TRAINING

# $i^3$ ...Display, Control, Communicate


# Tutorial

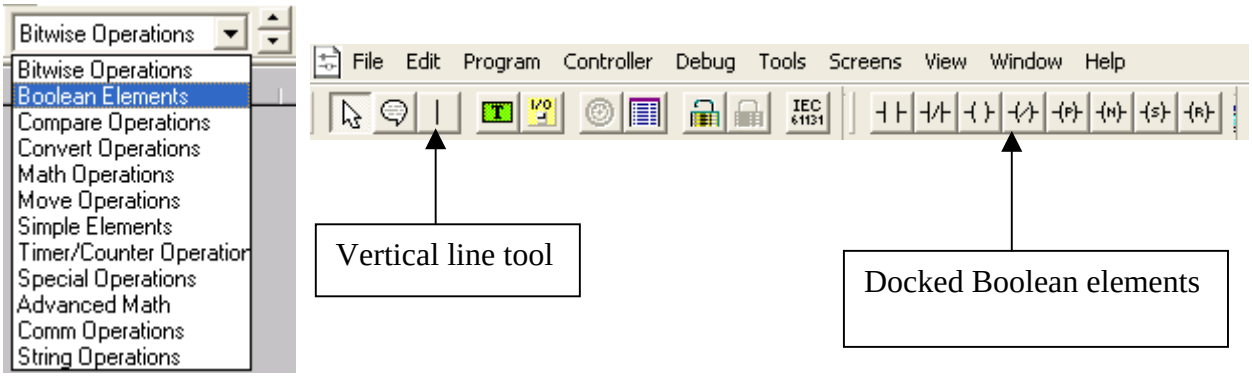
# Basic Operation

IMO

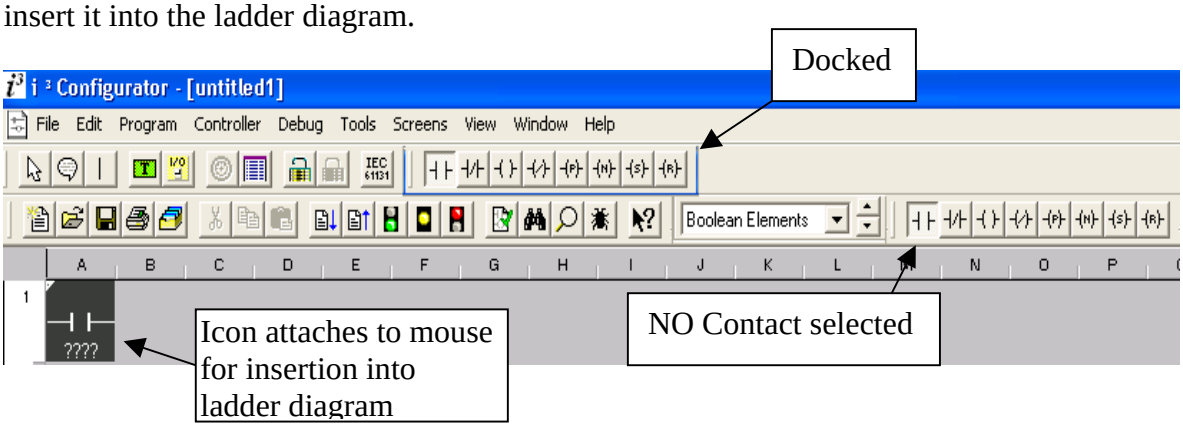
# Table of Contents

# Basic Ladder Elements

All the Functions, contacts and coils for the ladder programming are contained in a drop down menu to select what range of functions you want to insert. It is also possible to scroll through the sub-sections using the up and down arrow keys. These sub sections can also be docked on to the main programming interface.

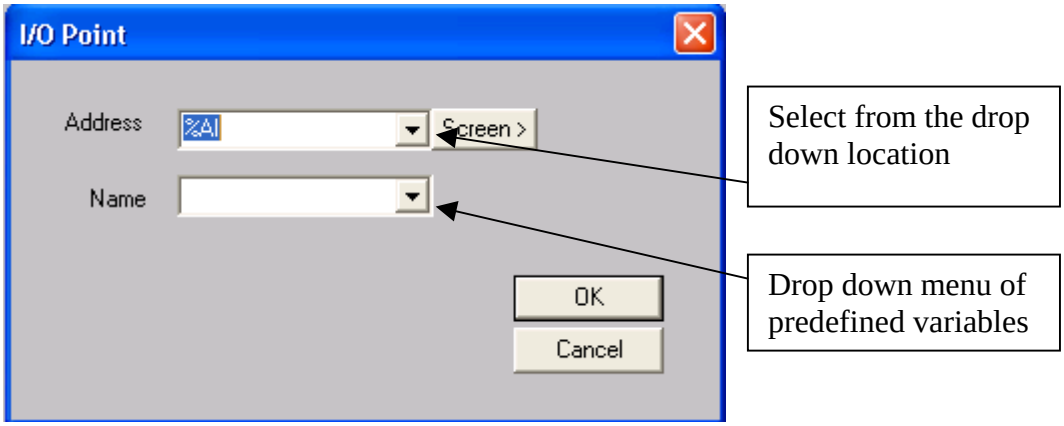



Once you have selected your sub section it is now possible to click on the icon to insert it into the ladder diagram.




Enter the contact / coil / function into the diagram by clicking it into the Ladder diagram.

When inserting a Contact or a Coil you will need to assign the variable to a location and giving it a name is a good idea for documentation.

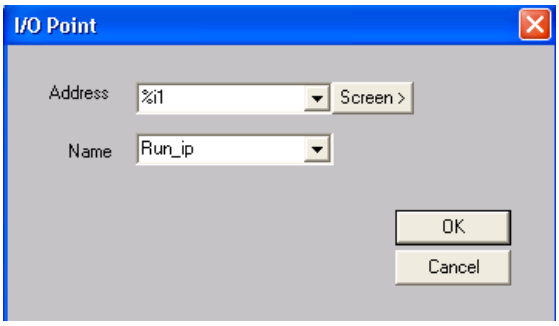


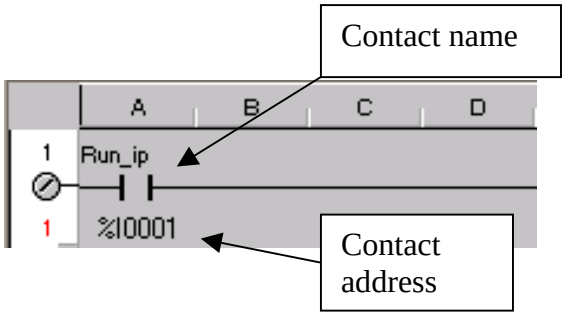
## *i³ Memory Locations*

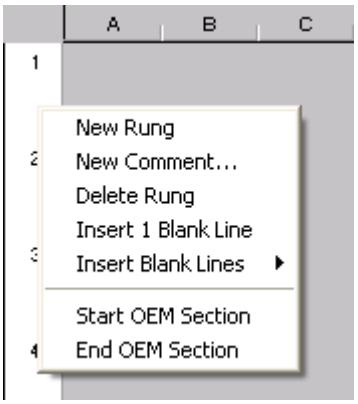
| Type | Description and example of what might use the type | Format |
|---|---|---|
| %I | Discrete Inputs from the field; proximity sensors, panel buttons, etc | BOOL |
| %Q | Discrete Outputs to the field; relays, indicator lamps, etc. | BOOL |
| %AI | Analogue Inputs from the field; Thermocouples, 4-20mA inputs | WORD |
| %AQ | Analogue Outputs to the field; 0-10VDC or 4-20mA outputs | WORD |
| %IG | Global Discrete Inputs from the CAN smart I/O; | BOOL |
| %QG | Global Discrete Outputs to the CAN smart I/O; | BOOL |
| %AIG | Global Analogue Inputs from the CAN smart I/O; | WORD |
| %AQG | Global Analogue Outputs to the CAN smart I/O; | WORD |
| %T | Internal Temporary bits, use for contacts and coils | BOOL |
| %M | Internal Memory bits, use for contacts and coils | BOOL |
| %R | Internal Registers, use for timers, counters & other data | WORD |
| %K | Keypad bits, reflect Function Key status | BOOL |
| %D | Display bits, control screens or indicate screen on/off | BOOL |
| %S | Internal System Bits (See System Registers) | BOOL |
| %SR | Internal System Registers (See System Registers) | WORD |

Note: The allocation of I/O starts from 1 the first input is %*01 and **not** %*00

.



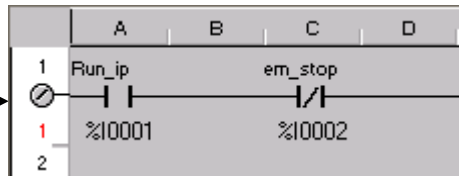




By right clicking the mouse in the left margin we can have some more options.
We have the options to add or remove rungs, insert documentation and start OEM sections.

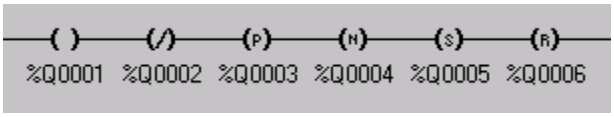
# Input / Output Types

The ***i***$^3$ has 2 types of inputs: Normally Open (N/O) and Normally Closed (N/C).

*The rung has been "Made" when this symbol is present.*



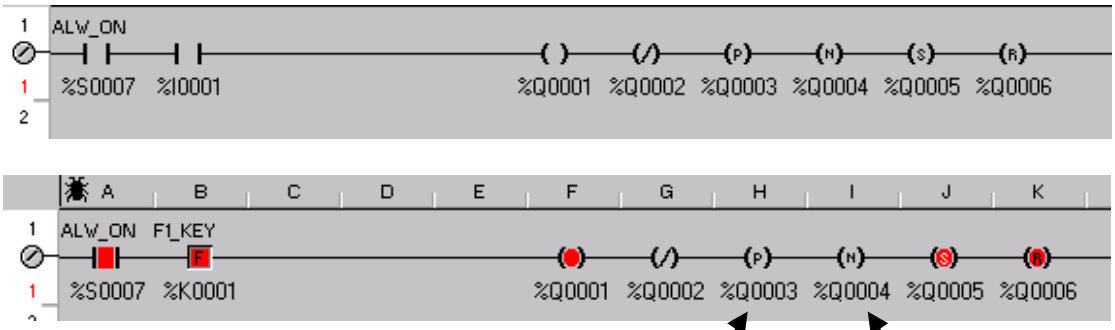

The ***i***$^3$ has 6 types of outputs: N/O, N/C, Positive Edge, Negative Edge, Set and Reset.




More than one output can be put on a single rung. This performs the same function as OR'ing the outputs. When the input condition is met all of the outputs on the rung will be activated.

**Example 1:**

When %I0001 is enabled, %Q0001 switches state, as does %Q0002. Whereas %Q0003 switches on the positive edge of %I0001 and %Q0004 will switch on the negative edge of %I0001. %Q0006 will reset from its current state.







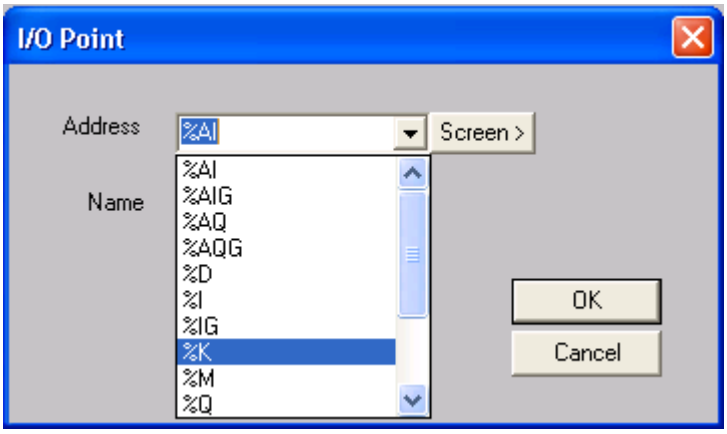
Q3 was on only on the positive edge of the input

Q4 was on only on the negative edge of the input

# Assigning Function Keys and Screens

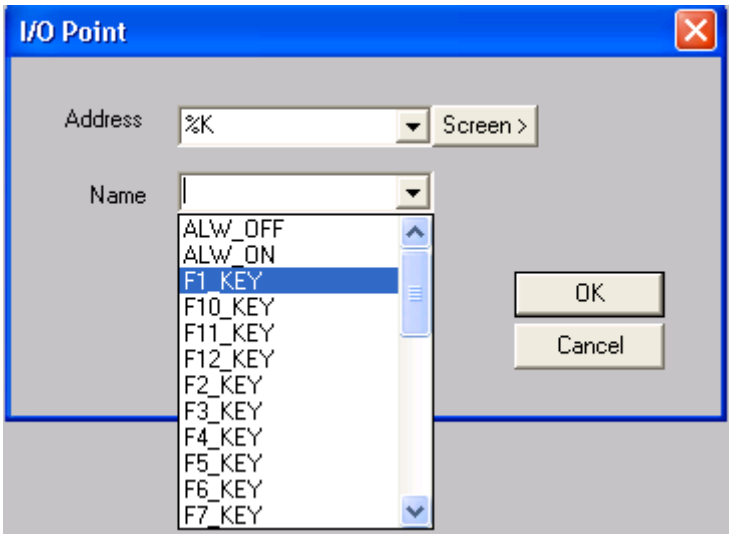
The ***i***$^3$ has 11 function keys that can be programmed into the ladder diagrams. It also has four buttons that can be programmed through the screen editor which are discussed later in this document.

Assigning Function Keys in the ladder diagram is very simple. Select an input contact and address it to the Keys!




Select %K and then assign the address for the key, 1-12

Or, select it by its predefined name.

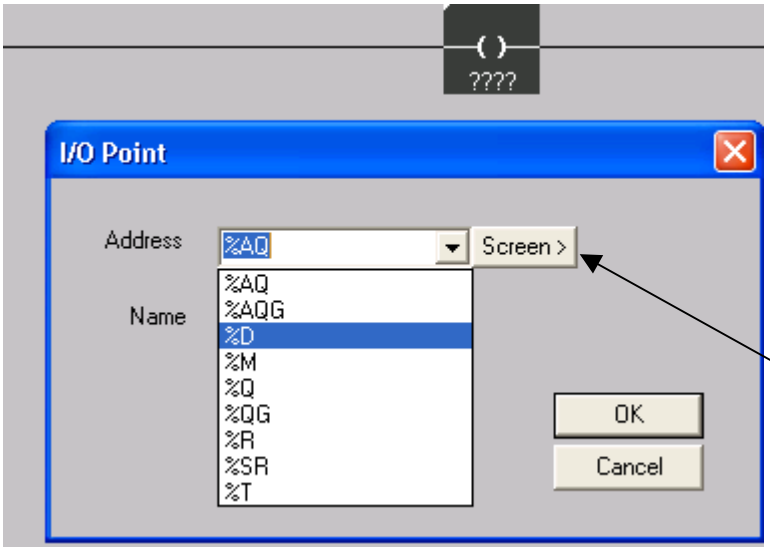




NO Function Key contact

NC Function Key contact

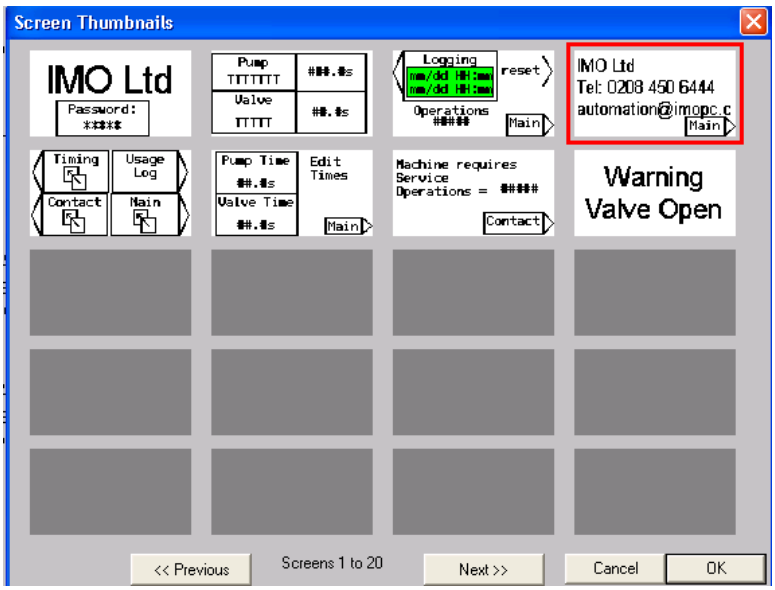
To insert a screen into the ladder logic, select an output coil and click it into the ladder diagram.

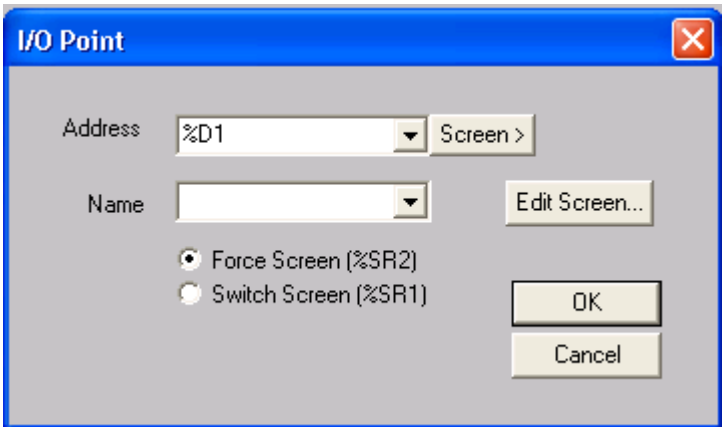



Select %D and then enter the number for the screen to display, i.e. %D0001 is screen 1.

By clicking the screen button you can choose a screen from the thumbnail menu.

*The screens shown were set up previously. If no screen has been set up then it will be shown as a blank grey box.*



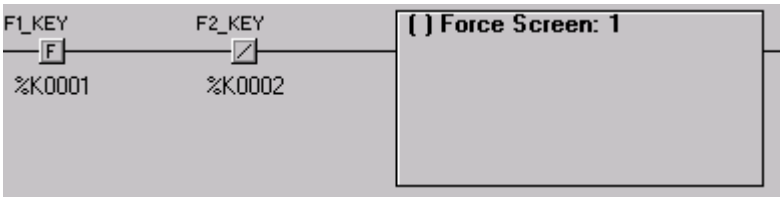

We can open the screen editor from this menu which is covered later in this document.

There are now two options to choose from which control the way an individual screen is presented.

**Force Screen:**

Displays the screen whilst the conditions to make the contact are met. When the condition is not met anymore the screen will go back to the previously set screen.

