



LT0345

**SmartConfigLite1.6.3.0  
User Manual  
(Lite Version)**

Thursday, 2 April  
2009

## **SmartConfigLite1.6.3.0**

**© 2009 ... Tyco Safety Products, Christchurch**

SmartConfigLite1.6.3.0 is a Windows program for editing, viewing, and/or printing the configuration database for several products of Tyco Safety Products, Christchurch.

Tyco Safety Products takes no responsibility for the use of this program. All installed systems should be fully tested to ensure that they meet the stated requirements of the customer and all relevant regulations, standards, and accepted trade practices.

# Table of Contents

<b>Part I Getting Started</b>	<b>5</b>
<b>Part II Menu Commands</b>	<b>6</b>
1 File .....	6
2 Edit .....	8
3 Select .....	9
4 Profile .....	9
5 View .....	9
6 Check .....	11
7 Window .....	11
8 Licence .....	11
9 Tools .....	12
10 Help .....	12
<b>Part III General Reference</b>	<b>13</b>
1 General Editing Instructions .....	13
2 Template Files .....	14
3 Licensing .....	15
4 Terminal Window .....	16
<b>Part IV FP1600 / Omega 64 System</b>	<b>19</b>
1 FP1600 Overview .....	19
2 FP1600 Version Differences .....	21
3 FP1600 Getting Started .....	21
4 FP1600 Tables .....	22
Zones .....	22
Zone Map Ancil .....	23
Zone Map Groups .....	23
Zone Map Variables .....	24
Group Config .....	25
Group Map Ancil .....	25
Group Map Variables .....	26
Ancil Config .....	26
Ancil Map Variables .....	27
Variables Notes .....	28
System Config .....	28
Change Log .....	29
Comments .....	29
5 Download / Upload .....	29
Communicating with the FP1600 Panel .....	29

6 FP1600 Reference .....	30
Checks .....	30
<b>Part V IO-Net</b>	<b>31</b>
1 Opening Files .....	31
2 Parameters .....	31
Address .....	32
Max Stations .....	32
Panel Number .....	32
TX Zone Data .....	33
RZDU Protocol .....	33
Allow reduced RAM .....	34
Off-line RAM Assumptions .....	35
3 Output Logic .....	35
General Format .....	35
Operators .....	37
Order of evaluation .....	38
Operand Types .....	39
IO-Net Inputs .....	39
IO-Net Outputs .....	40
FIP Zone Data .....	40
Variables .....	40
New Alarm Function .....	41
System Conditions .....	43
Constants .....	45
Timers .....	45
4 Change Log .....	50
5 Notes .....	50
6 Network Profiles .....	50
7 Example Programs .....	51
Duplicating Non-Programmed mode .....	51
Passing Data onto the IO-Net network .....	51
Accessing Zones .....	51
8 Checking a Program .....	52
9 Programming an IO-Net Controller .....	54
10 Extracting a program from an IO-Net Controller .....	55
<b>Part VI Whats New</b>	<b>57</b>
1 Version 1.6.2 .....	57
2 Version 1.6.1 .....	57
3 Version 1.6 .....	57
4 Version 1.4 .....	57
5 Version 1.3 .....	59
<b>Index</b>	<b>60</b>

# 1 Getting Started

Start with [File | New](#)<sup>[6]</sup>. Select the type of system you want to configure. This will give you a blank database to configure. Alternatively open an existing database with [File | Open](#)<sup>[6]</sup>.

Select the table or form to edit from the listbox on the left. Tables with multiple entities (e.g. Zones) are normally viewed in [table view](#)<sup>[13]</sup>, while tables with only one entity (e.g. System) are normally viewed in [form view](#)<sup>[14]</sup>.

Row and column headings are in light grey. Cells you can edit are generally in white. Cells with calculated useful information are in light yellow. Cells that are not applicable are in dark grey. (These colours may vary depending on your Windows colour scheme.) Other cells are generally white, but various colours may be used by SmartConfig to highlight less frequent settings.

