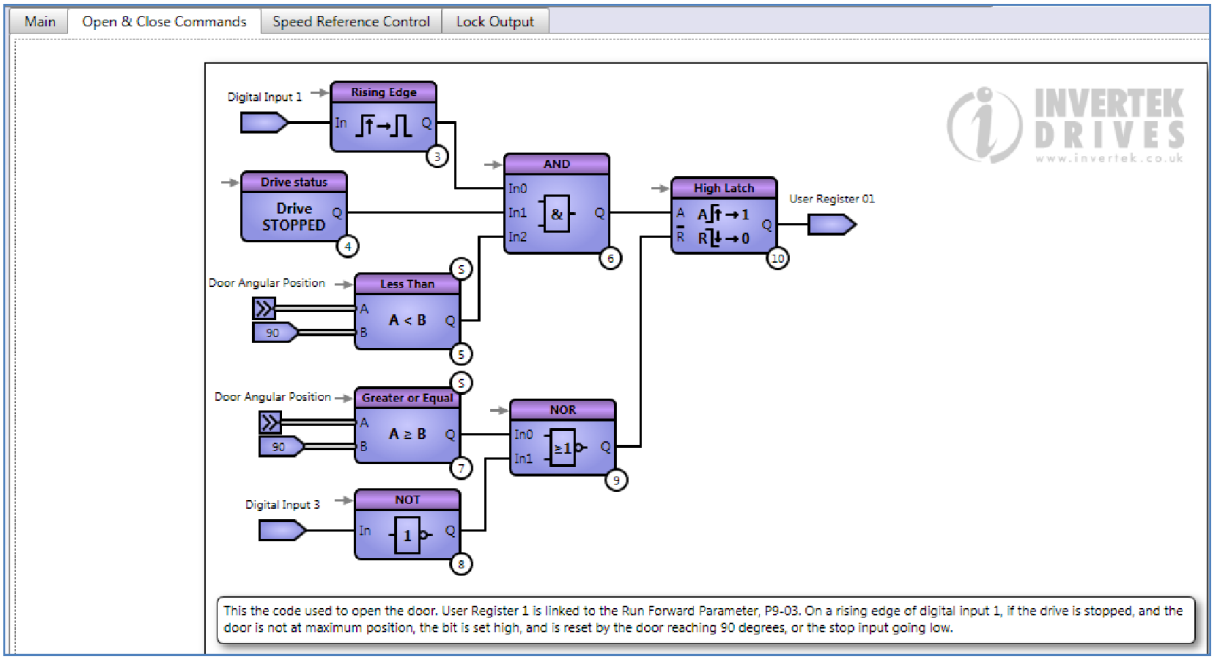
In group 9, we'll reconnect digital inputs 1 and 2 to their defaults, digital input 3 now being used to enable the oscillation. P9-10 will be set to User speed again, and P9-35 Relay 1 Control Source set to User defined to allow the programme to control the relay.

7.7. Door Position Control

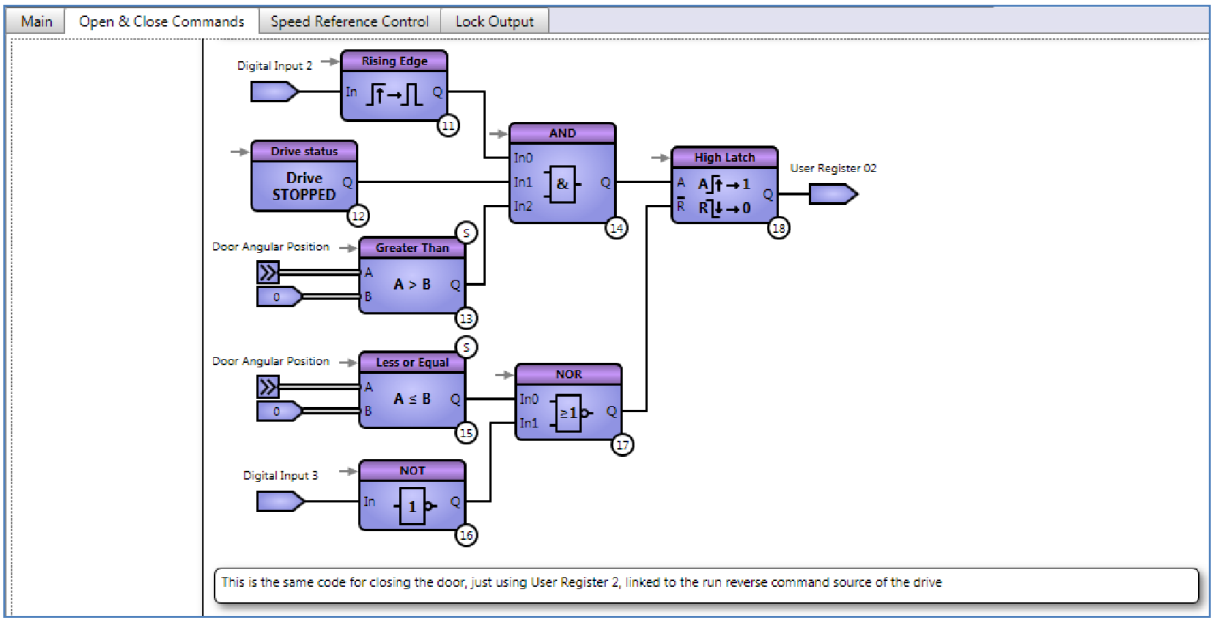
This last example is a real example of an application where a drive was used to open and close a door, based on an angular position sensor and commands from Digital inputs 1 and 2. The programme is well documented with comments, but is still difficult to follow as even a simple requirement soon grows when all possible conditions are taken into account. There are four pages of programme, shown below.