**Example 2:** Warning Message when a valve is open.

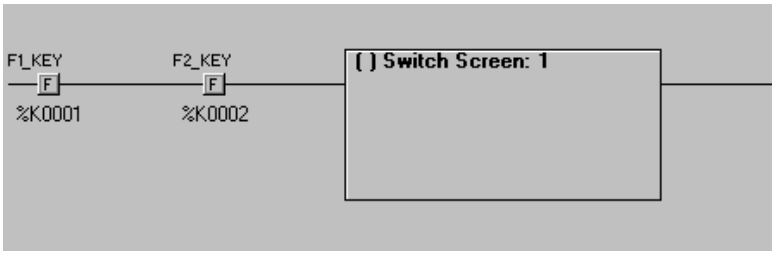



When F1_KEY *AND NOT* F2_KEY are pressed the ***i***$^3$ will display screen 1. When F1 has been released the ***i***$^3$ will display the previously set screen.

**Switch Screen:**

Changes the display to the desired screen when the condition is met.
When the logical condition is no longer true the selected screen remains on display.

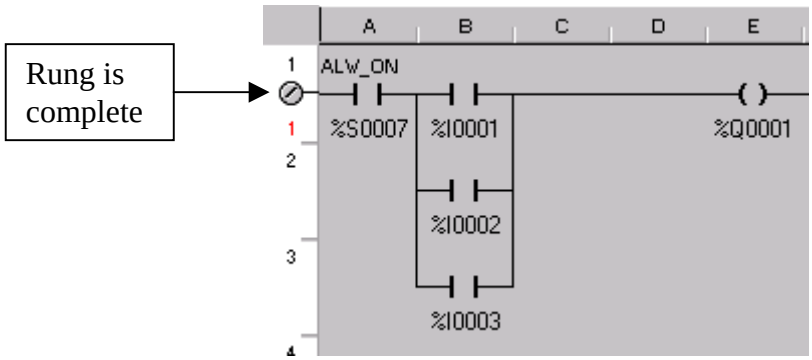
**Example 3:** Changing screens on a menu selection.

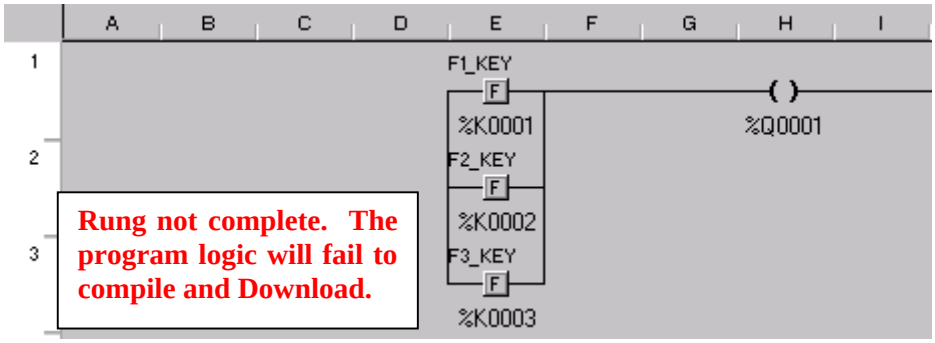


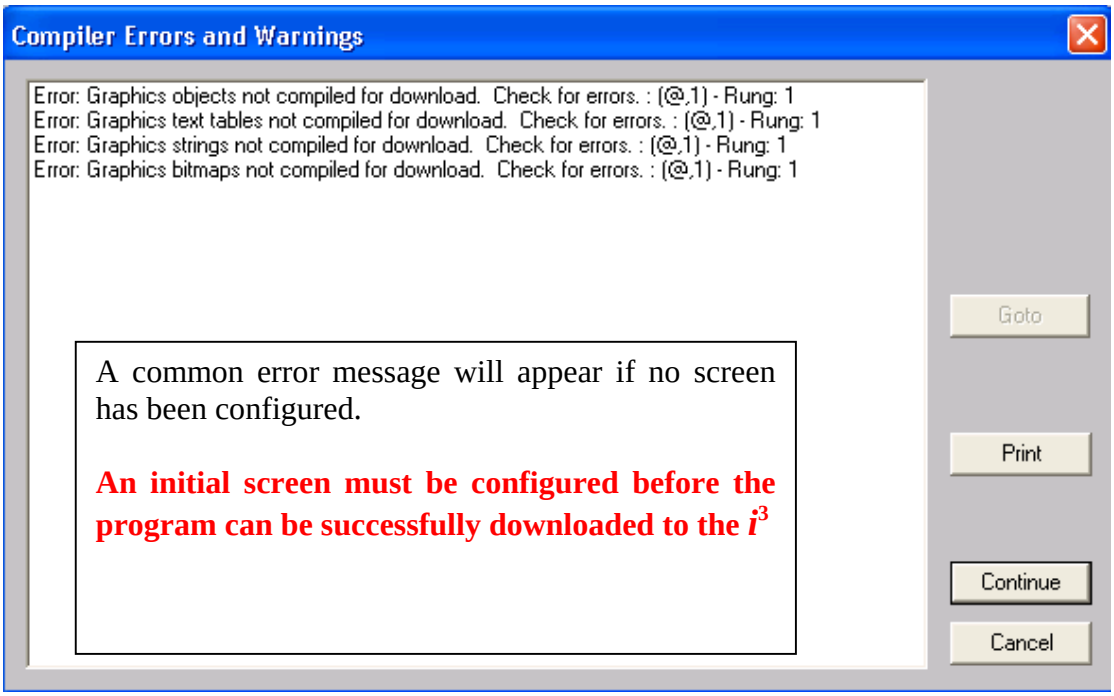

When F1_KEY *AND* F2_KEY are pressed the ***i***$^3$ will display screen 1. When F1 has been released the ***i***$^3$ Screen 1 will remain on the display until the next Screen is called.

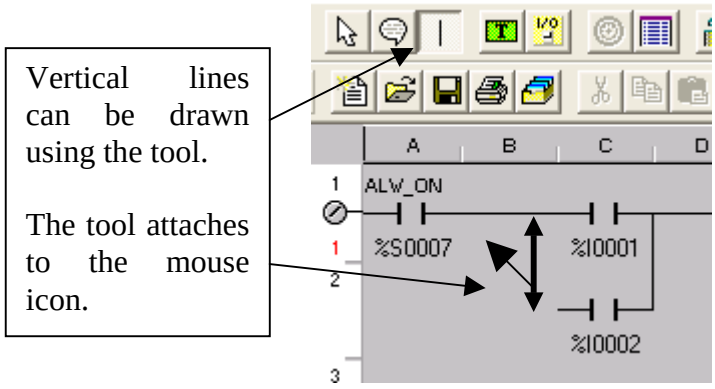
# Basic Ladder Logic

Using Ladder logic it is necessary to insert a contact in the first column. A good practice is to insert an ALWAYS ON (ALW_ON) contact at the beginning then add your logic after.

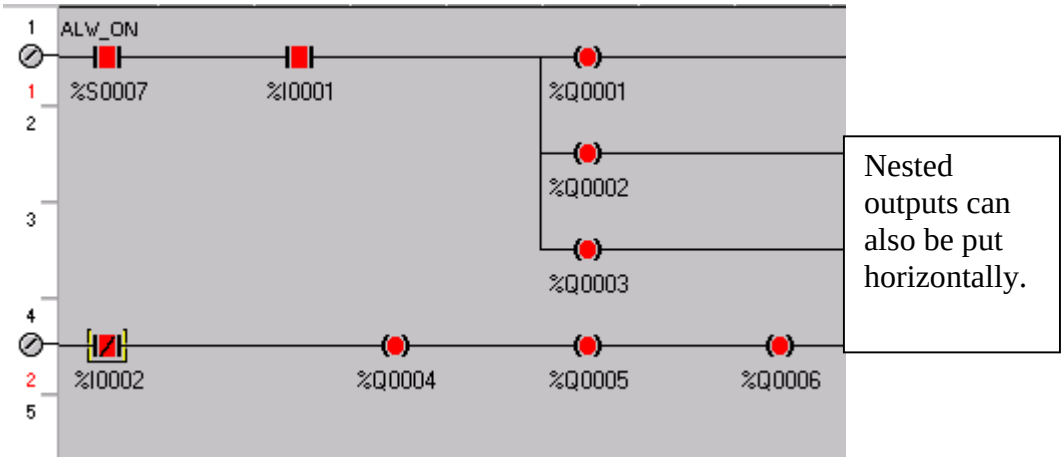


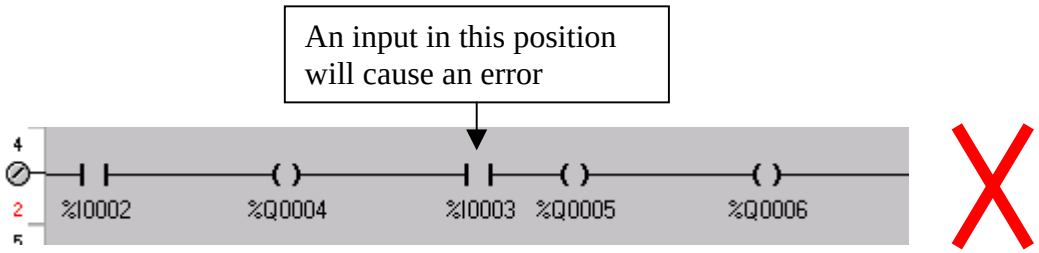



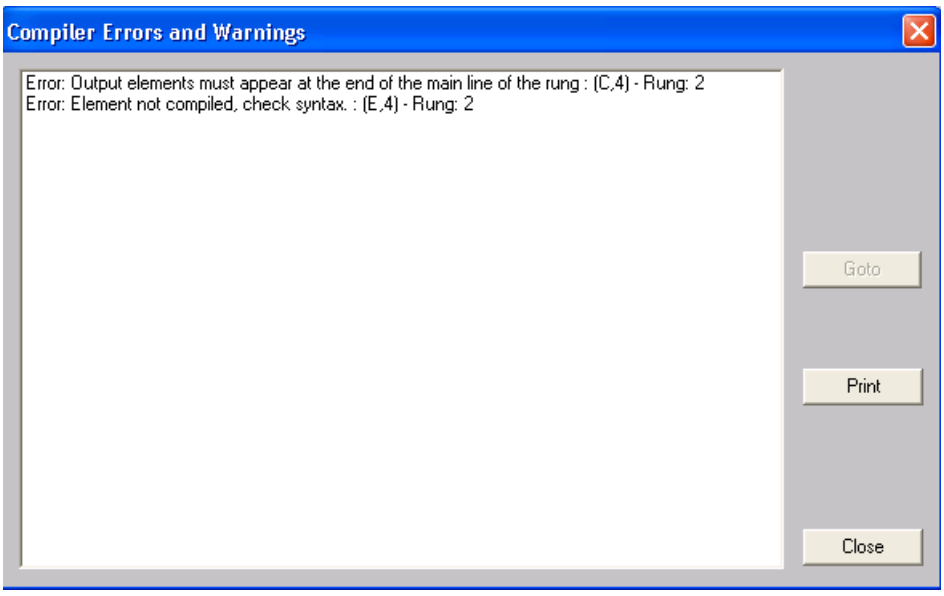






Each rung can have multiple outputs but they must be at the end of the line.

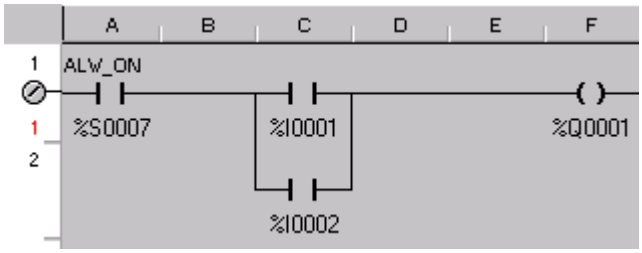


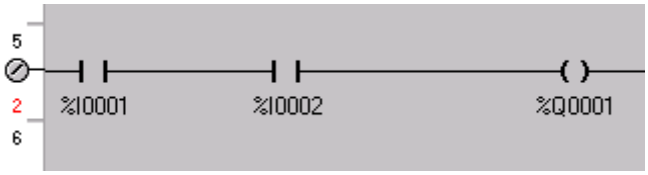
## *Or Gate*

When any one of two or more inputs will switch on the output:  In this example either %I0001 or %I0002 will switch ON %Q01.




## *And Gate*

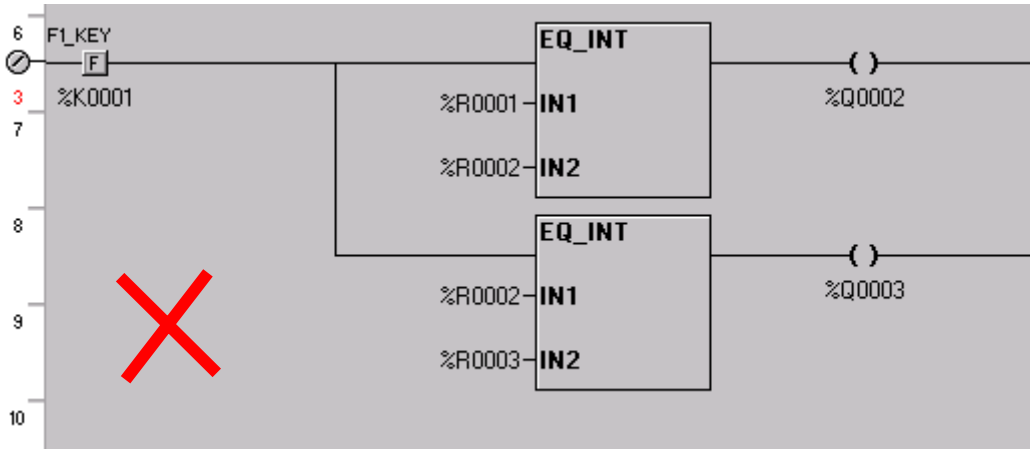
When all input conditions on a single rung are met the output will switch ON.  In this example both %I0001 AND %I0002 must be on for the output %Q0001 to be energised.

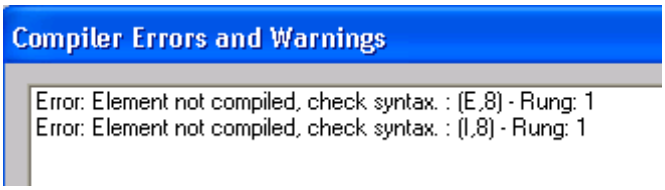


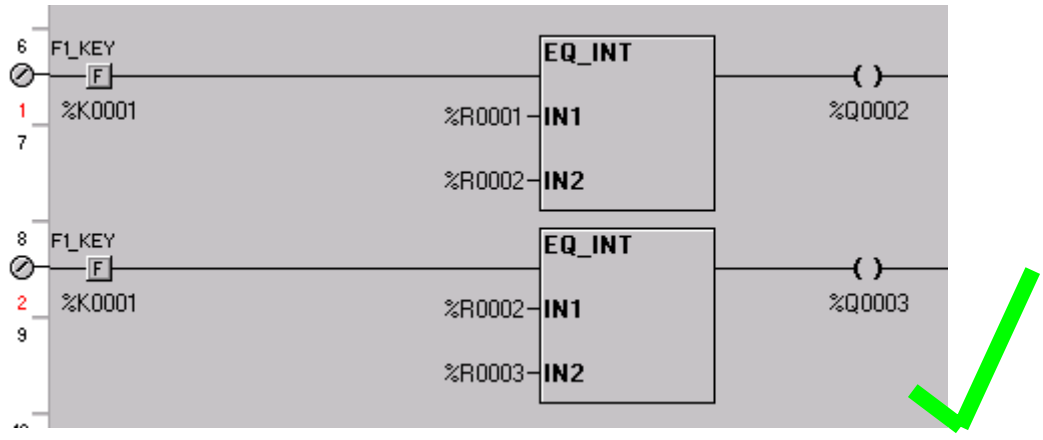
## *Function Blocks*

All function blocks have an enable input and a Boolean output. Other I/O parameters are dependent on the individual function block and are required to be set up when inserting the block.

Nesting function blocks will create an error.



## *Timers and Counters*

Timers and counters require two consecutive registers to store their data in memory. The first register contains the current value and the second contains the status bits of the counter or timer.

Register 1 = Accumulated value

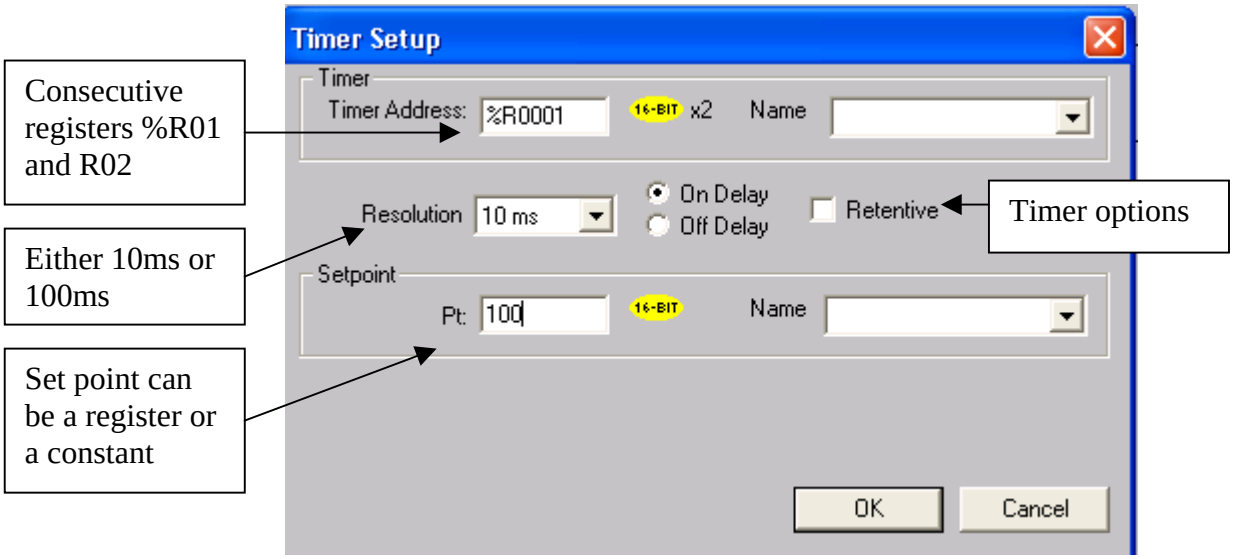
Register 2 .15 = Function Enabled
Register 2 .16 = Function Completed i.e. timing elapsed, preset value met.

### Timer Set up

There are two types of timer, on delay and off delay. The On delay timers can be retentive.