There are four basic types of cells -

1. Multichoice. These are displayed with a blue triangle on the right. Click the triangle to get the choices. In form view they may be displayed as a radio group or as a drop-down edit box.
2. Yes / No - displayed as a box with or without a tick
3. Integer
4. String

Refer to [General Editing Instructions](#)<sup>[13]</sup> for additional general information.

Check the configuration. [Check | Check tables](#)<sup>[11]</sup>

Save the completed configuration. [File | Save](#)<sup>[6]</sup>.

Refer also to the Getting Started or Overview section for the particular system type you are using.

## 2 Menu Commands

This shows the menu commands available in SmartConfig, and describes their function.

### 2.1 File

#### New

Creates a new database for the selected system type, and initialises it from the selected template file.

SmartConfig may be supplied with multiple template files for some panel types, and/or you can create your own templates. (See [Template Files](#)<sup>[14]</sup>)

#### Open

Opens an existing file saved in SmartConfig format. Additionally, for some system types, you may be able to open a file in the file format able to be directly uploaded / downloaded to the panel (e.g. s28 format.)

Whatever extension the file has, the file content will be examined so it will be opened and read correctly.

#### Close

Closes the current document window. A prompt is displayed if it has been modified, allowing you to save the file first, or to cancel the Close operation.

#### Compress Files When Saving

If ticked, then all saving of database files will be done in a compressed format, which will make files 10 - 20 times smaller on disk. Make sure this option is unticked if a file needs to be in a text format for reading by third party products e.g. CCU-Net configuration. Versions of SmartConfig earlier than 1.3.0.0 can open only uncompressed files. This setting will be remembered across SmartConfig sessions.

#### Save

Saves the file for the current document window in SmartConfig format.

#### Save As

As for save, but allows saving of the file with a new name.

Note that you have the option of saving it as a template. See [Template Files](#)<sup>[14]</sup>.

#### Select Default Data File Location

This command enables you to set the default location for loading and saving data files. The folder you select is remembered across SmartConfig sessions.

The default folder for "Save As" of a file not created from new will be the folder the file was loaded from.

If no folder has ever been selected with this command then when loading a file, or saving a file created from new, the default folder will be the "SmartConfig" folder, the parent folder of the version folder that SmartConfig is installed in.

When a second or subsequent file is opened, the dialog box will start in the folder where the last file was loaded from (even if this is not the set default data file location). However if the default data file location is changed, then when another file is opened, the dialog box will start in the new location.

## Export in Panel Format

This command is only available for some system types. It exports the current document window to a file in the format required for export to the panel. See also [Download to / Upload from Panel](#)<sup>[7]</sup>, which gives a more direct method of sending the database directly to the panel.

## Print Table

Prints the currently visible table for the current document window.

## Print All

Prints all data for the current document window (usually many pages).

## PDF

Prints all data for the current document window into an Adobe Acrobat .PDF file (to read use Acrobat Reader V3.x and above). When selected, a dialog box will open to allow the file name and location to be chosen. Use this feature to create a document for later printing, or as a record of the site's database that can be read without SmartConfig.

## Print Labels

Prints zone labels for the current document window. The size and layout suits the LED board(s) for the current product.

## Compare

Compares two databases and displays (and optionally prints) the differences. To use this command you must open the two databases you want to compare, and no other databases may be open. The newer database should be on top (have focus). The compare function has some intelligence (for instance added points do not have their default parameters listed).

## Terminal Window

Starts a terminal emulator to connect to the diagnostic terminal of a panel. You need to select the panel type so that the correct baud rate is automatically selected. The terminal can connect directly to the panel over an RS232 connection (using a COM port on your PC), or indirectly over a Telnet (TCP/IP) connection using, for example, the Internet, a corporate WAN, etc. See [Terminal Window](#)<sup>[16]</sup>.

## Download to / Upload from Panel

Sends the database from SmartConfig to the Panel, or retrieves a database from the Panel into SmartConfig. Refer to more specific instructions for the system type you are using. Look up "Download to / Upload from Panel" in the index to get a list of system types to select from.

NOTE With some panels this command will do the same as "Terminal Window", and you will need to then click "Send Database" or "Capture Database" to start the download or upload.