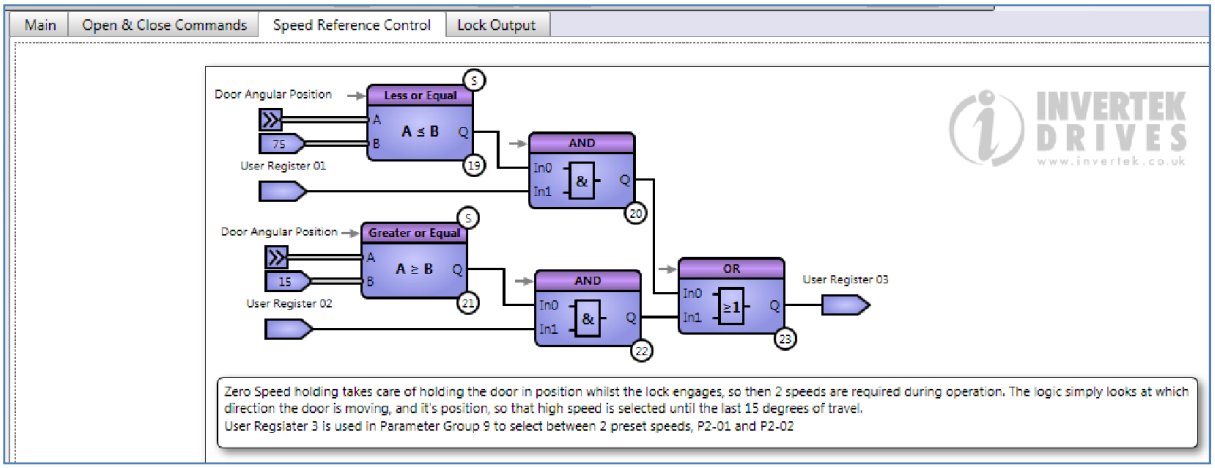
Page 1 (Main) shows the scaling and trimming functions for the position sensor.
Page 2 (Open and Close Commands) shows two similar blocks of logic to control the open and close functions. First the 'Open' logic:



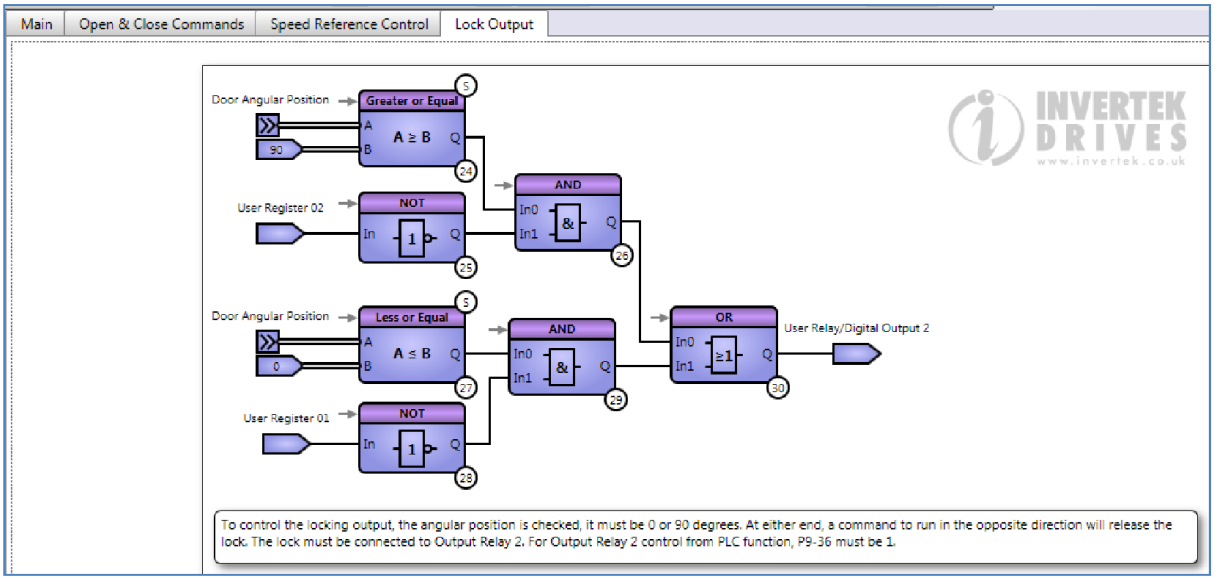
Now the 'Close' logic:



Page 3 (Speed Reference Control) selects a speed for closing, and allows a holding at zero speed for locking.