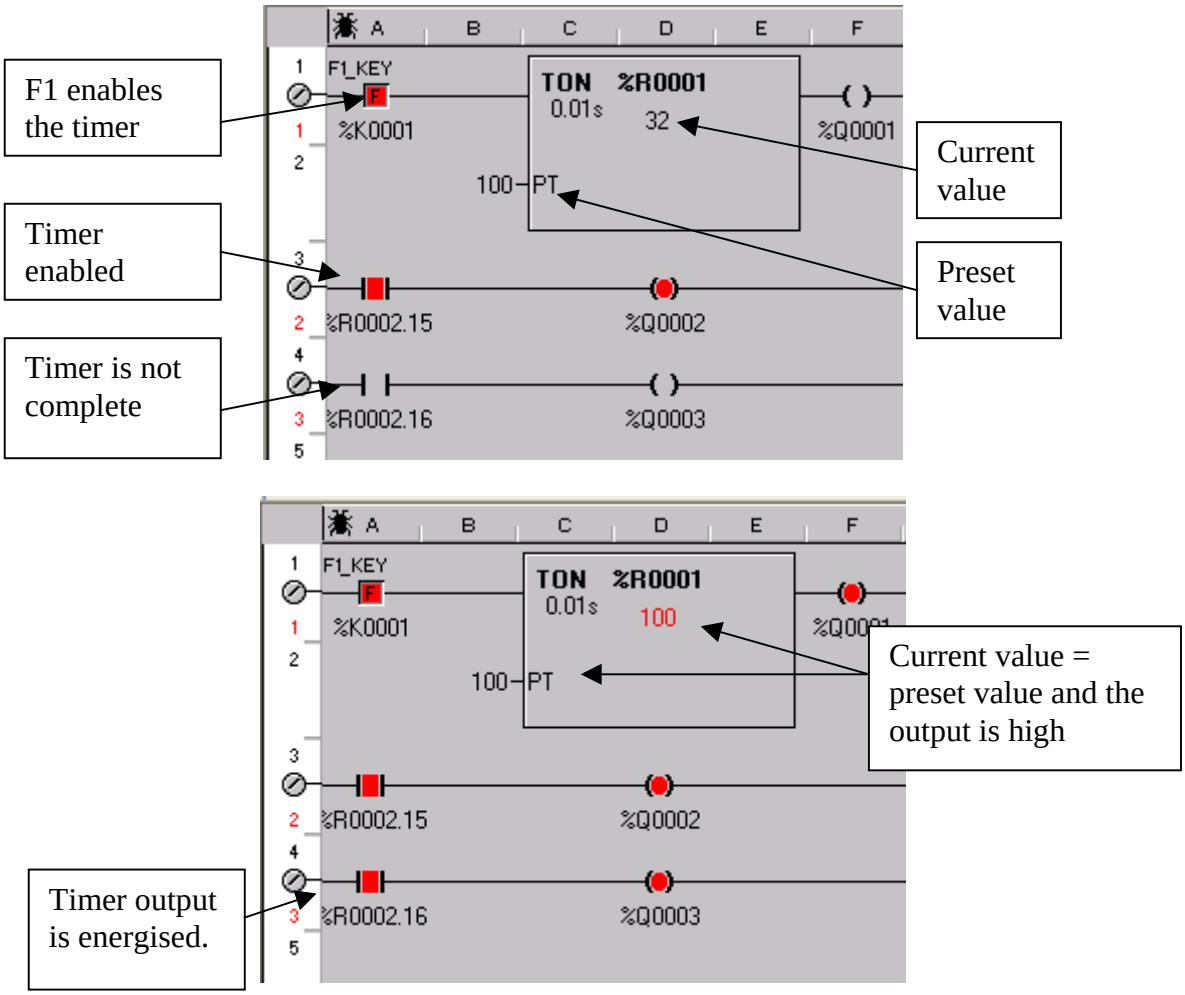
**On Delay**

Delays before the output goes high. On being enabled the timer starts. When elapsed time reaches the Preset the output will energise. The output will remain energised until the input to the timer has been removed.



**Example 4:**

This is an example of a 1s On delay timer.

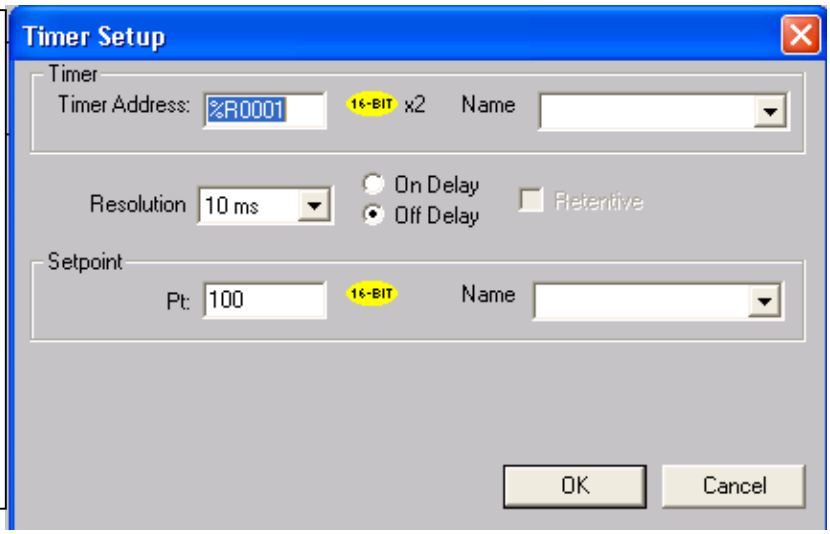


## Off Delay

The Off delay timer provides a delay before switching the output off when the input is enabled. When the input is removed the timing will begin, when the elapsed time equals the preset the output will reset.

The setup window is the same as the previous example but the "Off Delay" button is checked.

Notice that the Retentive button is no longer available

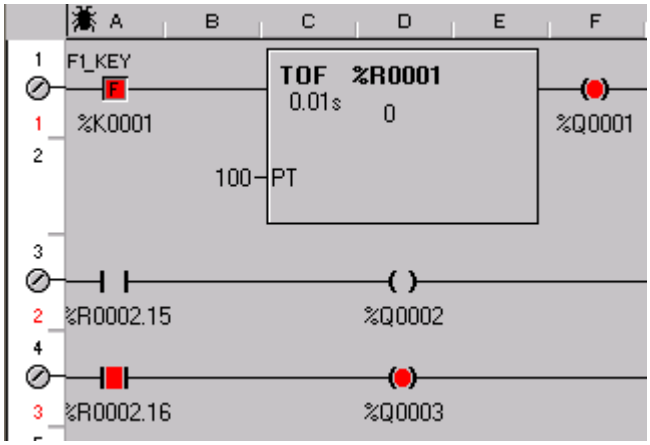


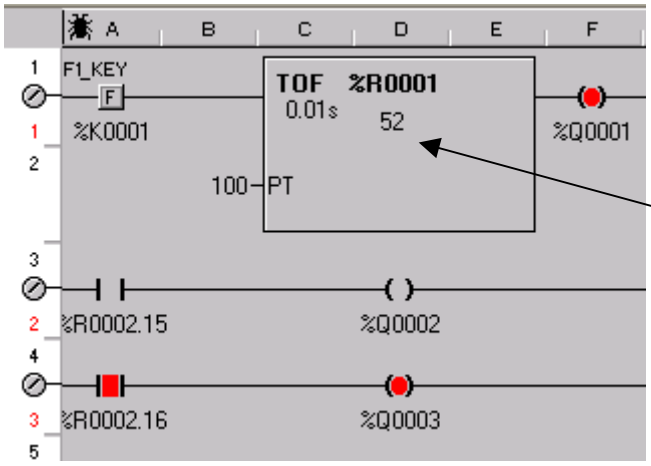

**Example 5:**

This is a 1 second Off-delay Timer. The output (%R0002.16) is energised when the input (%K0001) is made and will remain energised for 1 second after the input is removed.

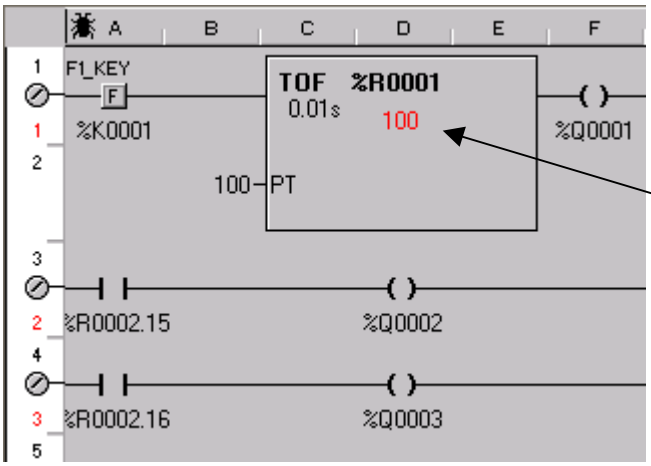
The input to the timer has been enabled and the output is high.

When the input is remove the timing begins.






The output remains high, whilst the current value is accumulating.

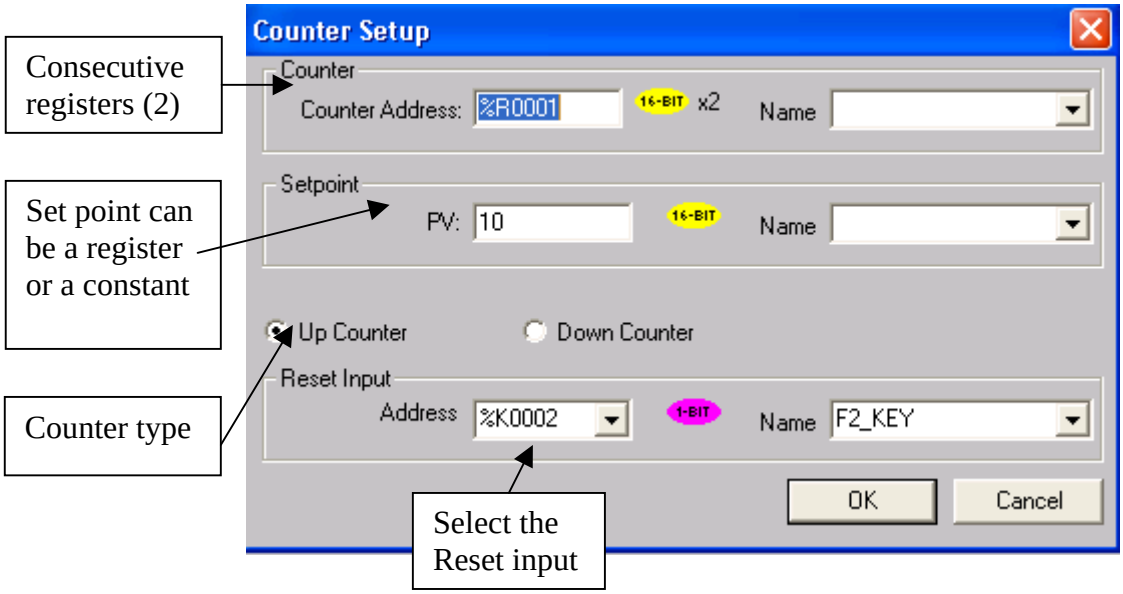


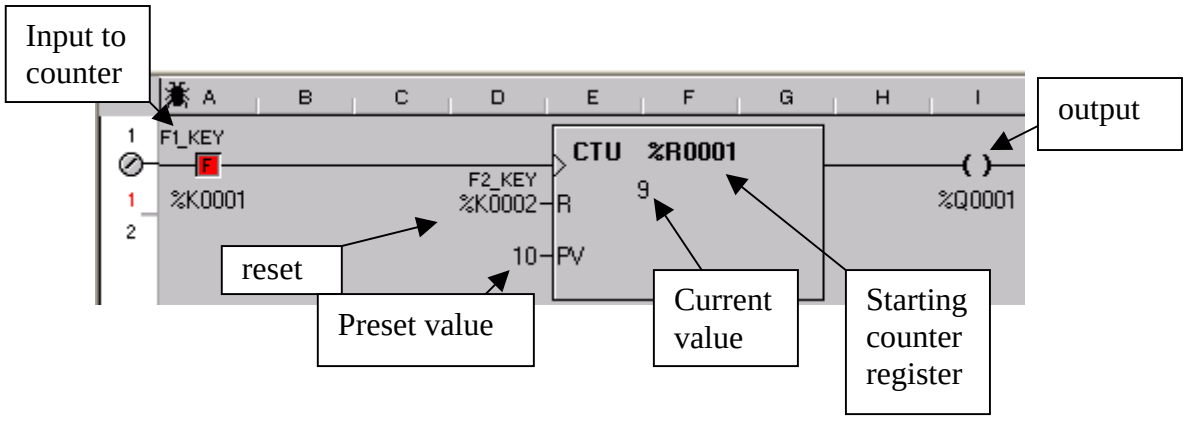

The current value now equals the preset value and the timer is now off.

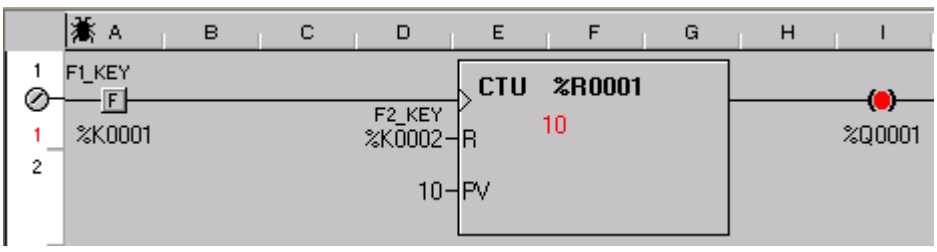
## Counter Set up

There can be two types of counters, Up Count and Down Count. All the counters increment / decrement the current value on the positive edge of a defined input and require a Reset input.

An up counter will increment its current value from 0 to the preset value at which point the output is energised. A down counter will decrement its current value (starting at a value equal to the Preset) until the current value reaches 0 at which point the output is energised. Both counters will overtake the preset if input pulses continue to be applied to the counter after the output is energised.

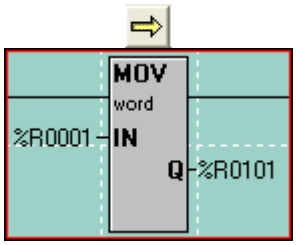


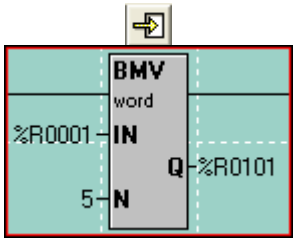




The count value has reached the preset energising the output.

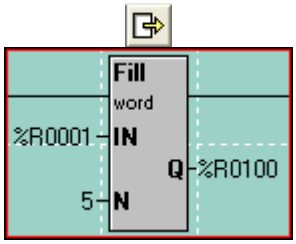
## *Move Functions*



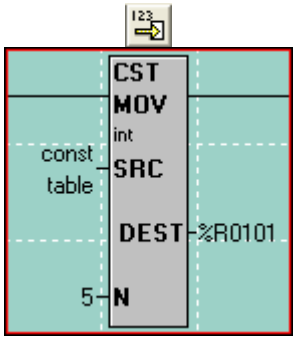

The first type of Move is the ‘Move Word’, or ‘MOV’. It is used to copy a single byte, word or double-word from one location to another. The count is locked at 1. In the case of the example to the left, the value in %R1 is copied into %R101. This only happens when the ladder rung receives power. The value in %R101 is NOT taken back out when power is lost to the rung. The IN can be either a register or a constant value.



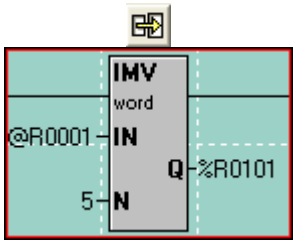

The next type of Move is the ‘Move Data Block’, or ‘BMV’. It is used to copy a group of bytes, words or double-words to another location. The count (N) determines how many registers are to be copied. In the example to the left, %R1-%R5 are copied into %R101-%R105. Again, this only happens when the ladder rung receives power. The IN must be a register reference and constant values are not allowed.




The next type of Move is the “Fill WORD”, or “Fill”. It is used to copy the contents of a single register or value into multiple other registers, thus filling that one value into a group of registers. The count (N) determines how many registers to fill that single value into. In the example to the left, the value in %R1 is copied into %R101-%R105 so that %R101-%R105 all will have the same value in them. This can be used to zero-out a group of registers. The IN can be either a register or a constant value.

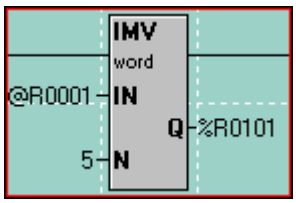



The ‘Constant Move’, or ‘CST MOV’, it is used to move a group of constant values into a group of consecutive registers. If, for example, you want to move the values 1, 2, 3, 4 and 5 into %R101, %R102, %R103, %R104 and %R105, respectively, then you can use the Constant Move function. The count (N) is automatically determined by how many constant values you enter into the configuration for this function. The source data can ONLY be constant data and cannot be register references.

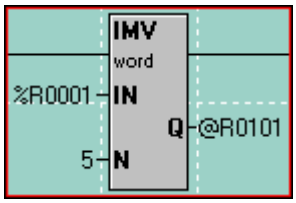



The ‘Indirect Move’, or ‘IMV’, it is used to move data from variable positions or to variable positions or both. It functions, for the most part, like the Block Move function. If specified as Indirect, the IN and/or the Q are used as pointers to where in the %R registers to get data from or put data to. When looking at the ladder logic, the @ symbol will appear next to the IN or Q address if it is specified as Indirect. This function is used in data logging applications.

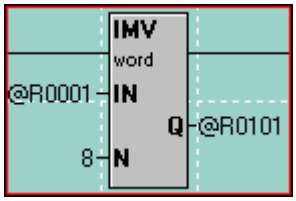
## Indirect Move Examples




In this example, the IN is specified as Indirect. This means the controller will look at %R1 and see a value within it. If %R1 has a value of 501 in it, the controller will go to %R501 to get the source data. 5 registers will then be moved from %R501-%R505 to %R101-%R105.




In this example, the Q is specified as Indirect. This means the controller will look at %R101 and see a value within it. If %R101 has a value of 851, the controller will take the data in %R1-%R5 and move it into %R851-%R855.

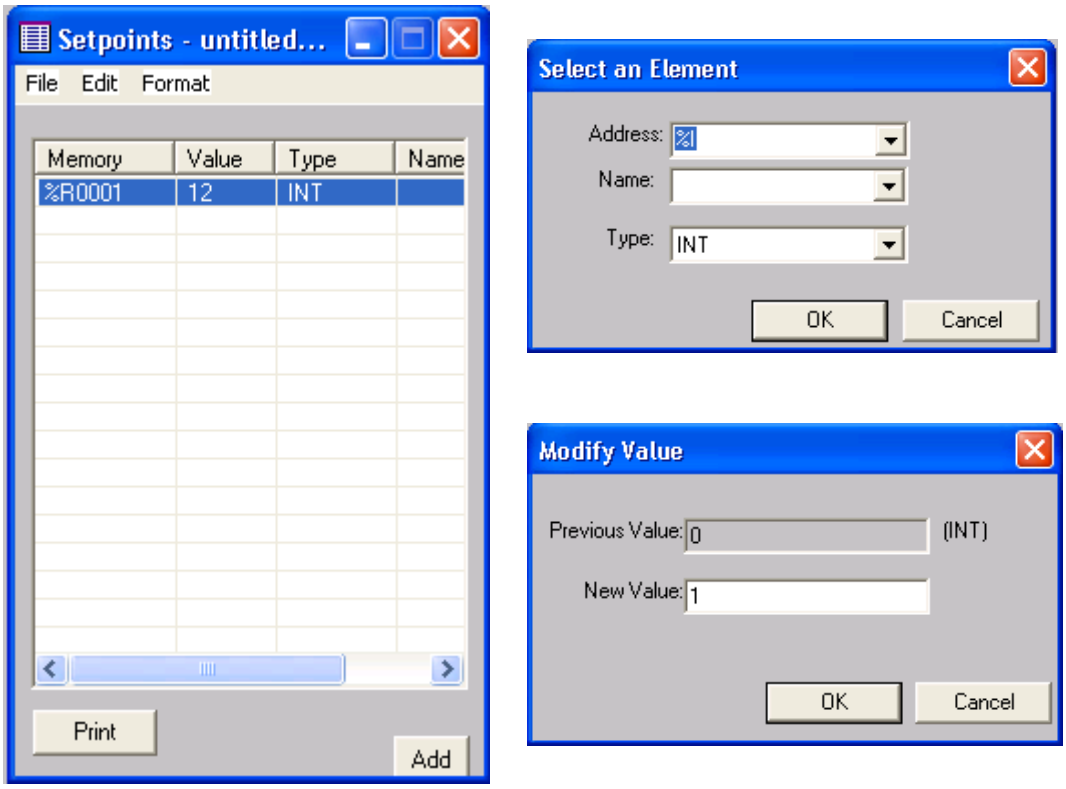



In this example, both the IN and the Q are specified as Indirect. This means the controller will look at %R1 and see a value. Let's say it is 241. The controller also looks at the value in %R101. Let's say it is 341. The controller will then take the values in %R241-%R248 and move them into %R341-%R348.