## Upload New From Panel

Use this command to read a database from a panel. (Alternatively you could create a new empty file from the correct template and then select "Download to / Upload from Panel".)

## Recently Used Files

The last 4 files opened or saved by SmartConfig will be listed here. When the mouse cursor is placed over each file, its full file name and location will be shown in the status bar at the bottom of the SmartConfig window. Click on a file name to open it.

## Exit

Exits the program.

## 2.2 Edit

### Undo

Undoes the most recent change (e.g. typing in a single cell, multiple cells changed with cut, delete, paste, fill down, or fill series, or a change to a logic equation).

Note - consequential changes may not be "undone". For instance if you change one or more point type(s), and then "undo" the changes, the initial point types will be restored, however any individual point parameters will be lost as these are always automatically reset when point types are changed.

### Copy

Copies the selected cells to the Windows clipboard. (Note - select the cells by dragging the mouse, or by clicking in one corner of a rectangular area, and Shift-clicking in the opposite corner.)

The clipboard may be pasted into SmartConfig, or other Windows applications such as Excel. To paste into a table in Word, you must either first create a table in Word, and select exactly the right number of rows and columns before pasting, or just paste as text, then select what you just pasted and "convert text to table". Alternatively paste into Excel first and then copy and paste from Excel to Word.

### Copy with Header Row & Fixed Cols

Copies the selected cells along with the fixed (light grey) cells above and to the left of the selected area. (There is no other way of selecting these fixed cells). The selected cells do not need to be adjacent to the header row and fixed columns.

This command copies, for example, the selected information in the point table along with the relevant column headings and the responder and point numbers, in a way suitable for pasting into another application.

### Paste

If only one cell is selected, this pastes the clipboard into a rectangular area with the top left corner in the current cell.

If an area is selected, its width in cells must be the same as that of the clipboard. Its height must be at least that of the clipboard. The clipboard is pasted into the selected area as many times as will fit vertically. (One example of this being useful would be to select the same point type for every 4th point, with all the intervening points being unused. Select the required point type for point 1. Select points 1 to 4 and "Copy". Select points 5 to 250 and "Paste". It may then be useful to "View used rows".)

The source of the information can be SmartConfig or a spreadsheet such as Excel. You can use this feature to import point text, zone names, etc., from other sources.

### Cut

Copies the selected cell(s) to the clipboard, and then clears the cell(s).

### Delete

Clears the selected cells.

### Fill Down

Copies the contents of the top cell(s) of the selected area into all other rows of the selected area. You can use this to quickly set a number of adjacent cells to the same value - if necessary using filter or sort first to group the cells you want to change together.

### Fill Series



Fills the selected cells with a series with a uniform increment equal to the difference between the top selected cell and the second top selected cell. If this increment is not a number between 1 and 10, 1 is assumed. You could use this for example to map a number of sequential circuits to sequential zones.

If the top selected cell contains alphabetic parts as well as a numeric part, the numeric part will be incremented. For example "Zone 1 text" will result in "Zone 2 text", "Zone 3 text", and so on.

### **Edit Cell**

Allows you to edit a grid cell. If the cell contains a number or text, you can edit it in the edit box at the top without destroying the cell's initial contents. If the cell is multichoice, you can select from the choices in a popup menu. If it is a yes/no choice, it will flip.

### **Select All**

Selects all cells if a table is displayed, or selects all text if the output logic editor is displayed.

## **2.3 Select**

The select menu provides an alternative method of selecting a table to edit. It has the same function as the listbox on the left of the screen. The advantage of the Select menu is that it does not need a mouse - type Alt+S, then the underlined letter.

The names of the tables vary according to the type of panel currently being edited. Refer to the panel specific sections for the table details.

## **2.4 Profile**

This gives a selection of "Profiles" to edit. Profiles contain tabular data like ordinary tables, however the data is usually predefined and does not require editing by the user.

Generally profiles provide named groups of predefined settings which can be selected by name in other (normal) tables. For example the required thresholds and filtering settings for a "Type A" heat detector.

You can create your own profiles if you wish, just add extra rows to a profile table. It may be useful to use SmartConfig's copy and paste facility to make a new row which is the same as an existing row, and then changing the data you want to change. Ensure that each row of the table has a unique name.