Page 4 (Lock Output) locks and releases the door at appropriate times.



7.7.1. Parameter Settings

In this application, Digital input 3 is used as the start and stop functions, so P9-01 and 7 are connected.
User Registers 1 and 2 act as run forward and run reverse controls, so they are connected to P9-03 and P9-04 respectively.
User register 3 is used to select the various speed values, so P9-10, speed source 1 is connected to Preset speed, and User register 3 controls P9-21 the first speed select parameter.
In other words, User register 3 sets P9-21 to a high or low, which selects between P2-01 (5Hz) and P2-02 (50Hz).
P9-36 must be set to user defined to allow the relay to be controlled by the programme.
P6-10 must be enabled, and P1-13 set to 0 for all this to function.

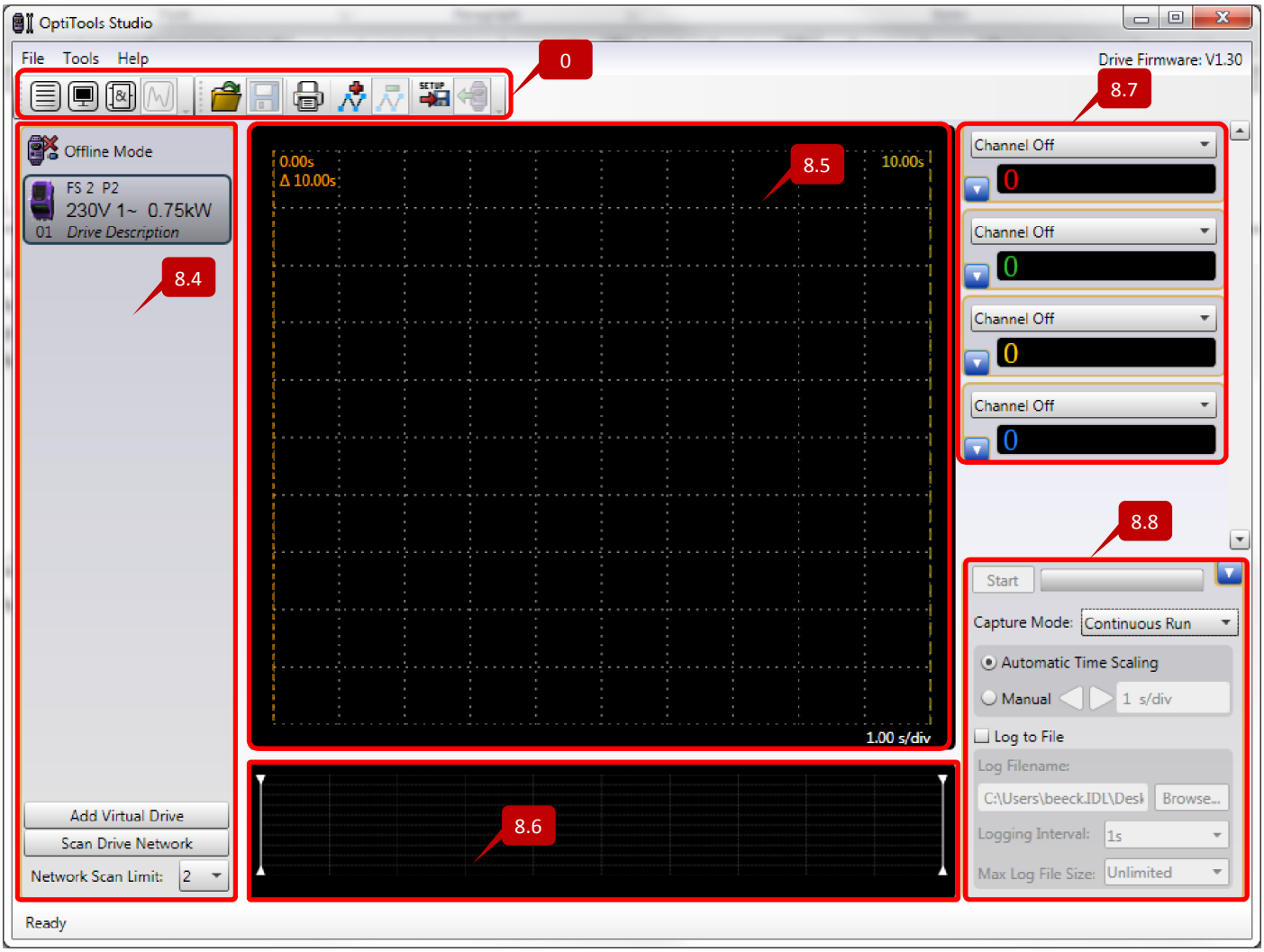
8. Scope / Datalogger

8.1. Overview

The Scope / Data Logger allows real time display and logging of signal levels from within connected drives. It is primarily intended as a commissioning / fault finding tool to enable the user to visualise the operation of the drive.
There are two basic modes of operation:-