## *Set Points*

It is possible to set registers with initial values by using the set point editor. To open it, select the Setpoint option from the program menu.

From within the editor we can "Add" a new register. Double click on the Value column to enter a Setpoint.

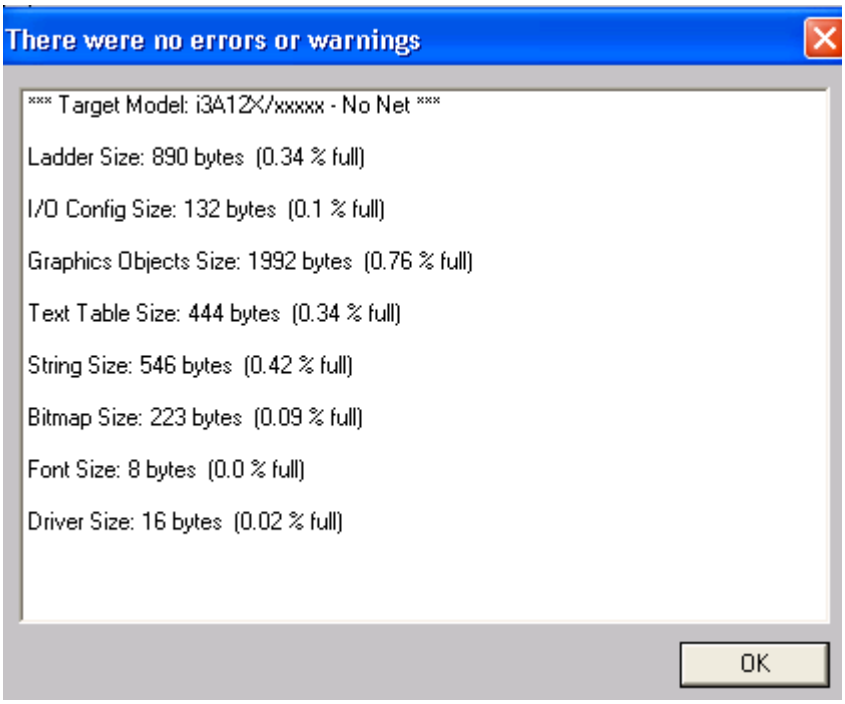



**NOTE:**

The Setpoint button must be selected in the Download Options when you are ready to download the application program into the ***i***$^3$.

## *Download Options*

Once you have created the Ladder logic and user screens the program can be verified by clicking the icon  . This will check the program for any errors. If there no errors are detected in the program or configuration of the ***i***$^3$ the download can proceed.




## Connecting to your i3

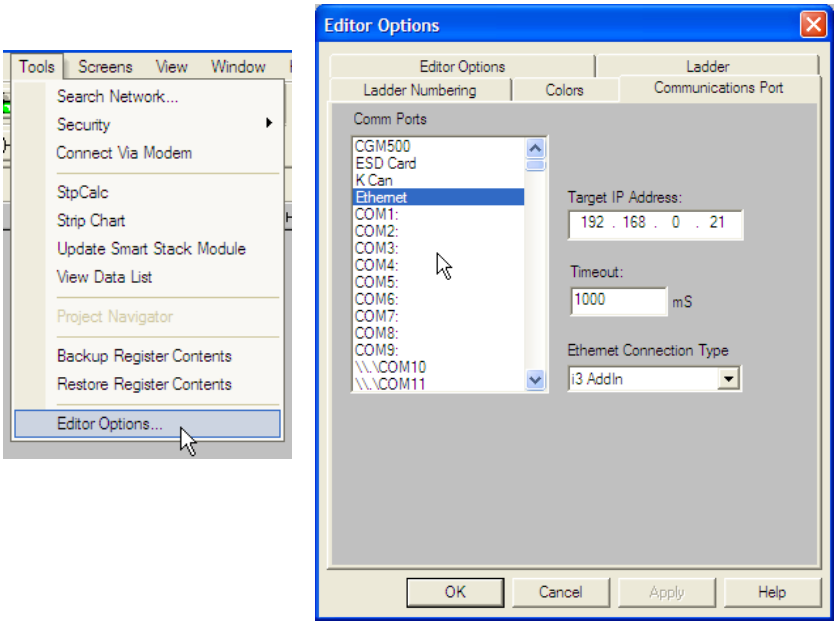
It is important to understand a little of how the ***i***$^3$ Configurator software has been designed in order to do this correctly. The Configurator is also a complete Project Management System capable of connecting to multiple ***i***$^3$ Controllers simultaneously to debug an entire ***i***CAN network. This means that you may have as many as 252 ***i***$^3$ Controllers linked together via the CAN port. Then by linking your PC via an Ethernet, Serial, or Modem link to just one of them; you can debug and monitor them all. So therefore, each project has the Network ID for that device stored as one of the settings.

An ***i***$^3$ that does not have a CAN port, such as the basic 10A01 model, has a default network ID of 253 (above the usable range of 1 – 252) as it will never be part of a network. When creating a new project for a device without CAN the ***i***$^3$ Configurator defaults the network ID to 253. When creating a new project for a device with CAN the software defaults the network ID to 1 – as is the default setting of a CAN enabled ***i***$^3$. Problems can occur if a project is started with the device configured for one option,

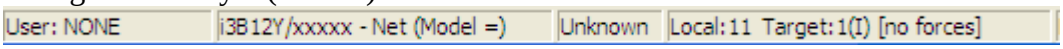
then switched to the other at a later time. The result will be a mismatch in the network ID settings between the Configurator project and the ***i*³** (it is trying to communicate with ID 253 when the device is set at 1, or communicate with ID 1 when the device is set to 253).

Also, the comms engine of the ***i*³** Configurator constantly runs in the background as soon as the software is started. The first indicator that Comms are established is that the 'Traffic Lights'  will show the status of the connected ***i*³**, as one of the buttons will appear depressed.

If this does not occur then the comms settings can be verified as shown below. This shows an Ethernet connection. Care must be taken if using a USB to Serial Adapter that the Com port assigned to the adapters matches the one selected through these options.



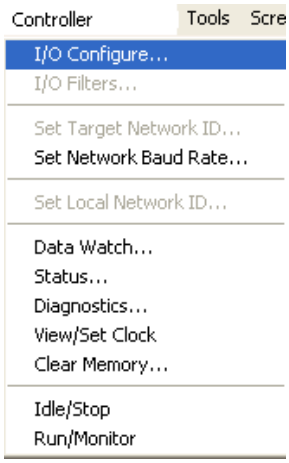

At the bottom of the Configurator screen we can see the status bar. This is showing that we are Locally connected to the ***i*³** of Network ID 11. This is the device that we are physically connected to. But, it is showing the Target ID of 1. So, despite being connected to ID 11; we are actually configuring/debugging/monitoring the ***i*³** with an ID of 1. It is showing us that this is an i3B model with CAN Net, but no I/O configuration as yet (/xxxxx).



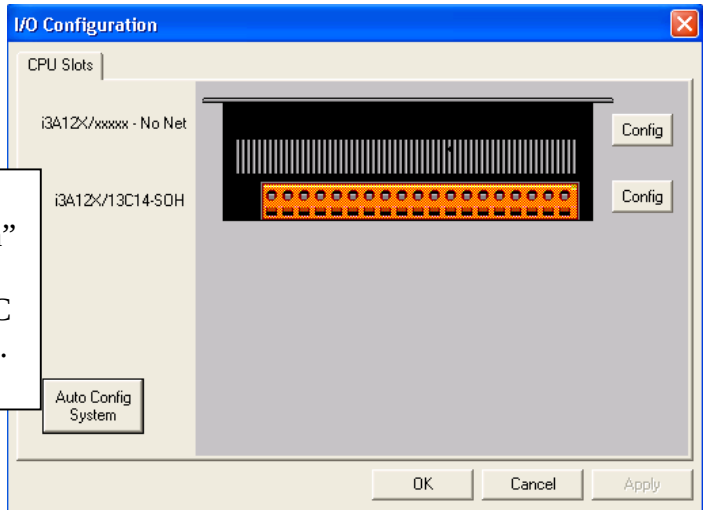
## Configuring the correct model.

Before downloading the program it must be configured to match the model of ***i***$^3$ to be used.

Select the I/O configure from the controller menu.




Then click the "Auto Config System" button ensuring the ***i***$^3$ is connected to the PC with an i3PC45 cable.

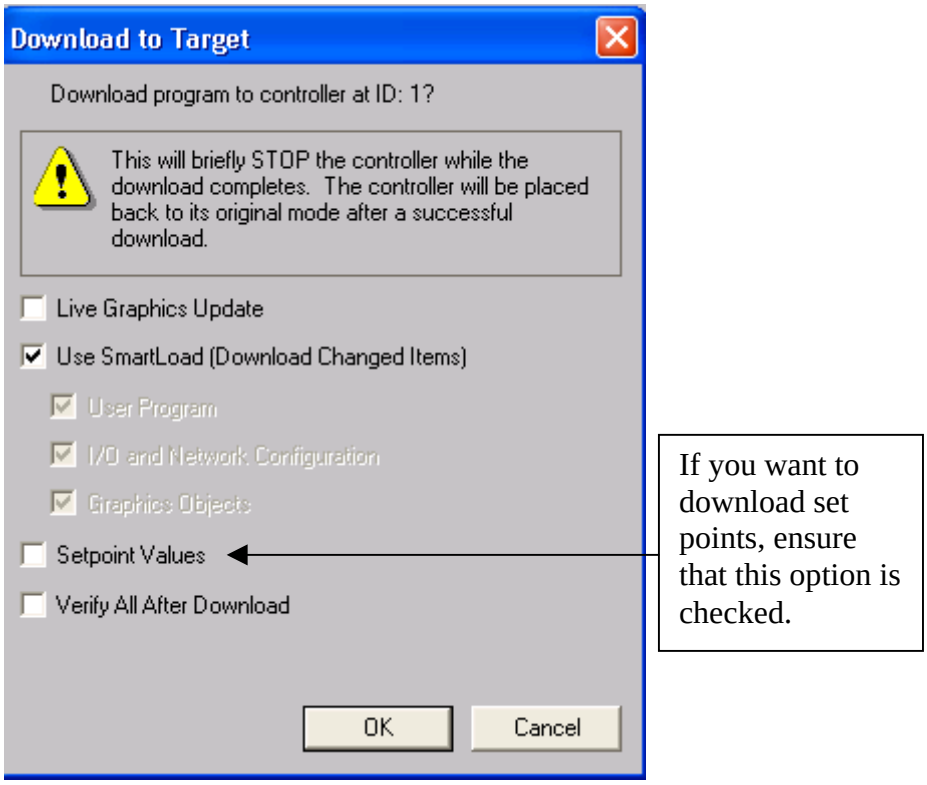



When configured the ***i***$^3$ part number will match the unit.

## Downloading the Program

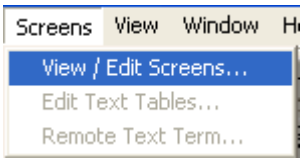
Then to download to the ***i***$^3$ click the icon  or select Download from the Program Menu.

The “Download to Target” menu will appear.



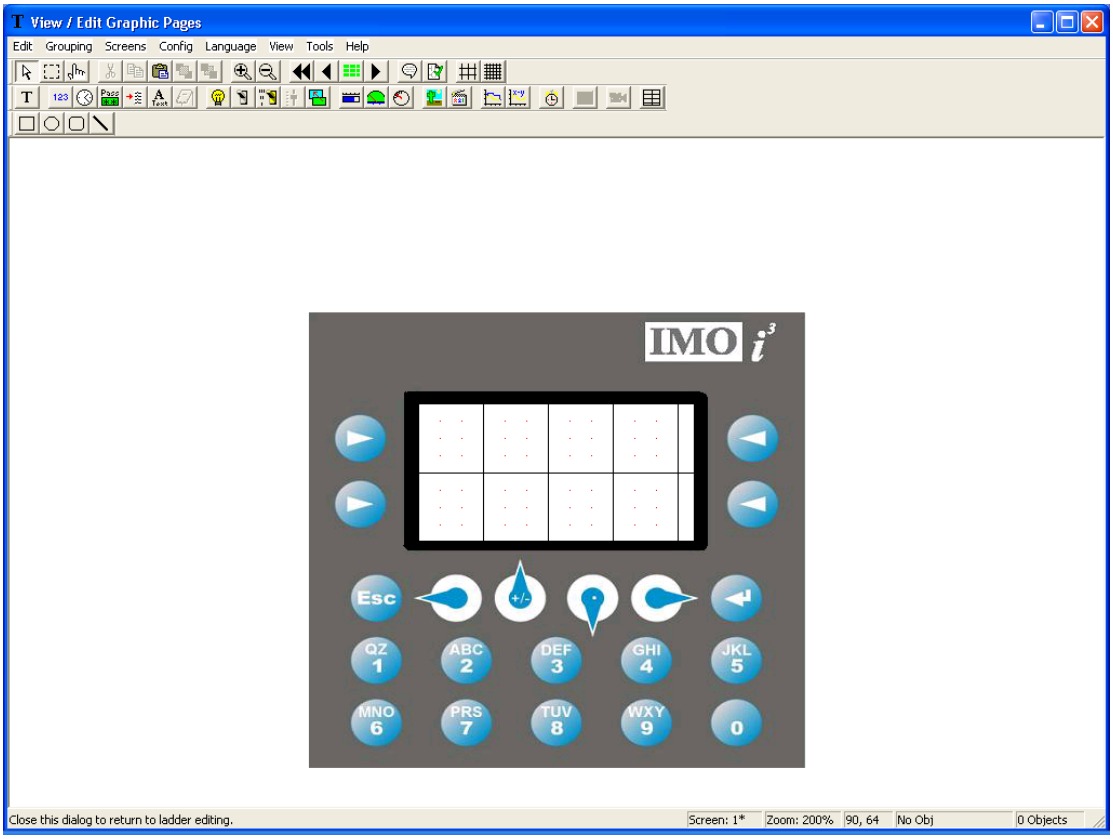
# Screen Editor

To open the screen editor click on the icon  or select the option from the screens menu.



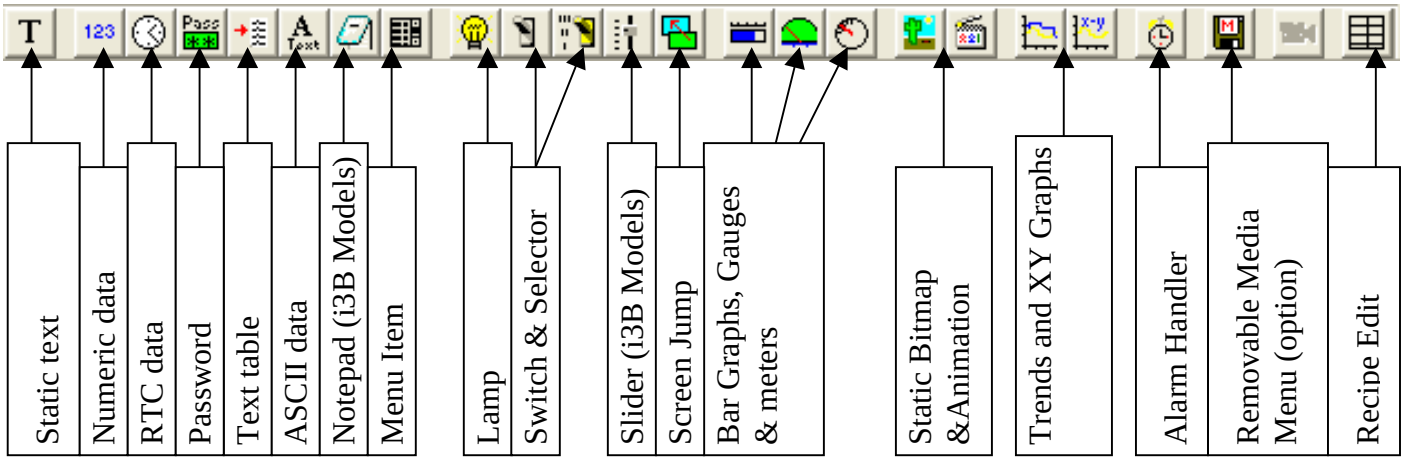

Please note that a default screen needs to be configured before a program can be downloaded to an ***i***$^3$.

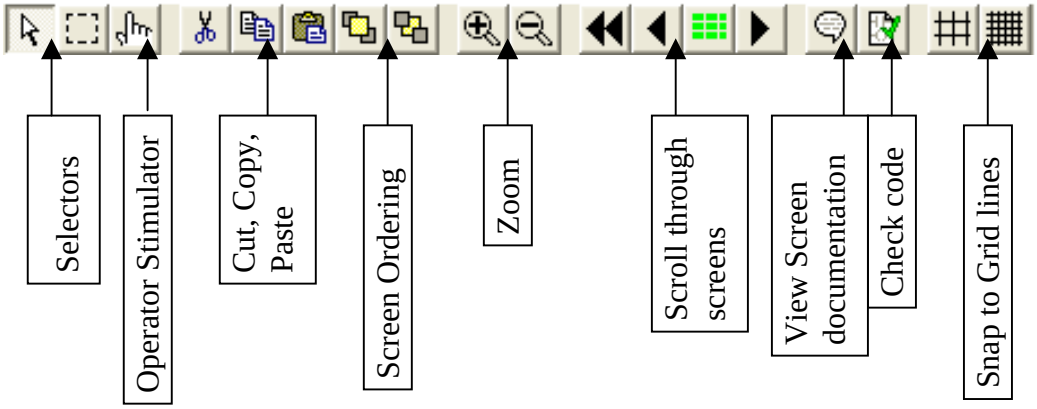
Before editing screens it is important to configure the I/O as previously described.




The screen editor program shows the ***i***$^3$ in the middle with the programming functions at the top and screen information in the bottom right corner. To exit click the top right corner where the X is.