WARNING Changing the settings in profiles may result in the system not being Standards compliant.

## **2.5 View**

### **As Table**

Refer to [table view](#) <sup>13</sup>.

### **As Form**

Refer to [form view](#) <sup>14</sup>.

### **Show All Rows/Unsort**

Makes all rows of the current table visible (e.g., after a "Filter", or "Show Used Rows"). Also returns the sort order to the default order. If a table allows rows to be "collapsed" and "expanded", Show All Rows will expand all such rows.

### **Show Used Rows**

Makes only the currently used rows of the active table visible. NOTE - the selection of rows displayed may not necessarily continue to be "used rows" after any data is changed.

### **Collapse All**

On a table that uses + and - icons to show (expand) or hide (collapse) sub-items (e.g., MX1 Points table), this will collapse all items, causing only rows that are not sub-items to be displayed.

### **Go to Related Table**

Goes to another table showing data relating to the current cell, positioning this data at the top of the view.

### **Back**

Goes back to the previous table.

### **Sort Ascending**

Sorts the table so that the column containing the current cell is in ascending order. (For integer columns the order is numeric. For other columns the order is alphabetical). NOTE - the order of rows displayed may not necessarily continue to be sorted after any data is changed.

This command is also available by right clicking in the heading of the desired column.

You can sort according to the contents of more than one column, by sorting on the least significant column(s) first, then finally the most significant column.

### **Sort Descending**

As above but in the opposite order.

### **Filter...**

Allows you to display only, or all but, the rows where the column value matches a value which you enter. (By default the value is that of the current cell.) NOTE - the selection of rows displayed may not necessarily continue to obey the filter after any data is changed.

This command is also available by right clicking in the heading of the desired column.

### **Move Column Left**

Moves the column containing the current cell, one position to the left. Note that currently this setting is lost when you exit the program.

### **Move Column Right**

As above but moves the column right.

### **Find**

Displays a dialog box for you to enter some text to find in the current column of a table displayed in table or form view, or in the Output Logic Editor if it is displayed.

If a table or form is being displayed, the search starts downwards in the current column. (In form view the current field is searched while scanning forwards through the remaining records). If the bottom of that column is reached, you will be given the choice of restarting at the top of the same column, searching any remaining columns, or cancelling. If you choose to search remaining columns, that choice will stay in effect until a new "Find" is started.

For both table view and form view, any rows which are not visible due to a Filter will not be searched.

This command is also available by right clicking in the heading of the desired column.

### **Find Next**

Finds the next occurrence of the text.

## 2.6 Check

### Check O/P Logic

Checks the syntax of the output logic (if present / available). Errors are explained and the offending line will be highlighted.

### Check Tables

Checks all tables for consistency and errors. Also checks Output Logic (if present / available). Errors will be explained and the current cell positioned so that you can fix the problem. Warnings and information will be displayed in a results window.

### Show CRC

Calculates and displays a "CRC" of the configuration data.

This CRC is stored in the database sent to some types of panels, and can be displayed on the panel's LCD and/or diagnostic terminal so that you can check if the file you have open in SmartConfig corresponds with what is in the panel.

This feature will be added to other panel types in the future.


This new CRC differs from the existing CRCs used for checking Flash / EEPROM chips in the panel, which are different every time you download, as downloads include time and date and other varying data. The new CRC is dependent only on the data you enter into SmartConfig.

Note - the CRC is calculated over **all data** you can enter, even data that makes no difference to the operation of the panel (e.g. Notes in tables and comments in logic). If you want the feature to be most useful, make sure that you don't change the database after downloading it to the panel.

## 2.7 Window

This menu contains the usual Multiple Document Interface commands, including tile / cascade etc, and a list of open documents to select from.

## 2.8 Licence

For more information about the licence refer to [licensing](#) .

### Info

Gives information about your licence and the time remaining before the licence must be renewed. Check this from time to time so that you can get your licence renewed before it expires.

### Terminate

Terminates your licence on the computer you are using. The only way to re-instate the licence is to get a new one.