- Continuous Run
  - Data is exchanged between the drive and PC in real time, and displayed directly on the screen
  - Minimum Sample time is 1 second
- Trigger On Event
  - Data is stored in the drives internal memory, and read out by the PC software
  - 500 Sample points per channel are stored
  - Minimum sample time is 1ms

8.2. Scope / Data Logger Display



8.3. Toolbar

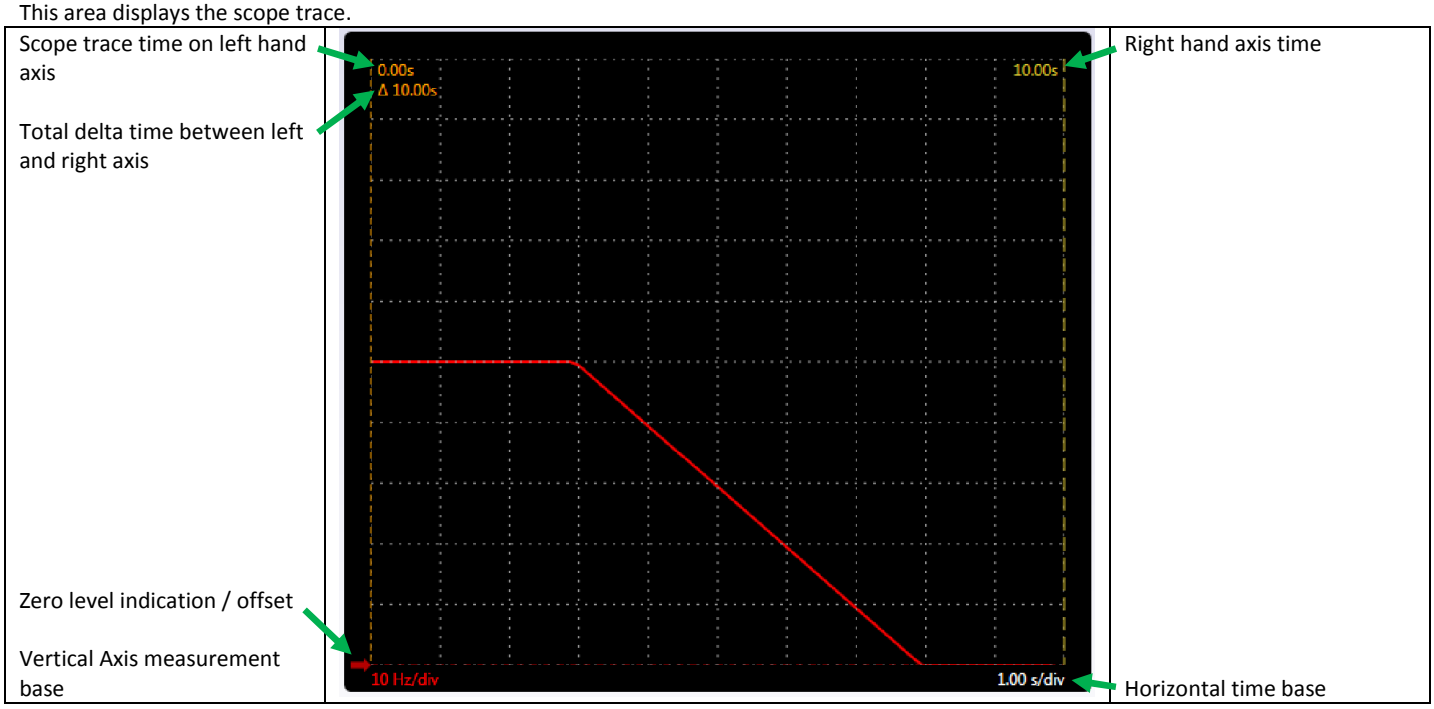
Button	Function
	Selects the Parameter Editor
	Select the Drive Monitor
	Selects the Function Block Editor
	Selects the Scope / Datalogger
	Load a scope trace from a file
	Save a scope trace file
	Print the scope trace
	Show scope data sample points
	Hide scope data sample points
	Save current scope configuration to a file
	Download scope data from the drive memory

8.4. Drive Network

The drive network window operates in the same manner as for the parameter editor tool described in sections 3.5 to 3.10

8.5. Main Display

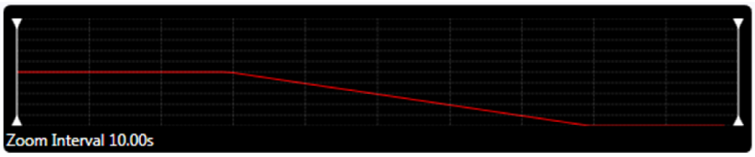
This area displays the scope trace.