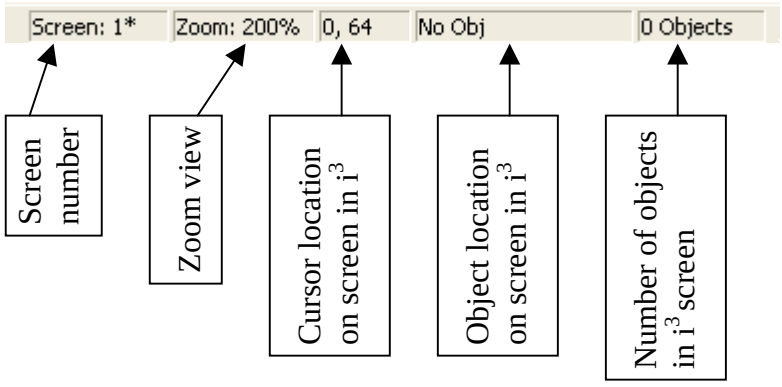
### *Screen Editor Tool Bar.*




Drawing Tools 

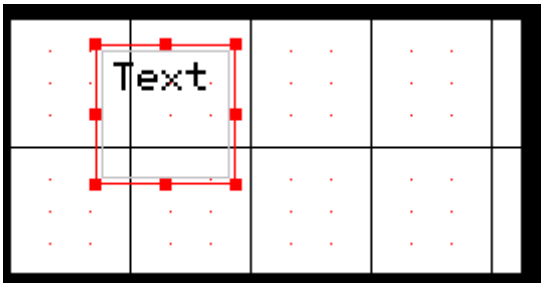
Bottom right corner of Editor

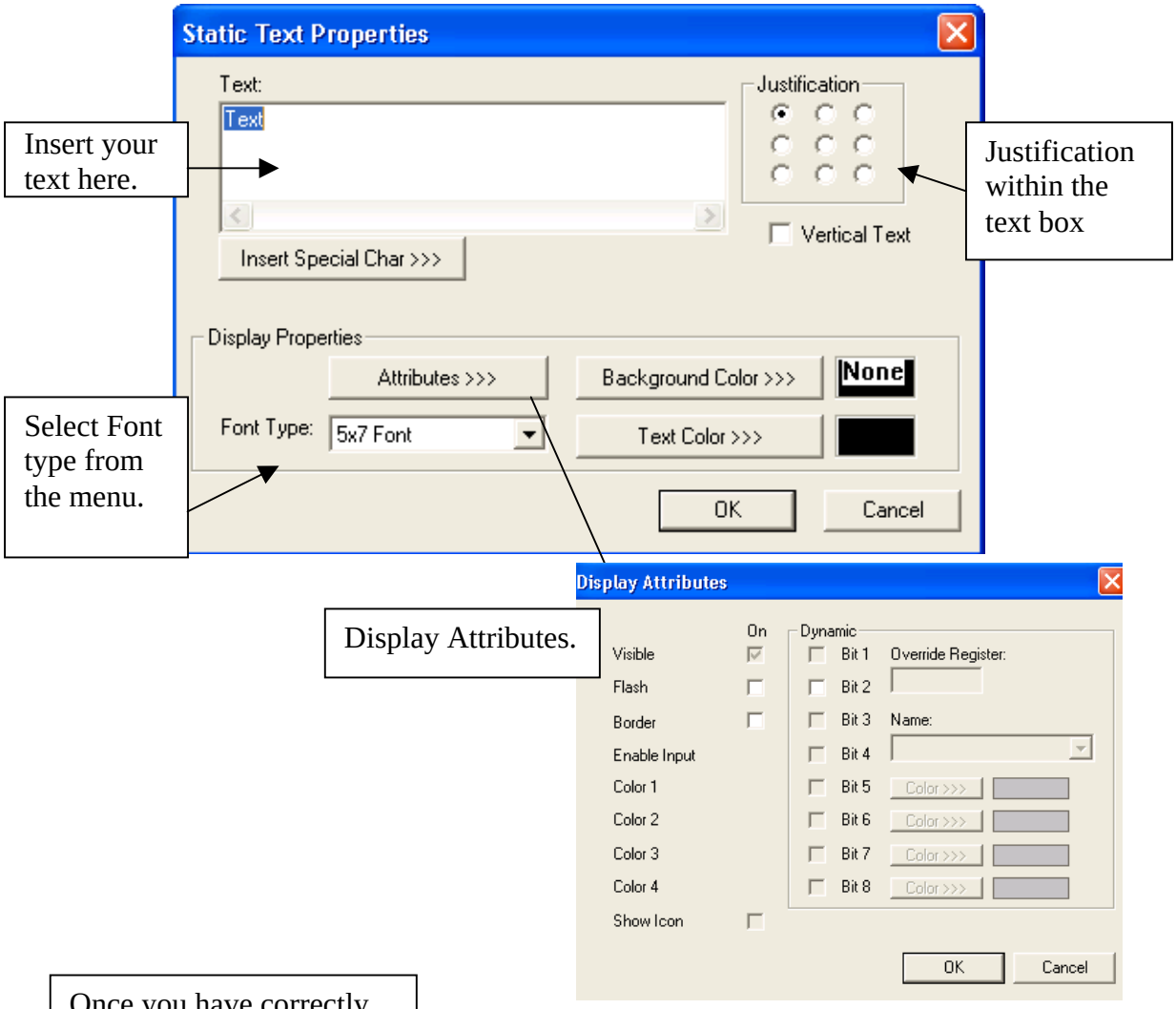


## *Static Text*

To insert a Static Text message click on the icon T and click it to the screen.

The box can be resized as required. To enter the text to be displayed double click on the box.



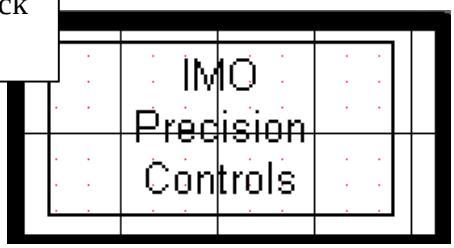

Insert your text here.

Justification within the text box

Select Font type from the menu.

Display Attributes.

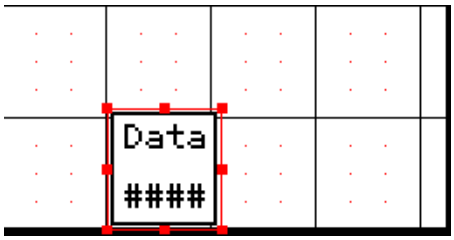
Once you have correctly set up the static text, click OK to confirm and exit



## *Numeric Data*

To insert a numeric data display click on the icon 123 and click it into the screen.

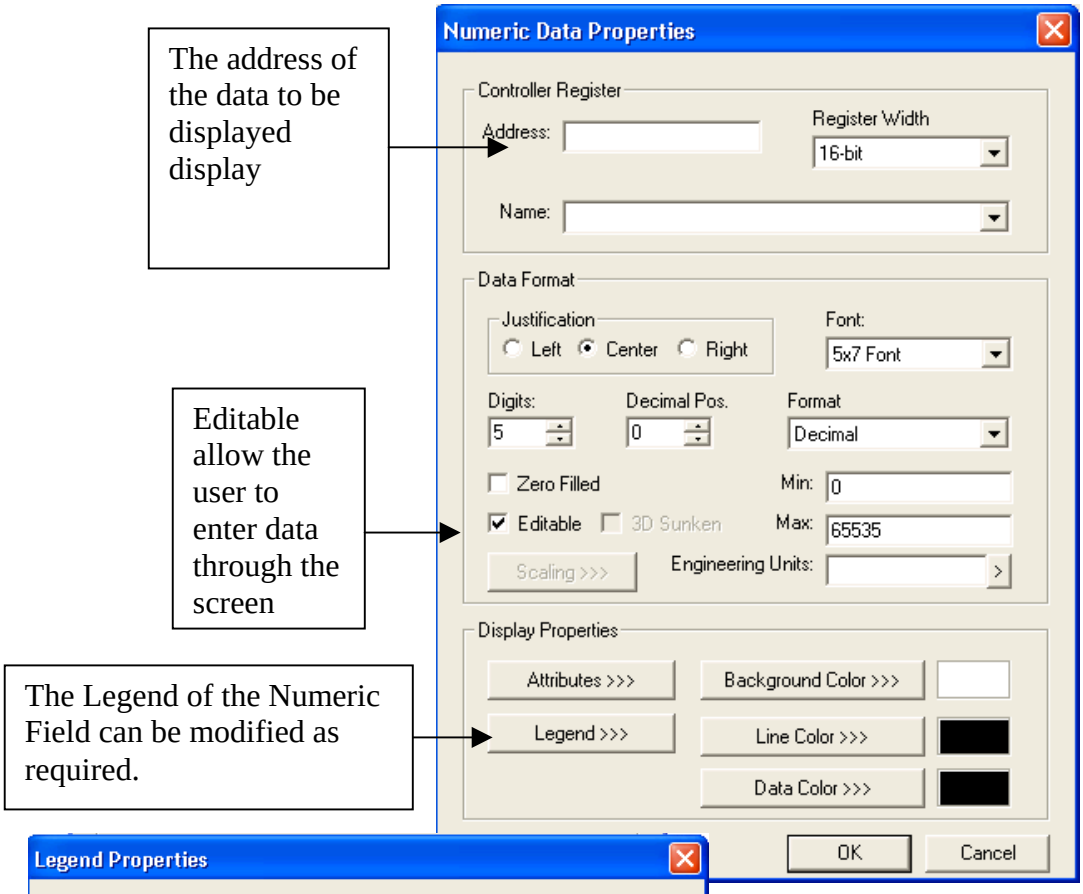
The box can be resized as required. To Enter the details of the numeric data double click on the box.

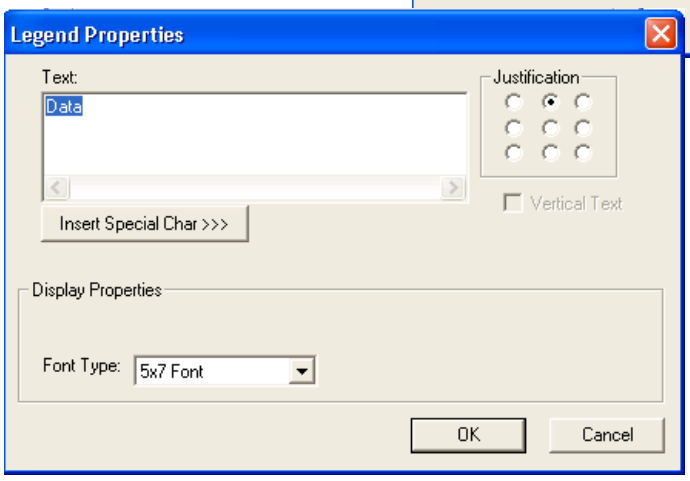


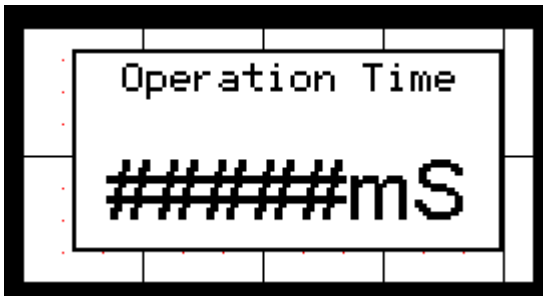

The address of the data to be displayed display

Editable allow the user to enter data through the screen

The Legend of the Numeric Field can be modified as required.



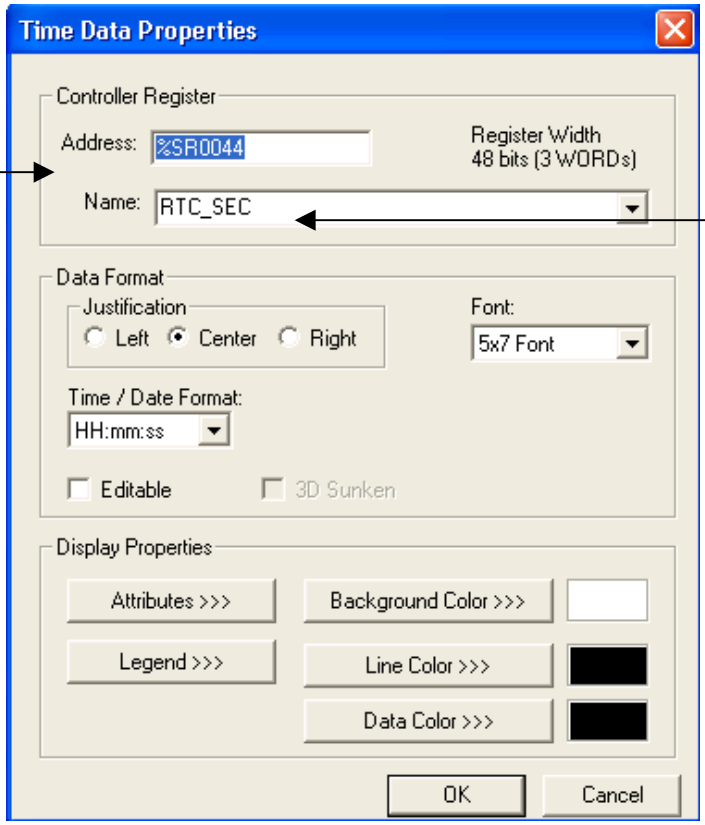



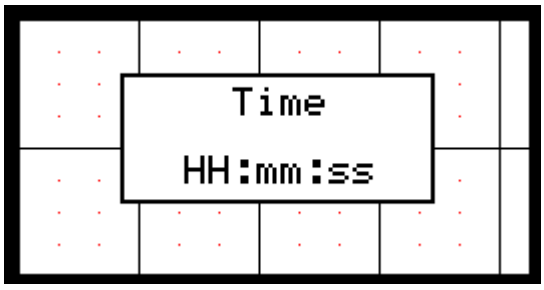
## *Time Data*

To insert a time data display click on the icon  and click it into the screen. Double click the box to edit the properties.

The system time is stored in consecutive SR registers from %SR44



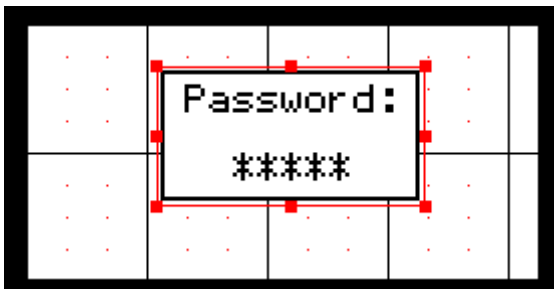

It may simpler to select the RTC registers by choosing their SR register name

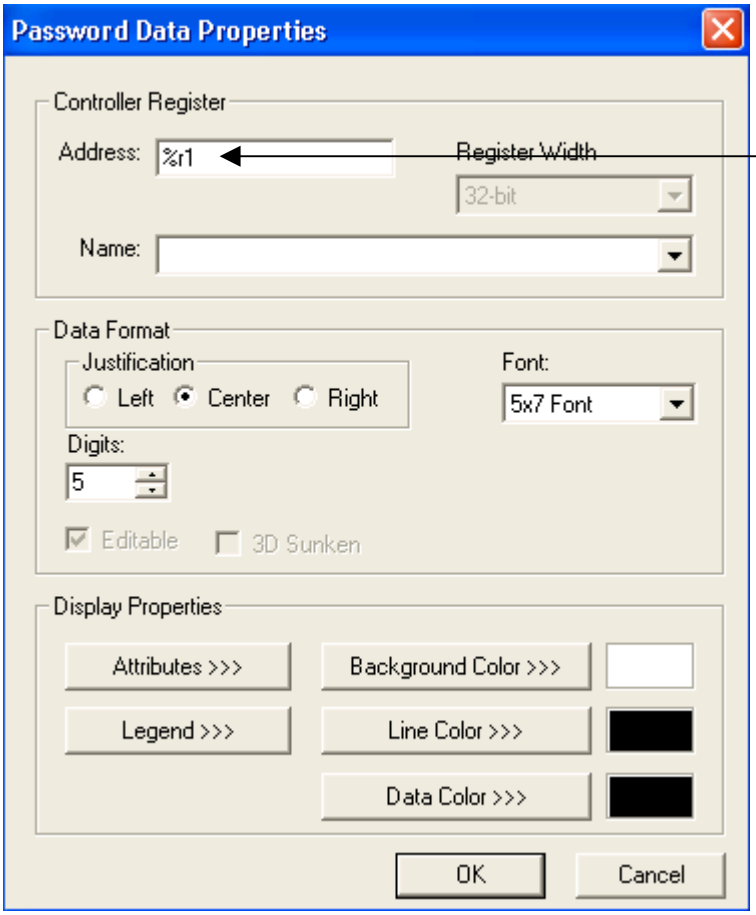


## *Password*

To insert a Password display, click on the icon and click it into the screen.

The password box is similar to the numeric data display but the numbers are hidden.



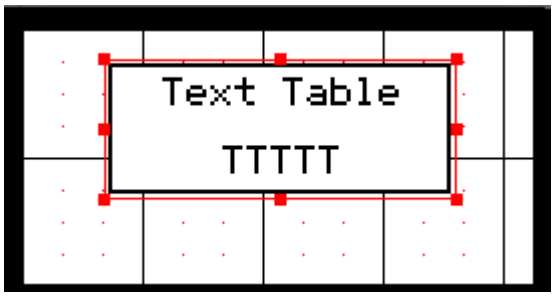

Address where the password will be entered.

Simple password logic uses a Compare Function block. When the Password Number entered is equal to the stored value an additional control condition can be unlocked.

## *Text Table data*

To insert a Text Table display, click on the icon and click it into the screen.

The text table can be used to display text messages instead of numeric data for a given register.




A message table can be selected for a word, byte or a bit.

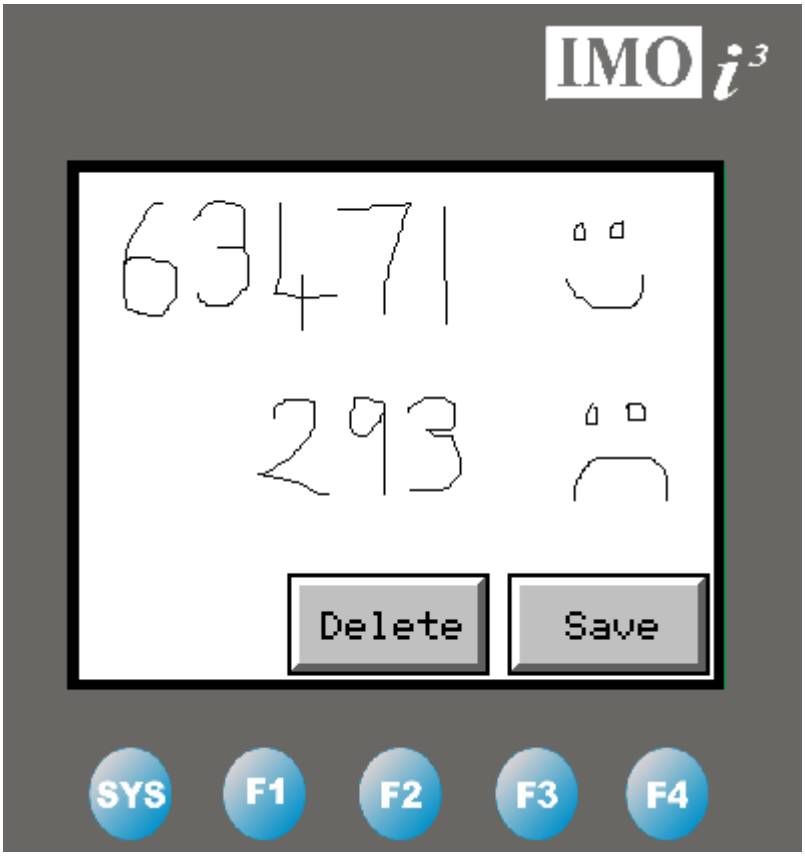
There can be up to 255 tables and one can be used more than once.