### Renew

Renew allows you to get a new Licence before your existing licence expires. Also this dialog box will be automatically displayed when you run SmartConfig for the first time, or if the licence has expired. You will be given a 24 digit code to send to your licence issuing office. They will send back a 40 digit licence number for you to enter. Type or paste the licence number into the Licence field and click "Use New Licence". To close the window without entering a new licence click "Close".

Note that if you make substantial changes to the hardware installed in your PC, you may need to get a new licence. The dialog box described above will be automatically displayed.

## 2.9 Tools

### Program Firmware

This allows you to program the firmware of those panel types that support on-site programming of their firmware.

## 2.10 Help

### General Help incl All Systems

This displays the complete help file for SmartConfig including all supported panel types. This includes general help for SmartConfig relevant to all panel types, plus specific help for each panel type supported by the version of SmartConfig.

### Context Sensitive Help

Displays help for the table you are editing or screen you are viewing. The same help can be obtained by pressing F1.

### SmartConfig Manual

Displays this overall help in Acrobat PDF format, suitable for printing.

### Whats New

Shows changes in SmartConfig since its previous version.

### Database Converter Manual

Shows the manual for the F4000/MX4428 Database converter program (in Acrobat PDF format).

### Help for Individual Panels

You can display help for the specific panel types supported by your version of SmartConfig. This may be less confusing than the overall help, especially when using the index and search features.

### About

Shows information about SmartConfig, and the operating system on the PC you are using.

## 3 General Reference

This is a general reference description about the use of SmartConfig that applies to ALL system types. System-specific additional information is available in the relevant sections.

### 3.1 General Editing Instructions

#### Keyboard Shortcuts

Many of the menus and keys can be activated with the keyboard.

To activate a top level menu or visible button, press Alt + the underlined key on the menu or button text.

Individual menu items may have a shortcut key. This key is shown to the right of the menu text.

#### Resizing

You can re-size the SmartConfig window (depending on your video card and monitor) by dragging the edges of the window. SmartConfig will remember your settings for the next time you run it.

#### Undo

To undo changes to the current cell, or a number of cells altered by a single command (e.g. Paste), press Alt+Backspace. See [edit](#)<sup>8</sup>.

#### Input Validation

Some validation of input is performed during typing -

1. if the cell should contain a number you can only type digits.
2. if the cell should contain a list of numbers you can only type digits, commas, and spaces.
3. if the cell should contain a string in a particular format (for example C1/1, C1/2 or R1/1, R3/4, AR, BR etc) you may only be able to type the appropriate keys.

When entering data into a cell which should contain a single number, the number may be adjusted to fit within maximum and minimum limits. In table view you can see the adjusted value in the grid as you type into the top edit box. In form view the number will be adjusted as you exit the cell.

Some cells which usually contain a number may have special cases that are displayed in words e.g. "Default", "Disabled", "Alt", etc. You can select the special cases by typing the first character of the word.

Further validation of input is done as a table is exited, i.e. as another table is selected. If there is a problem the cell causing the problem will be highlighted and you will be given the choice of stopping to fix the problem immediately, or temporarily ignoring the problem so that it can be fixed later or fixed by editing **another** table.

Further checking is done as a database is exported in panel format. Finally the [check tables](#)<sup>14</sup> command does all the previous checks and also warns you about things which **might** be errors.

#### Table View

In table view you can see/edit multiple entities of the same type, e.g., multiple zones or multiple points. However sometimes not all the information may fit on the screen so it may be useful to select [Form View](#)<sup>14</sup>, which displays one entity at a time.

Use the mouse to select a cell to edit. Or navigate to it with :

1. TAB (moves right)
2. SHIFT+TAB (moves left)
3. Up arrow (moves up)
4. Down arrow or ENTER (moves down)

To overwrite the text in a cell, just start typing. To edit the text, press F2, or click in the wide edit box near the top of the SmartConfig window. When editing the text of a cell, the cursor will always be positioned in the wide edit box.

You can change column widths (not saved when you exit the program) by dragging the divider between columns in the header row.