The scope traces are colour coded to their respective channels.
Offset and vertical scaling may be manually adjusted within the scope channel.
The vertical dashed lines on the left and right of the display may be dragged across with the mouse. The instantaneous values of the channels are then shown in the channel data area.

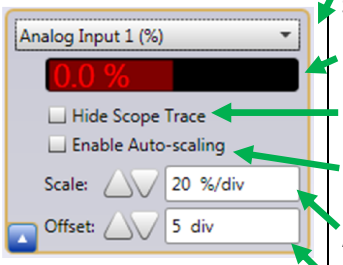
8.6. Sub Display

The sub display allows the user to select an area of a trace which will be displayed in the main display above. By moving the white vertical lines at either end, the zoom area can be selected.



8.7. Channel Selection

The scope function supports a maximum of four channels. Each channel can be individually configured as shown below.



Selects the source from the channel. The available sources depend on the drive type in use.

Displays the real time value of the channel when online, or the selected trace point value when offline

Hides the trace

Enables automatic scaling

Allows adjustment of manual scaling

Allows adjustment of the vertical offset of the trace.

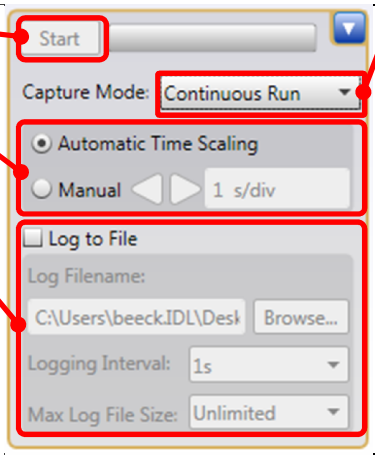
8.8. Operating Mode

Continuous Run Mode

Start / Stop Real Time Scope Function

Select Automatic or Manual Time Scaling
In manual mode, the time base cane be selected

Enable Real Time Logging Directly to a file
Select the file name and location
Define the logging interval
Limit the Size of the file



Selects the scope Mode

Continuous Run – Data is logged in real time from the drive, and displayed on the screen

Trigger on Event Mode

Start / Stop the internal data logger of the drive

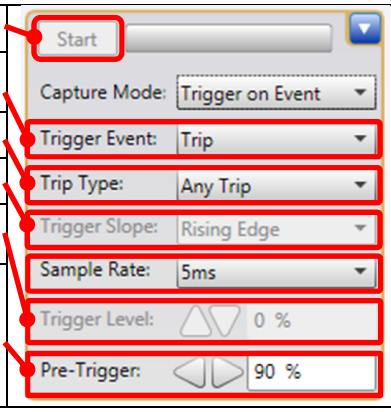
Select the Trigger Event – See below for further details

Selects the trigger type

Set the sample rate

Sets the trigger level when an analog signal is used

Select the pre / post trigger percentage



Trigger on Event – Data is logged internally in the drive at high speed, then loaded to the PC afterwards

8.8.1. Trigger Event Selection

When the internal data logger is used with compatible drives, a trigger source is required to start the logging process. This trigger can be one of the following:-

- Trip
  - Any Trip may be used, or a specific trip can be selected in the Trip Type dialogue box
- Level on Channel 1
  - The Level can be selected using the Trigger Level
- Digital Inputs 1 – 8

The pre trigger adjustment allows the user to select how much of the data memory is used to log data prior to the event, e.g. if pre trigger is 90%, 90% of the data logging memory is used to log data before the trigger, and 10% is used to log data after the trigger. This can be useful for events such as trips, where the important information is what happened prior to the trip occurring.

8.9. Scope Function Support

The following functions are supported by respective drive types.

Drive Type	Continuous Run	Trigger On Event
Optidrive P2 (V1.30 F/W or later)	Yes	Yes
Optidrive E2	Yes	No
Optidrive E2 Single Phase Output	Yes	No
Optidrive HVAC (V1.30 F/W or later)	Yes	Yes
Optidrive HVAC Eco (V1.30 F/W or later)	Yes	Yes
Optidrive Elevator (V1.30 F/W or later)	Yes	Yes

9. Function Block Editor - Inputs and Outputs

This section lists the available inputs and outputs from the Function Block Editor I/O list section. Some of these, such as the digital inputs are clear in their function, others are explained more fully below. Some settings are registers that have been included for completeness. Information concerning individual settings can be obtained by selecting the setting and hovering the mouse over the input or output block.