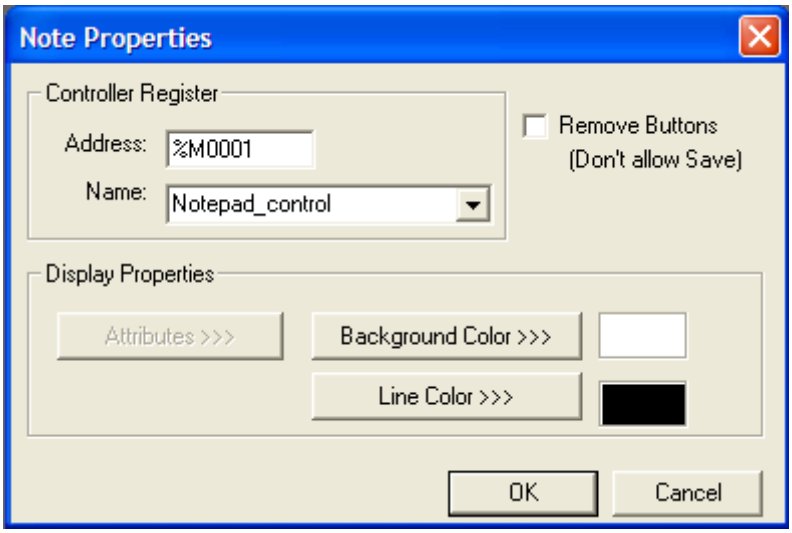
The number of digits to display must match the maximum length of message to display.

Select a value and enter a message to correspond to it. Now the message will be displayed instead of the numeric value.

## *Notepad*

This is function is only for touch-screen models such as the i3B12Y range. It provides the operator with a screen upon which they can write or draw a simple diagram. This data can then be stored into the HMI memory for recall at a later date. A useful feature should an operator find themselves without a pen and in need of writing down batch numbers.

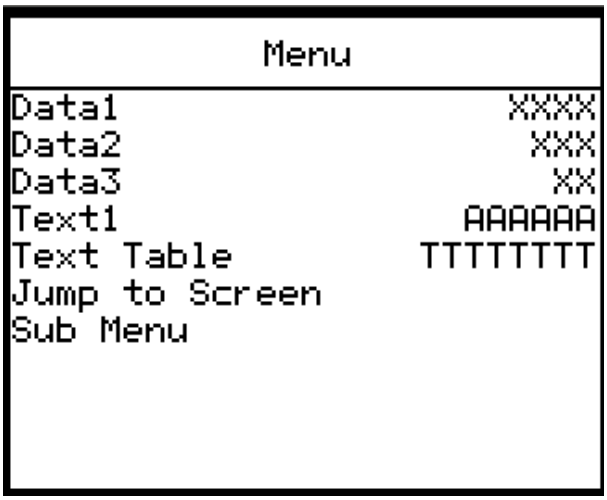


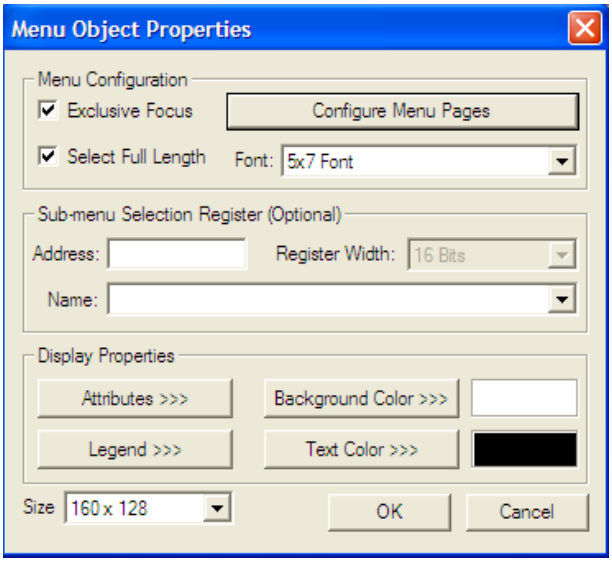

The Controller Register enables or disables the Note Object. The Remove Buttons checkbox removes the ‘Delete’ and ‘Save’ buttons from the Note object making sure it is erased every time the Note screen is exited.

## *Menu Item*

A Menu Item allows many settings and screens to be configured through one single object.



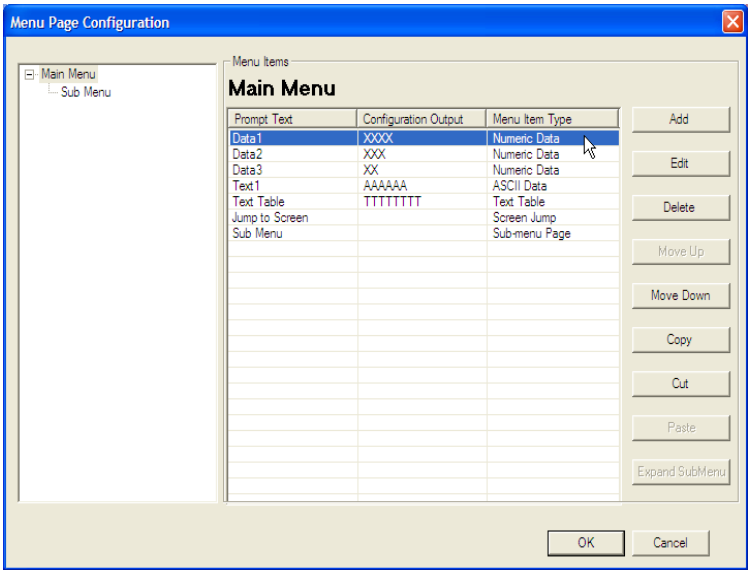

Inside the Menu Item it is possible to easily configure groups of Numeric Data Items, Text Tables, ASCII text Items, Passwords, Screen Jumps and also Sub-menu items.
The Sub-Menus allow more pages of configurable data to be displayed, allowing an operator to make many changes from within a single screen.
The menu object is navigated via the soft keys or touch screen.




If the Exclusive Focus is selected then no other objects on the screen can be activated.

Clicking on the Configure Menu Pages button opens the window below.

The 'Size' of the Menu must be selected when using on a Touch-Screen ***i***$^3$.



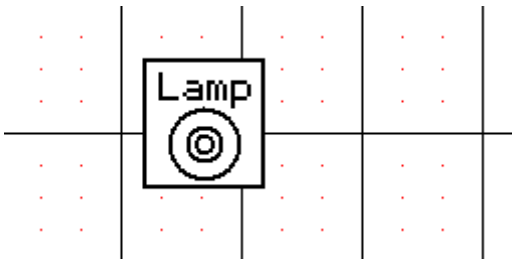

Menu Items can be added, then, configured in a similar way to how they are outside of a Menu Object by specifying the Control Register, Data Format, and Display Properties. The only difference is that 'Prompt Text' is used instead of a Legend.

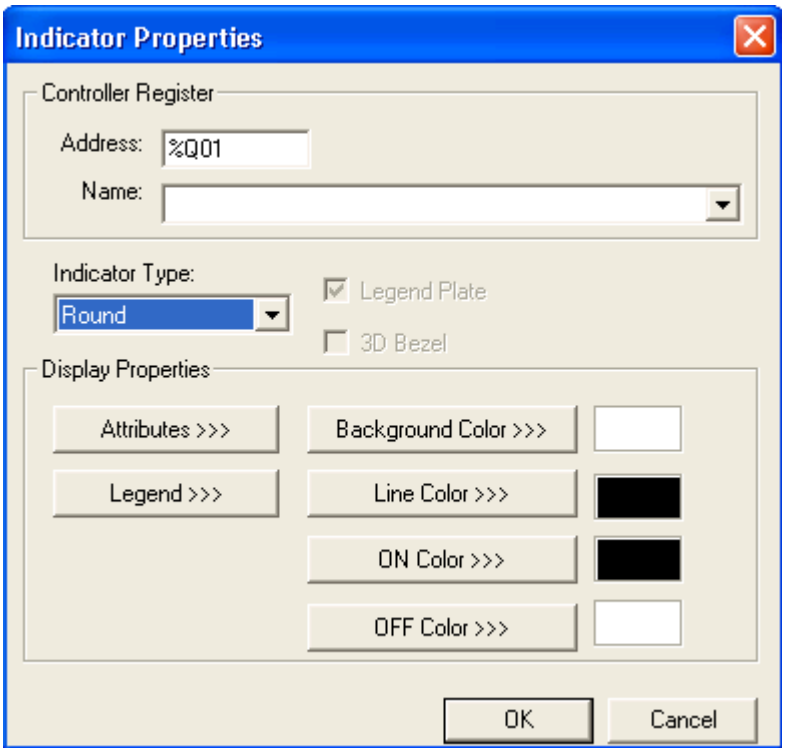
Sub-Menus can also be added, to partition the data into different areas as required.

## *Indicator Lamp*

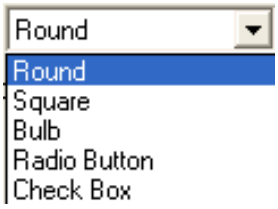
To insert an indicator lamp, click on the icon  and click it into the screen.

The indicator lamp can only be assigned to a bit (Q, M, S or T).



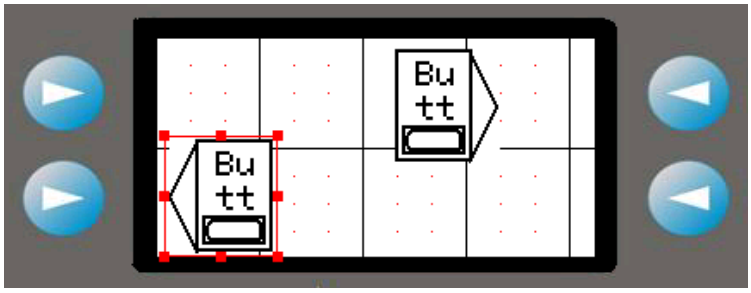

There are several different options for display of lamp.



## *Button*

There are four buttons that can be programmed through the screen editor, the remaining keys are programmed in the ladder editor. To insert a button, click on the icon  and click it into the screen.

The button will then allocate itself to the nearest key next on the side of the screen. Only one button can be assigned to a single key per screen.



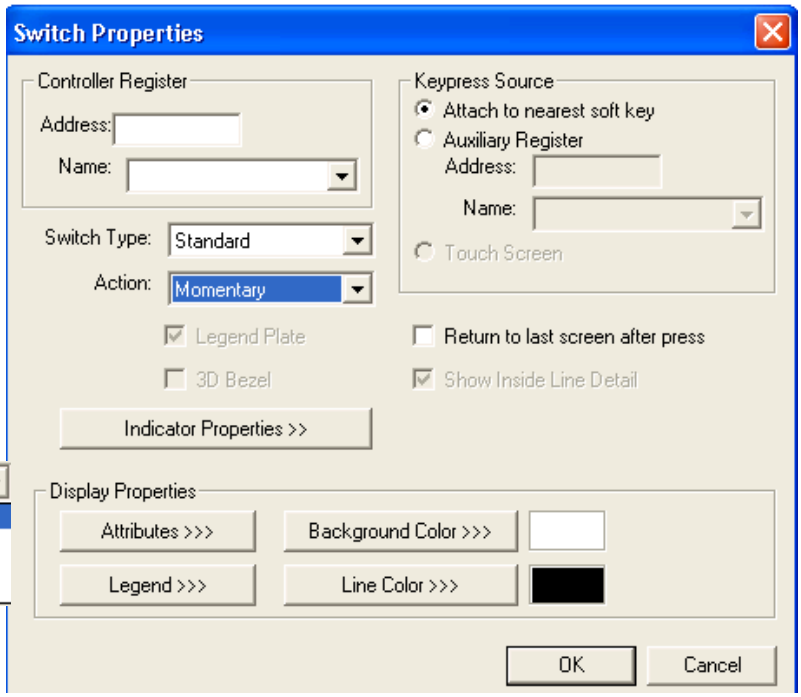

The buttons are binary and can only be assigned to a bit

Switch type

Standard
Round
Square
Rocker

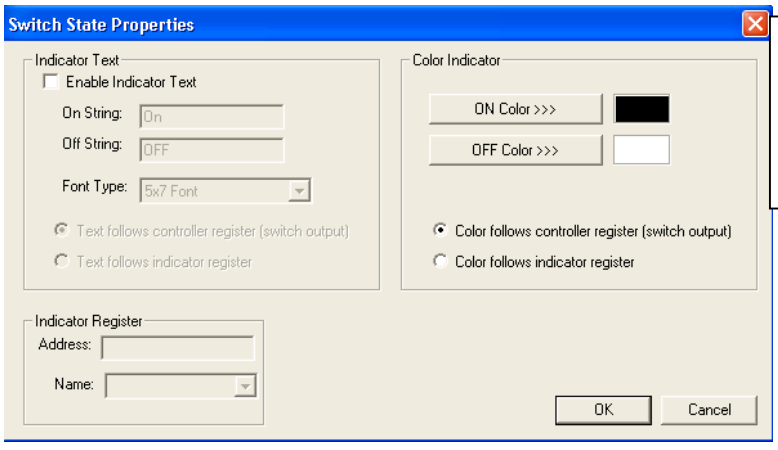
Action

Momentary
ON
OFF
Toggle




The Keypress source defaults at the nearest softkey (4 keys either side of the screen.

However they can be set to another address in the ***i***$^3$.

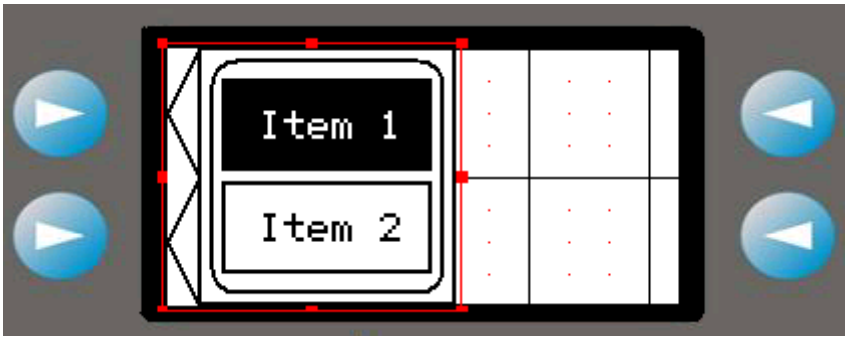



We can also edit the indicator properties

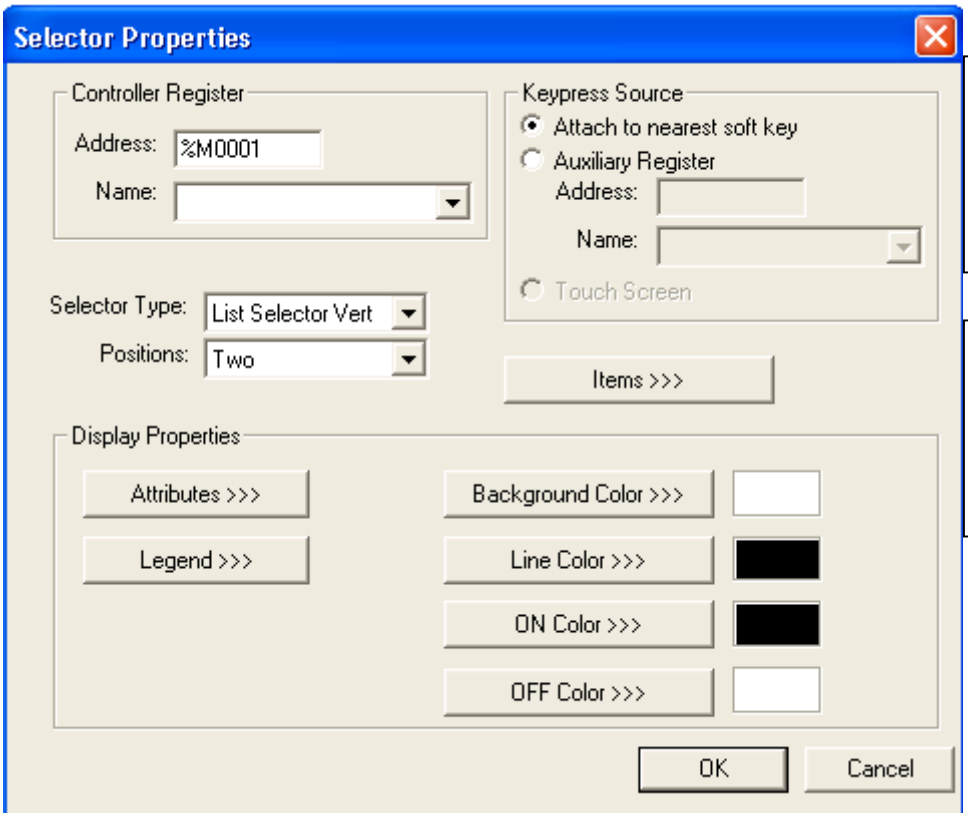
## *Selector Switch*

To insert a selector switch, click on the icon  and click it into the screen.

It will then select the nearest buttons to it on either side of the screen. With the selector switch the maximum the ***i***$^3$ can have is two positions.




A selector switch will be either one of two states.