A number of common "Edit" and "View" commands are available with a **right mouse click**. The commands vary depending on whether the mouse is positioned over the heading (top) row of the table or a data row. You can right click a fixed column (a light grey column with fixed information) - this is the only way to sort or filter based on the fixed columns, or to find a cell of a fixed column containing a specific value.

### Form View

In Form View you can see only one entity at a time, e.g., one zone or one point. This is the default view for one-off entities e.g. the System form and the Network form (system dependant).

Use TAB and SHIFT+TAB, or the mouse, to move to another field. To move to another entity (e.g. another zone or another point), use the Next or Previous button.

Most cells can be edited in the form. However if you want to edit a cell where the text does not fit in the edit box on the form, it can be edited in the wide edit box by clicking there.

Passwords (where applicable) are normally displayed as asterisks, e.g., \*\*\*\*\*. This prevents casual observers from seeing secure information. However if you want to check a password, select it and then double-click in the wide edit box at the top of the screen.

### Status Display on Title Bar

The title bar of the each window displays the filename of the window and the name of the currently visible table. The filename will be preceded by an \* if it has been modified but not saved. If the table displayed is sorted or filtered, "Sorted or Filtered" will be appended.

If the document window is maximised (the default) it does not have its own title bar, and the information for the current window will be displayed in the application title bar.

### Status Display on the Status Bar

There is a "Status Bar" at the bottom of the SmartConfig window. This often contains "hints" for the menu command under the cursor, including the full pathname of files on the recently used list. Otherwise it contains the full pathname of the active document.

### Previous Button

Moves to the previous row of the table or previous entity in form view.

### Next Button

Moves to the next row of the table or next entity in form view.

### Find Button

The same as the [Find](#)  menu item.

### Find Next Button


The same as the [Find Next](#)  menu item.

### Notes column in tables

Many tables have a "Notes" column. This is for entering textual notes for you own use, and for the use of subsequent programmers. It has no effect on the operation of the panel.

## 3.2 Template Files

One or more template files for each panel type will be supplied with SmartConfig.

Template files are a starting point for new databases, and contain many settings and profiles. [File | New](#)  gives a choice of template files to start a new job with. Sometimes multiple templates for a panel type will be provided with SmartConfig (e.g. different templates for Australia and New Zealand), and furthermore you can create your own templates.

WARNING. Many of the settings in template files are required for Standards compliant operation.

This applies particularly to "Profiles" (for those panel types that have profiles). Do not change the settings unless you are sure that they will not affect Standards compliance.

### Creating new Templates

Starting from an existing template (i.e. File | New | Panel Type | Template), edit it as required. Click File | Save As. Fill in the Change Log. In the Save As dialog box, click the "Save as type" drop down box, and select template files. Type in a name with no extension. Ensure that the folder selected is the folder SmartConfig is installed in. Click Save.


This new template will then be available when you click "File | New".

It is recommended that you do **not** overwrite the templates supplied with SmartConfig, as your customised version will be overwritten when you install a new version of SmartConfig. However note that the installation of SmartConfig now saves all existing templates under the "Saved Templates" folder (i.e. usually Program Files\Vigilant\SmartConfig\Saved Templates).

### Deleting Templates

You can delete template files in the normal manner in Windows Explorer, My Computer, or even a File Open or File Save dialog box. They will then disappear from the File | New menu.

## 3.3 Licensing

See also [licence](#)  for the menu commands to obtain a licence.

Some distributions of SmartConfig are only available to authorised Tyco employees. If this is the case with your distribution you will not be able to use it until you enter a licence number.

SmartConfig has a security system which requires it to be "enabled" with a licence (supplied by Tyco Safety Products) for each PC on which it is to be used. Licences will typically be issued for a 12 months period, and SmartConfig will stop working after this time unless the licence is renewed. If SmartConfig finds the licence has less than 30 days to run then it will show a warning dialog box when it is started, so that you can renew your licence before it expires.

To obtain or renew a licence the easiest way is to use the computer and onboard e-mail program (e.g., Outlook Express). Enter the requested information and click the Send button to e-mail the request. If you don't have e-mail then the data can be copied to the clipboard and e-mailed, faxed or quoted over the phone.