9.1. Logic inputs

Logic inputs have a binary state only – they may be TRUE or FALSE

Name	Function
Digital Input 1	Indicates the status of the respective digital input. Digital Inputs 1 – 5 are the Optidrive's on-board digital inputs – refer to the relevant drive User Guide for further information
...	
Digital Input 5	
Digital Input 6	Indicates the status of the relevant digital input. Digital Inputs 6 – 8 are the digital inputs on the extended I/O option module. If this module is not fitted to the drive, the status of these inputs will always remain FALSE.
Digital Input 7	
Digital Input 8	
Digital Output 1	
Digital Output 2	
Digital Output 3	
Digital Output 4	
Digital Output 5	
Off condition	Always FALSE
On Condition	Always TRUE
User Register 1	
...	
User Register 31	
User Relay / Digital Output 1	
User Relay / Digital Output 2	
User Relay / Digital Output 3	
User Relay / Digital Output 4	
User Relay / Digital Output 5	

9.2. Logic Outputs

Logic Outputs have a binary state only – they may be TRUE or FALSE

Name	Function
User Analog Output 1	
User Analog Output 2	
User Register 1	
...	
User Register 31	
User Relay / Digital Output 1	
User Relay / Digital Output 2	
User Relay / Digital Output 3	
User Relay / Digital Output 4	
User Relay / Digital Output 5	

9.3. Data Inputs

Data inputs contain analog signal levels. The table below lists the available inputs, and where required, the internal units used and expected range of the signal.

Name	Units	Scaling / Range	Function	Supported			Par
				P2	HVAC	Eco	
24 Hour Time Clock (minutes)	Minutes	0 - 1439	Continuous 24 hour time minutes value	X	X	X	-
Ambient Temperature	°C	-	Measured Control board temperature	X	X	X	P0-72
Analog Input 1 (%)	%	0 – 1000 = 0 – 100.0%	Analog input 1 signal level expressed as %	X	X	X	P0-01
Analog Input 1 (q12)	12bit + Sign	0 – 4096 = 0 – 100.0%	Analog Input 1 signal level, 12bit + sign	X	X	X	
Analog Input 2 (%)	%	0 – 1000 = 0 – 100.0%	Analog input 2 signal level expressed as %	X	X	X	P0-02
Analog Input 2 (q12)	12bit + Sign	0 – 4096 = 0 – 100.0%	Analog Input 2 signal level, 12bit + sign	X	X	X	
Analog Output 1	12bit + Sign	0 – 4096 = 0 – 100.0%	Analog Output 1 Signal level, 12 bit format	X	X	X	
Analog Output 2	12bit + Sign	0 – 4096 = 0 – 100.0%	Analog Output 2 Signal level, 12 bit format	X	X	X	
DC Bus Voltage	Volts		Measured DC Bus Voltage	X	X	X	
Digital Input Status (1-5)	-	-	Digital Input Status represented as follows Bit 0 = Digital Input 1 Status Bit 1 = Digital Input 2 Status Bit 2 = Digital Input 3 Status Bit 3 = Digital Input 4 Status Bit 4 = Digital Input 5 Status	X	X	X	P0-03
Digital Speed Pot	-	3000 = 50.0Hz	Value of the motorised pot	X	X	X	P0-06
Drive Power Units	-	-	Indicates if the drive power units are kW or HP 0 = kW 1 = HP	X	X		
Drive Temperature	°C		Measured heatsink temperature	X	X	X	P0-21
Drive Type Code	-	-	Internal type code of the drive	X	X	X	
Extended I/O Input	-		Digital Input Status of the Extended I/O option module represented as follows Bit 0 = Digital Input 6 Status Bit 1 = Digital Input 7 Status Bit 2 = Digital Input 8 Status	X	X	X	
External Keypad Button Status			Shows the status of the buttons on a remote keypad (Optiport or Optipad) as follows:- Bit 0 = No function Bit 1 = Down Button Bit 2 = Up Button	X	X	X	