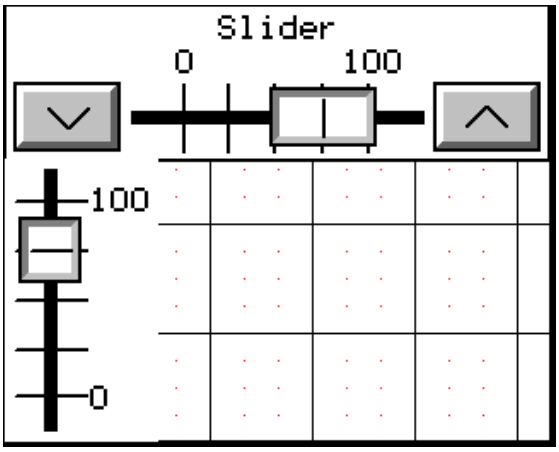


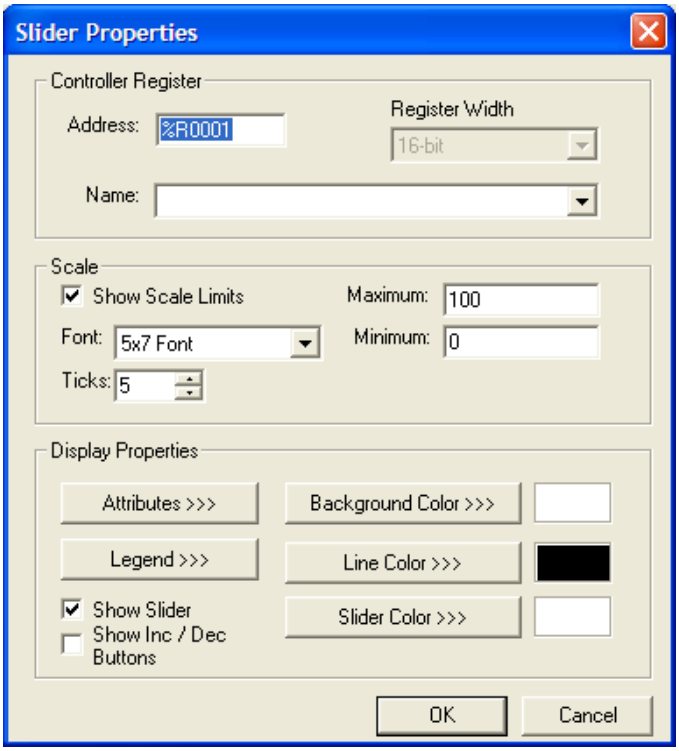

The address can be a bit or a register.

The items can be given meaningful names

## *Slider*

This is an option only for touch-screen models such as the i3B12Y range. It is a quick way to change a value inside a 16-bit register. The object can be configured with or without extra buttons for fine control, and automatically changes it orientation from horizontal to vertical depending how it is sized – similar to the Bar Chart Item.



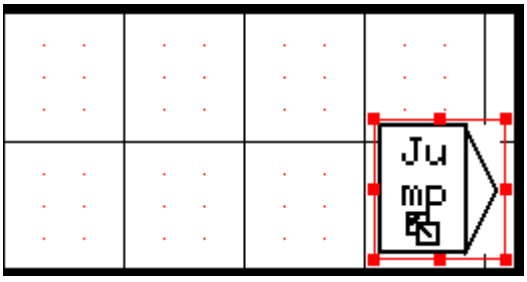




The Slider has configurable scale limits. It also has two checkboxes that allow the graphic of the slider to be turned off along with the Inc/Dec buttons.

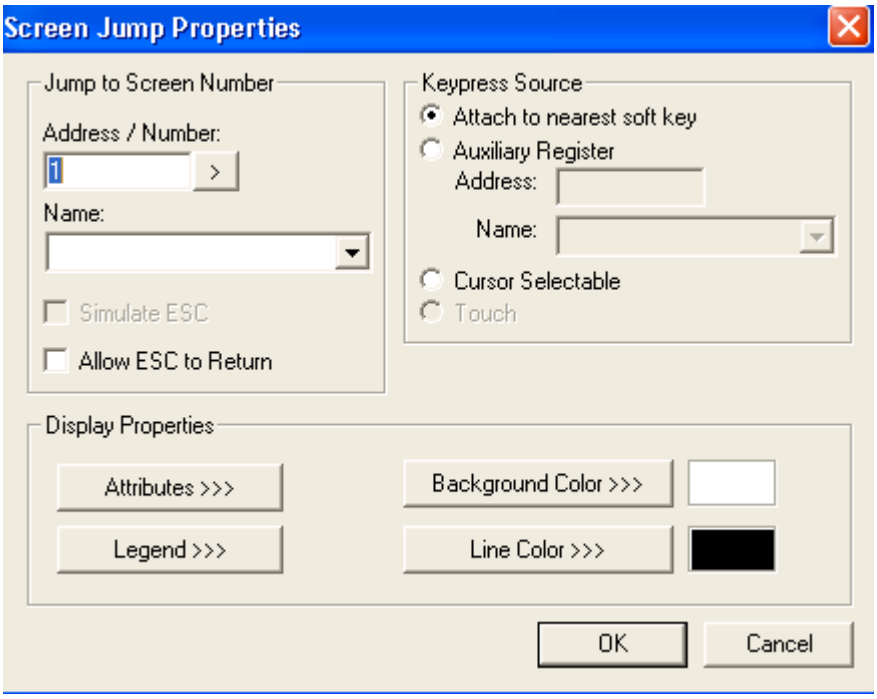
## *Screen Jump*

Screen jumps are allocated like buttons but are for jumping between screens in a menu fashion. To insert a screen jump, click on the icon  and click it into the screen.

The jump button will allocate itself to the nearest key at either side.




Select what screen number to jump to and the display properties of the button.



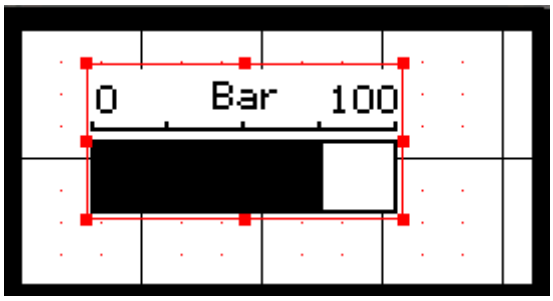

The Keypress source defaults at the nearest softkey (4 keys either side of the screen.

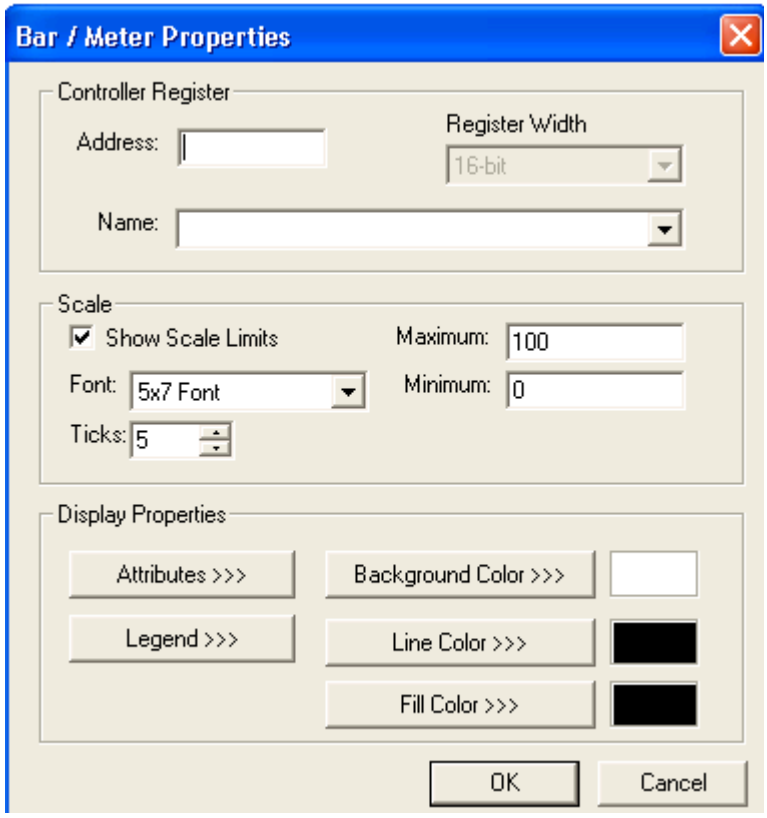
However they can be set to another address in the ***i***$^3$ or can even be selected using the cursor and arrow keys

## *Bar Graph*

To insert a bar graph, click on the icon  and click it into the screen.

To graphically display a register value on the screen. Click on the edge of the box and drag to make the bar graph bigger.



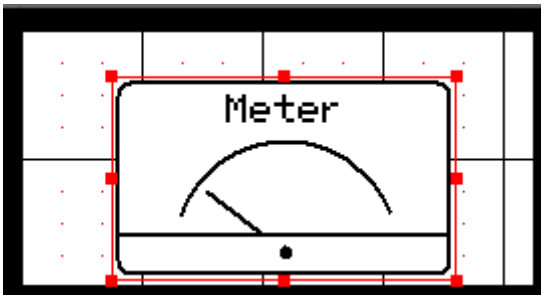

The bar graphs must be a word.

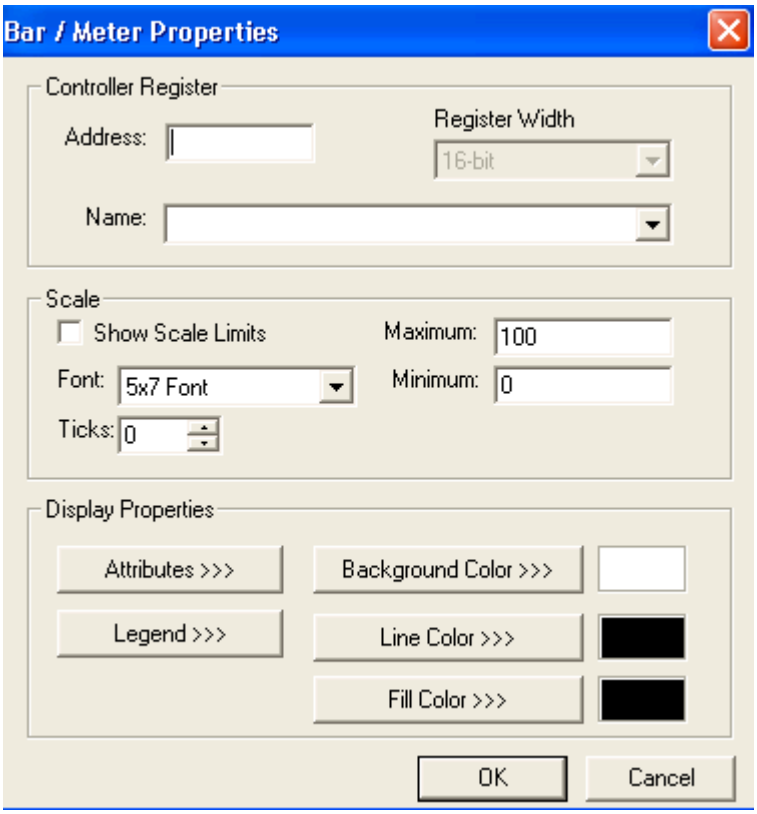
The bar graph has to be scaled and this can be shown on the screen.

## *Meter*

To insert a meter graph, click on the icon  and click it into the screen.

Another option to the bar graph is o display the data as a meter.



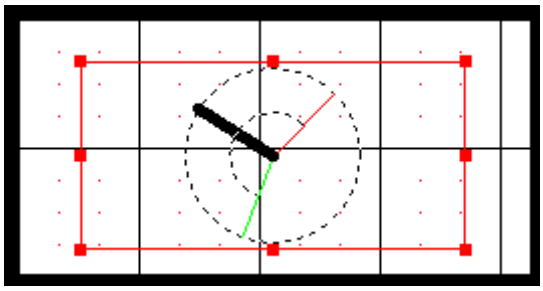



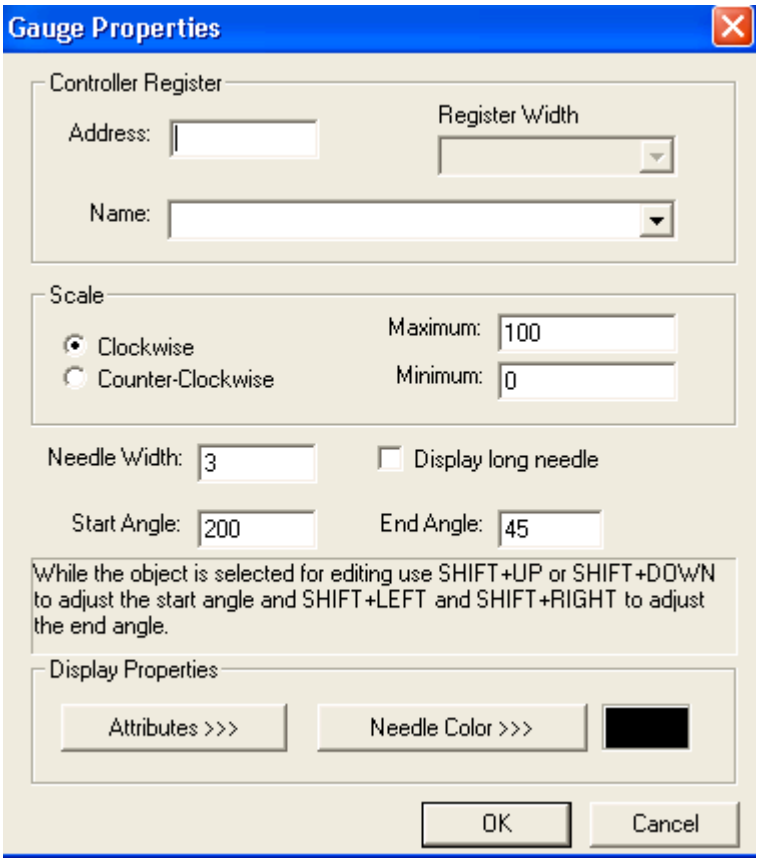

The details are the same as with the bar graph.

## *Gauge*

To insert a gauge graph, click on the icon  and click it into the screen.

The Gauge is more complex than a bar / meter. This gauge is an automotive style gauge and can be placed over bitmaps.





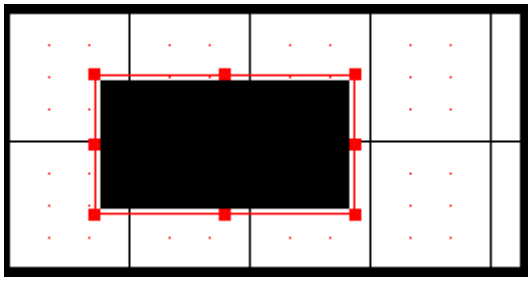

Select starting and ending angle and the needle will rotate within the set scale.

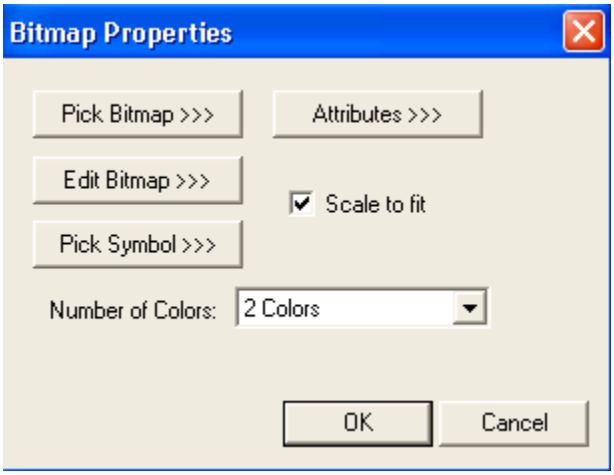
## *Static Bitmap*

A bitmap can be used as a screen back drop, where a company logo can be inserted.

To insert a static bitmap, click on the icon  and click it into the screen.

The ***i***$^3$ has a mono screen, therefore the bitmap must comply to these restrictions.



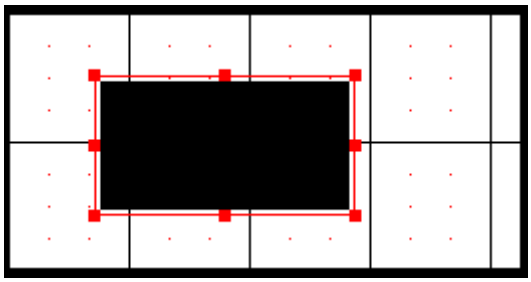


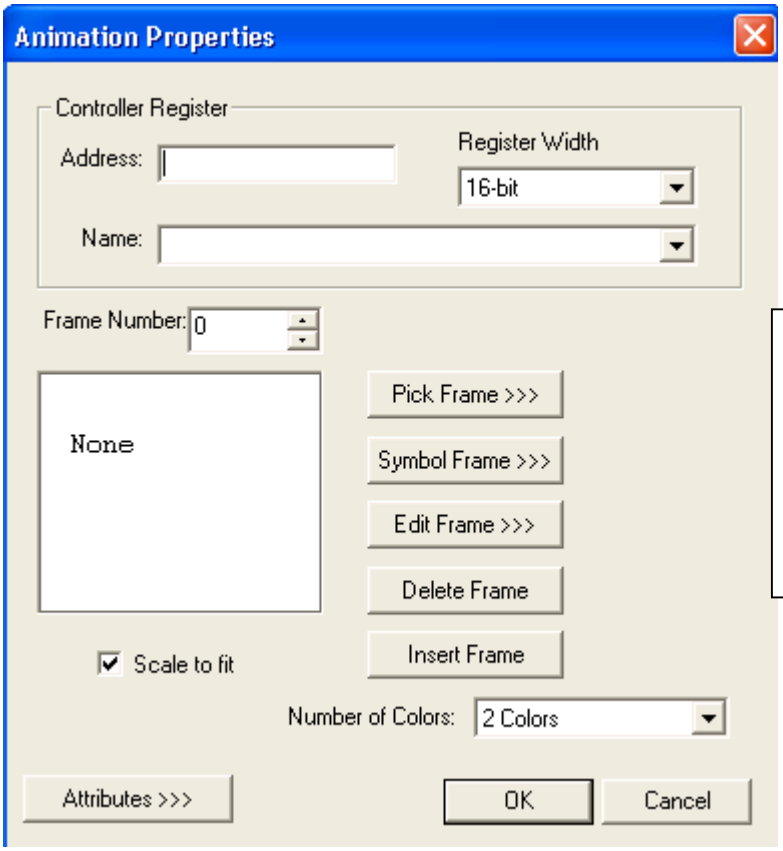

Bitmaps can be created in simple packages like MS Paint.

## *Animation*

The animation displays a series of bitmaps depending on the value of a register, double word, word, byte or bit.

To insert an animation, click on the icon  and click it into the screen.



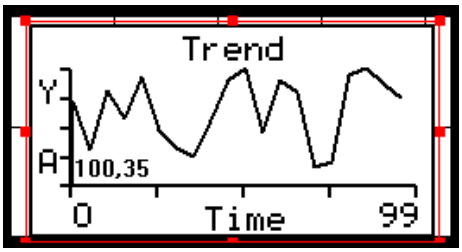


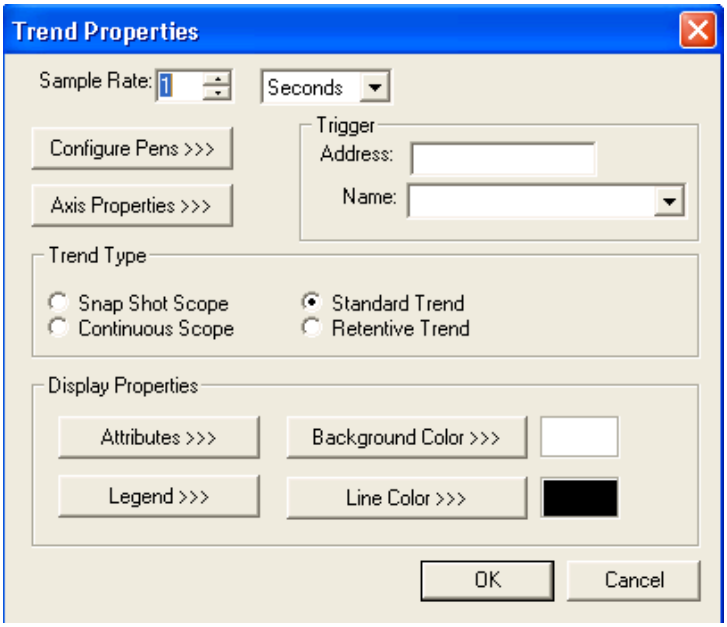

Select the bitmaps for the frames to match the value in the registers, i.e. a bit will have two frames.