The licensing contacts are:

Australia		New Zealand	
Email:	reg.vigcfg.au@tycoint.com	Email:	reg.vigcfg.nz@tycoint.com
Name:	Mark Wills	Name:	Craig Barnett
Phone:	03 9538 7220	Phone:	09 826 1715

### Licence Termination

To terminate the licence for SmartConfig and remove SmartConfig from a PC, the best procedure is

- 1 Run SmartConfig. Click the menu "Licence" then "Terminate".
- 2 Exit SmartConfig, then uninstall SmartConfig by clicking "Start", "Programs", "Vigilant", "Uninstall SmartConfig". Alternatively "Start", "Settings", "Control Panel", "Add/Remove Programs", scroll down to and highlight "SmartConfig(remove only)", click "Add/Remove", click "Uninstall".

### Notes

Reinstalling SmartConfig, or a newer version of it, should not affect an existing licence. This

applies regardless of whether the new version is installed "over the top of" the current version (the default), or in a different location.

Uninstalling SmartConfig always removes the licence, even if you uninstall an old version that you are no longer using.

Also note that some licence files are located where you first installed SmartConfig, even if you install a newer version in a different location. Deleting the folder where SmartConfig was first installed will remove the licence.

## 3.4 Terminal Window

The terminal window provides a terminal emulator for use as the diagnostic terminal of a panel. It can connect to several products from Tyco Safety Products Christchurch, and the settings (baud rate, databits, stop bits, parity etc) are pre-programmed for these products.

It can display the 24 lines of the current "terminal" plus 1000 previous lines (the scrollbar area). The 24 current lines are highlighted (compared with the scrollbar lines). The scrollbar can be used to see the scrollbar area.

It interprets cursor addressing and colour commands sufficiently to function as a terminal for an MX4428 in PanelX mode, and to display colour when used as an MXP diagnostic terminal.

It can use a serial port on the PC, or connect to a panel over the a TCP/IP (Telnet) connection. In the latter case, the panel will usually be connected to an "RS232 to Ethernet adapter" that provides Telnet (TCP/IP) connectivity to a local area network or has a gateway to the Internet.

**The TelNet feature allows remote fault diagnosis over the Internet, provided your PC is connected to the Internet and the panel is connected as described below. It also potentially allows remote configuration. Note that downloading of configuration databases remotely will require someone on site to change the Write-Enable jumper as required, and to fully test the system after completion of the download.**

The baud rate (for direct RS232 connections) is selected by choosing the panel type on the File | Terminal Window Menu.

Once the terminal window has been opened, you can reactivate it by clicking in it, or if it has become lost behind the main SmartConfig window click Window | Terminal, or click the terminal icon on the toolbar.

The following commands and settings are at the top of the terminal window.

### Comms Port

The port (Com1 to Com8 or Telnet) can be selected.

NOTE The port will be remembered and automatically used the next time you start the terminal window.

When you select **Telnet**, you must enter the host name or IP address, and optionally the port number, of the Ethernet adapter connected to the panel. **Note** Telnet has been successfully tested over a fast network to a Lantronix "RS232 to Ethernet" adapter connecting to an MX4428 and MX1. The use of Telnet over a wireless or other possibly slow connection has had only limited testing at this time and is not guaranteed. Future information about SmartConfig's Telnet capabilities will be made available as it becomes available.

The RS232 to Ethernet adapter requires programming as follows -

MX4428	MX1
--------	-----



Baud	9600	19200
Databits	8	8
Parity	None	None
Serial Flow Control	X-On / X-Off	None
Network Protocol	Raw TCP/IP (not Telnet)	Raw TCP/IP (not Telnet)

It will also require an IP address and one or more port numbers set up. You will need to get this information from the administrator of the Local Area Network it is connecting to, and possibly get the administrator to arrange access to this IP address and port from the Internet.

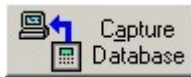
### AutoSend / Capture

When ticked, SmartConfig will automatically navigate to the right point when sending a database to the panel, or capturing a database from the panel. When not ticked, you must navigate manually.