Name	Units	Scaling / Range	Function	Supported			Par
			Bit 3 = Navigate Button Bit 4 = Stop Button Bit 5 = Start Button Bit 6 = Hand Button Bit 7 = Auto Button				
Fieldbus Speed Reference	-	3000 = 50.0Hz	Speed reference received from Fieldbus interface	X	X	X	P0-07
Fieldbus Torque Reference	-	0 – 4096 = 0- 200.0%	Torque reference received from fieldbus interface	X			P0-67
Frequency Speed Reference	-	3000 = 50.0Hz	Speed reference derived from pulse frequency input	X			
Fieldbus Module Type ID	-	-	Displays the type of Fieldbus interface fitted as follows 0000h = No modules fitted 0005h = Profibus DP 0025h = DeviceNet 0065h = Control Net 0080h = Modbus TCP Single Port 0084h = Profinet Single Port 0085h = Ethernet IP Single Port 0087h = EtherCat 0090h = CC Link 0093h = Modbus TCP Dual Port 0096h = Profinet Dual Port 009Ch = Ethernet IP Dual Port 0098h = Sercos III 0099h = BACNet MS/TP 009Ah = BACNet IP	X	X	X	P0-71
kWh meter (non resettable)	kWh	0 – 999	Energy consumption meter kWh	X	X	X	P0-26
kWh meter (User resettable)	kWh	0 – 999	Energy consumption meter kWh	X	X	X	P0-26
Language Index			Selected language of the OLED display (where fitted), indicated as follows 0 = English 1 = German 2 = Spanish 3 = Italian 4 = French 5 = Swedish 6 = Russian	X	X	X	
Master Speed Reference	-	3000 = 50.0Hz		X	X	X	
Master Torque Reference	-	0 – 4096 = 0- 200.0%		X			
Modbus RX Result 1 ... Modbus RX Result 8	-	-	Value read from a connected Modbus slave using the Modbus Read function	X	X	X	-
Modbus RX Status 1 ... Modbus RX Status 8	-	-	Status of the relevant Modbus Read Function	X	X	X	-
Modbus TX Status 1 ... Modbus TX Status 8			Status of the relevant Modbus Write Function	X	X	X	-
Motor Current	A	10 = 1.0A	Output current with 1 decimal place resolution	X	X	X	
Motor Power	kW	100 = 1.00kW	Output power with 2 decimal places resolution	X	X	X	
Motor Speed	-	3000 = 50.0Hz	Motor speed with internal scaling	X	X	X	
Motor Torque	-	0 – 4096 = 0- 200.0%		X			P0-12
Mwh meter (non-resettable)	MWh	-	Energy consumption meter MWh	X	X	X	P0-27
MWh meter (User resettable)	MWh	-	Energy consumption meter MWh	X	X	X	P0-27
OLED Display Firmware Version	-	-	Firmware version of the OLED display (where fitted)	X	X	X	
PLC Scan Period	ms	-	Cycle time of the last execution of the PLC program in ms	X	X	X	
PLC User ID	-	-	User ID of the PLC program	X	X	X	
Option Module ID	-	-	ID of the Option Module (where fitted) as follows 0 = Reserved 1 = Incremental Encoder 2 = Reserved 3 = Extended I/O / Cascade 4 = Fieldbus 5 = Reserved 6 = Reserved 7 = No Module fitted	 X X X X	 X X X X	 X X X	P0-70
Reserved 1	-	-	No function				
Reserved 2	-	-	No function				
Run Time Since Last Enable (hrs.)	Hours	-		X	X	X	P0-34
Run Time since Last Enable (Seconds)	Seconds	0 - 3559		X	X	X	P0-34
Scaled Display Value 1	-	-	Scaled display value by P2-21	X	X	X	
Scaled Display Value 2	-	-	Scaled PID display value by P3-12		X	X	
Scope Channel 1 Data	-	-	Value of the selected Scope Channel 1	X	X	X	
Scope Channel 2 Data			Value of the selected Scope Channel 2	X	X	X	
Scope Index 1 / 2			Indicates which values are selected in the Scope channels	X	X	X	
Slave Speed Reference	-		Speed reference from Master	X	X	X	
Speed Reference	-	3000 = 50.0Hz	Speed reference value	X	X	X	
Time to Next Service	Hours	-	Remainder of the Service Interval Time	X	X	X	
Torque Reference (%)	%	1000 = 100.0%	Torque setpoint (%)	X			P0-05
Torque Reference (q12)	-	4096 = 200.0%	Torque setpoint (12 bit)	X			
Total Run Time (hrs.)	Hours	-	Drive Lifetime (Hours)	X	X	X	P0-65
Total Run Time (seconds)	Seconds	0 - 3559	Drive Lifetime (Seconds)	X	X	X	P0-65
User Analog Output 1	-	0 – 4096	Analog Output 1 Signal Level	X	X	X	
User Analog Output 2	-	0 - 4096	Analog Output 2 Signal Level	X	X	X	

Name	Units	Scaling / Range	Function	Supported			Par
User PID Control Feedback Input							
User PID Control Output (q12)							
User PID Control Output (Speed)							
User PID Control Ref Input							P0-08
User PID Controller Feedback							P0-09
User PID Controller Reference							
User Register 1	-	-	Value of the relevant User Register	X	X	X	
...							
User Register 31							
User Scaled Value Decimal Place							
User Scaling Value Source							
User Speed Reference	-	3000 = 50.0Hz	Value of the User Speed Reference	X	X	X	
User Torque Reference	-	0 – 4096 = 0- 200.0%	Value of the User Torque Reference	X			
User Fieldbus Ramp Time	Seconds	10 = 1.0s	Ramp times from Fieldbus Interface	X	X	X	