## *Trend*

The data trend tracks data over a period of time. To insert a trend, click on the icon  and click it into the screen.

A data trend can track up to four registers over a set period of time.

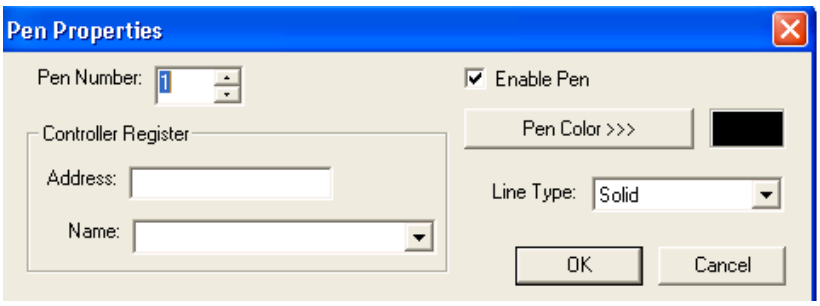


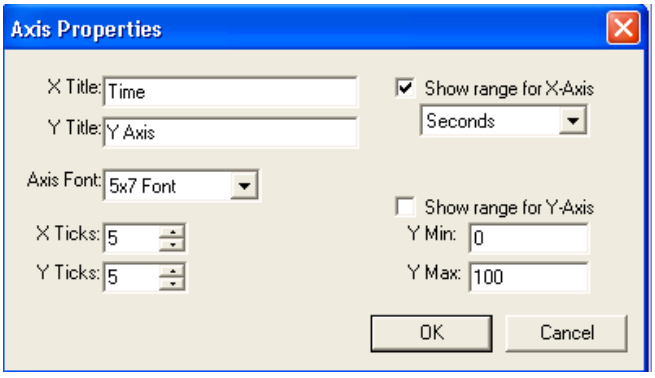

The sample can be in seconds, minutes or hours

The trigger address is required to activate the trend.

The trend can be 1 of 4 different types, see the help file for detailed information



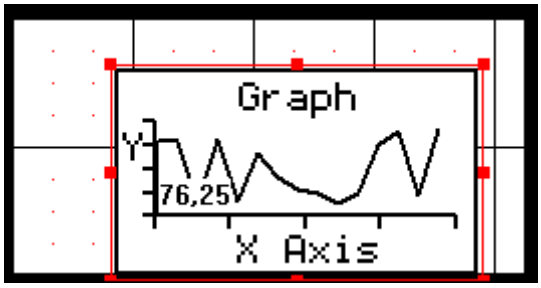

Up to 4 pens per trend can be edited. i.e. 4 data registers.

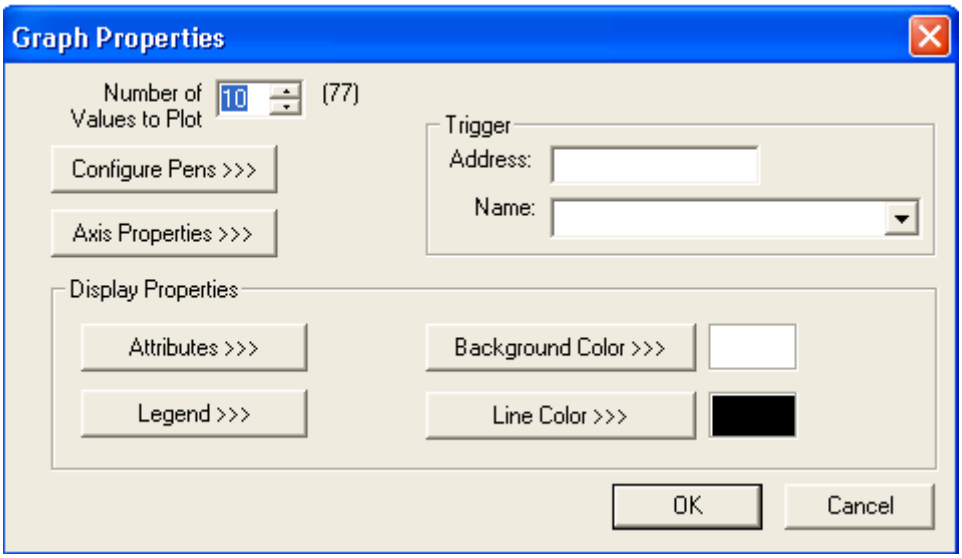



The axis titles can be edited to something more meaningful, and the scale can be adjusted

## *X – Y Data Graph*

To insert an X – Y Graph, click on the icon  and click it into the screen. The X-Y graph represents variation of a variable in comparison to variations in one or more variables.



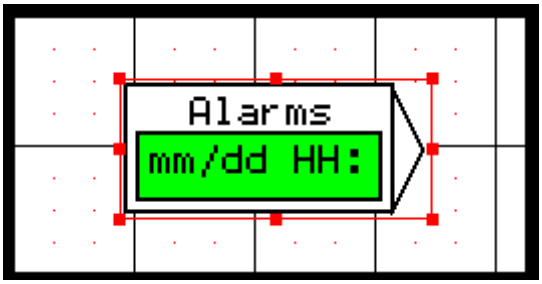




The trigger address is required to reset and refresh the plotting process.

## *Alarms*

To insert an Alarm Log, click on the icon  and place it into the screen. There are two types of alarm: Summary and History. Summary only displays the alarm when it is currently active and History logs the alarm. There are two steps to setting up the alarm, first the button needs to be set up then the log itself.

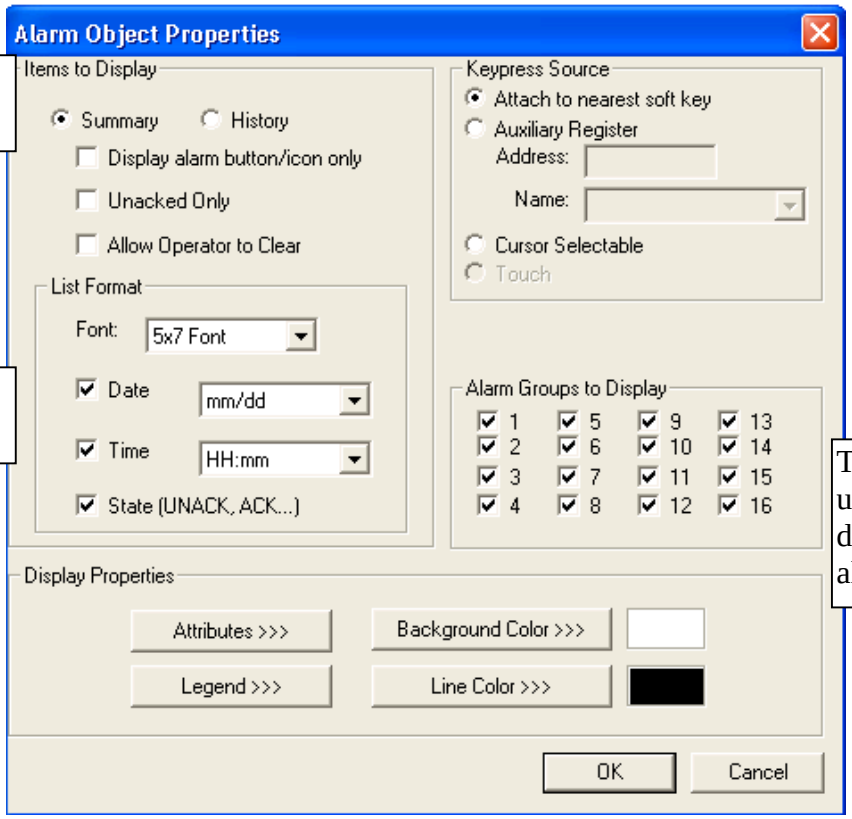
The alarm will display a message and time stamp it for when it occurred.




Type of alarm log

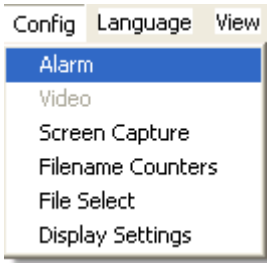
Details to display

There can be up to 16 different alarm groups



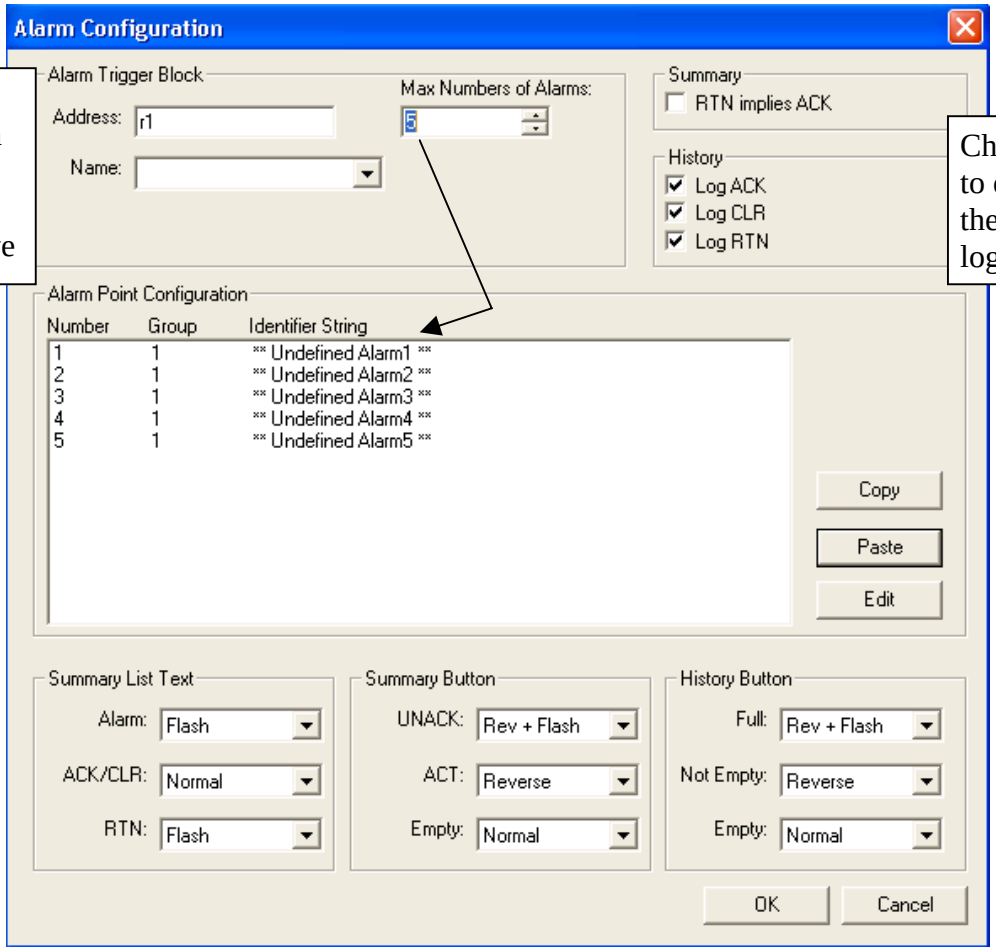
## Configure the Alarm Log

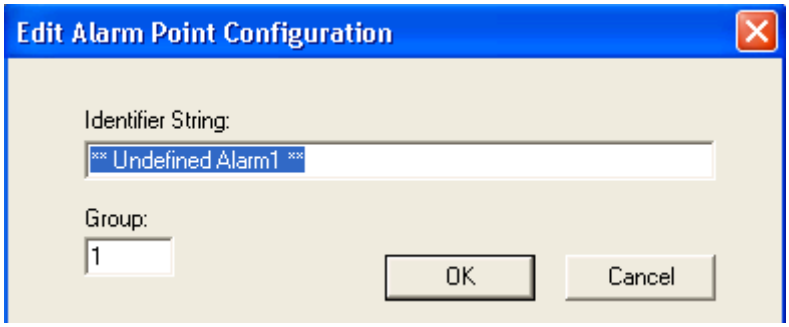
Select Alarm from the Config Menu drop down, to open the Alarm Log editor.




The alarm trigger can be 1 register or consecutive

Choose what to display in the History log



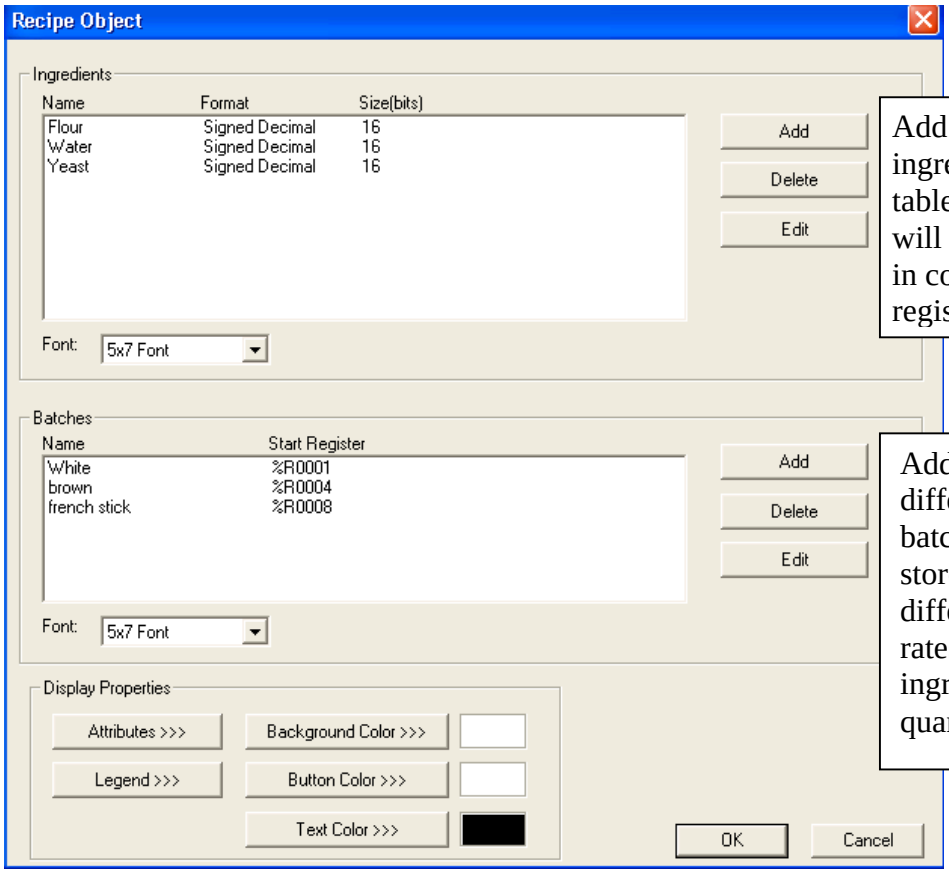




Alarm message to display and group related to.

## *Recipe Editor*

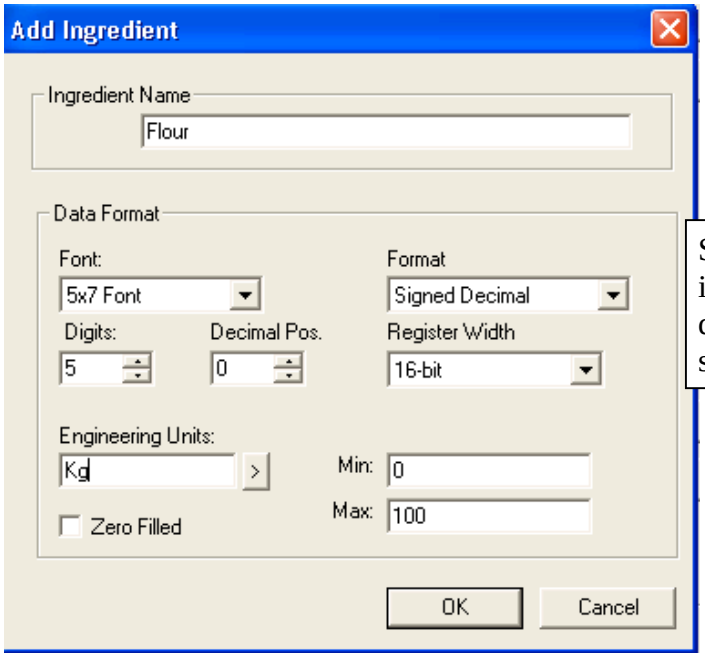
To insert a Recipe Editor Object, click the icon on to the screen.

A recipe can be created to store the values of variables for different batch quantities.

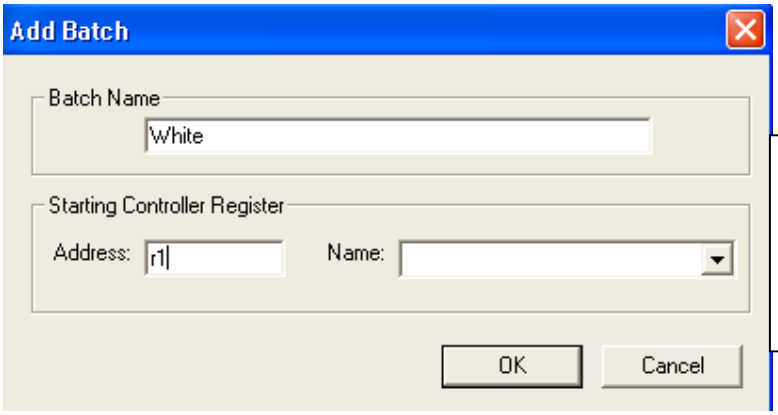




Add the ingredients table. These will be stored in consecutive registers.

Add the different batches, to store the different rates of ingredient quantities




Set up the ingredient display and scales.




Add the batch properties and storage register.

This recipe function can be used in conjunction with move function blocks, to move recipe data from one location to another.

IMO Precision Controls Limited
1000 North Circular Road
Staples Corner, London
NW2 7JP United Kingdom
Tel: +44 (0)20 8452 6444
Fax: +44 (0)20 8450 2274
Email: imo@imopc.com
Web: www.imopc.com

IMO Jeambrun Automation SAS
165 Rue Jean Jaures,
94700 Maisons Alfort
Paris, France
Tel: +33 (0)1 45 13 47 05
Fax: +33 (0)1 45 13 47 37
Email: info@imopc.fr
Web: www.imopc.fr

IMO Deutschland
Für weitere Einzelheiten zu IMO Agenten und Distributoren in Ihrer Nähe schreiben. Sie bitte ein E-mail an folgende Adresse:
imo@imopc.com

IMO Italia
Viale A. Volta 127/a
50131 Firenze, Italia
Tel: +39 800 783281
Fax: +39 800 783282
Email: info@imopc.it
Web: www.imopc.it

IMO Canada
Unit 10, Whitmore Road
Woodbridge, Ontario.
L4L 8G4 Cananda
Tel: +1 905 265 9844
Fax: +1 905 265 1749
Email: imocanada@imopc.com