9.4. Data Outputs

Name	Units	Scaling / Range	Function	Supported			Par
				P2	HVAC	Eco	
24 Hour Time Clock (minutes)	Minutes	0 - 1439	Continuous 24 hour time minutes value	X	X	X	-
Scope Index 1 / 2			Indicates which values are selected in the Scope channels	X	X	X	
User Analog Output 1	-	0 – 4096	Analog Output 1 Signal Level	X	X	X	
User Analog Output 2	-	0 - 4096	Analog Output 2 Signal Level	X	X	X	
User PID Controller Feedback							P0-09
User PID Controller Reference							
User Register 1	-	-	Value of the relevant User Register	X	X	X	
...							
User Register 31							
User Scaled Value Decimal Place							
User Scaling Value Source							
User Speed Reference	-	3000 = 50.0Hz	Value of the User Speed Reference	X	X	X	
User Torque Reference	-	0 – 4096 = 0- 200.0%	Value of the User Torque Reference	X			
User Fieldbus Ramp Time	Seconds	10 = 1.0s	Ramp times from Fieldbus Interface	X	X	X	

OptiTools Studio

SCATTERGOOD & JOHNSON LTD
ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS
Est.1899

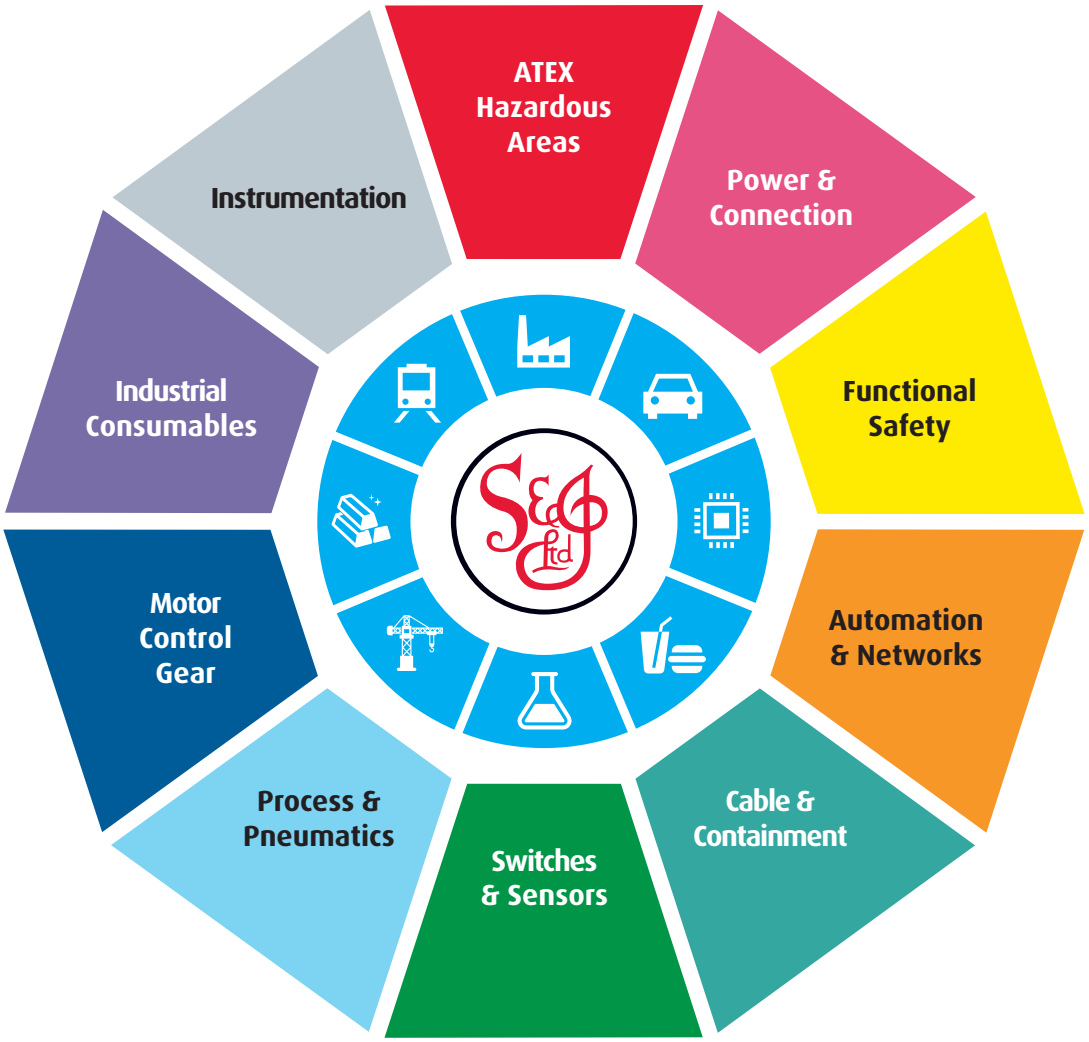
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

www.scatts.co.uk