### X Off

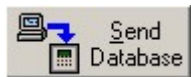
Use X On / X Off flow control in both directions. Normally this should be ticked.

However if you are sending or capturing a binary file, it should be un-ticked. (This is not required in normal operation with supported panels.)



### Capture Database

Capture a database from the connected panel and open it. (Note - the captured database is now automatically opened, in previous versions this was not the case.)



### Send Database

Send the current database to a panel. The "current database" is the database represented by the topmost window of SmartConfig, and its name will be shown as a hint when you pass the mouse cursor over the button.

### Capture File

Capture data received from the connected panel to an arbitrary file which you can select. When capturing, the button will change to "Cancel", and clicking it will cancel the capture and close the file.

### Send File

Send an arbitrary file which you can select to the connected panel. When sending, the button will change to "Cancel", and clicking it will cancel the send function. When cancelled, or the file is completely sent, the button will change back to "Send File" again.

### Close

Close the Serial port or TCP/IP connection, and the terminal window.

### Return

Return to SmartConfig's main window, leaving the Terminal window running. Alternatively click on the main window.

### Menu

Brings up a menu of commands to print and/or copy areas of the display. The same menu can also be obtained by right-clicking the terminal window. Commands include

- **Clear** - erase all lines including the scrollbar region.
- **Copy Screen** - copy the 24 current (highlighted) lines to the clipboard.

- **Copy Screen + Scrollback** - copy all lines to the clipboard (including the scrollbar area starting with the first non-blank line).
- **Print Screen** - print the 24 current (highlighted) lines.
- **Copy Visible Area** - copy the 24 lines currently visible to the clipboard.
- **Print Visible Area** - print the 24 lines currently visible.

## 4 FP1600 / Omega 64 System

This contains information about using SmartConfig to program an FP1600 / Omega 64 system's database.

### 4.1 FP1600 Overview

#### Overview

FP1600 / Omega 64 is a conventional Fire Alarm System that can be expanded in multiples of 16 zones from 16 zones (1 FP1600 board) to 96 zones (1 Master FP1600 board and 5 Slave FP1600 boards).

Each board provides the following programmable inputs and outputs.

1. Input circuits programmable for different modes of operation
2. Ancillary Relay Output (default function "Common Fire")
3. A20 - On board open collector output (default function "Common Defect")
4. A21 - On board open collector output (default function "Common Normal")
5. A1-A19 - 19 programmable open collector outputs (default function matches earlier model LED mimic connector).

#### Related Documents

Refer to document LT0196 "FP1600/OMEGA 64 FIRE ALARM SYSTEM TECHNICAL MANUAL" for more information on the FP1600 and document LT0312 "FP1600/OMEGA 64 FIRE ALARM SYSTEM INSTALLATION AND CONFIGURATION MANUAL" for details on installation and configuration.

#### Hardware / Software Requirements

To use SmartConfig, the FP1600 board(s) must be Mk 3 or later hardware which is clearly identifiable as it has a 9 way D serial connector and a 3 digit display. These boards will have version 4.00 software or later.

#### Terminology

<b>Circuit</b>	An input (1-16) of a board.
<b>Zone</b>	A system-wide numbering scheme for circuits, whereby circuits on the first board are zones 1-16, circuits on the second board are 17-32, etc. Referred to as Zn where n is 1 to 96.
<b>Group</b>	A set of circuits on the same board which must ALL be in alarm to activate the group. Referred to as Gn where n is 1 to 8.
<b>Universal Variable</b>	(Also simply Variable). There are 16 universal variables which are common across the system. These can be activated by zones and groups, and can themselves activate ancillaries. Referred to as Un where n is 1 to 16.
<b>Ancillary</b>	The ancillary relay output (referred to as AR) , or one of 21 ancillary open collector outputs of a board which may drive relays (referred to as An where n is 1 to 21).
<b>System</b>	The collection of FP1600 boards intercommunicating to act as one.

In SmartConfig, when qualification is required to identify which board a group or an ancillary is on (for instance in the Mapped To and Mapped From columns of the Variables Notes table), it will be preceded by Bn where n is the board number (1-6). For instance B1G2 means group 2 of board 1.

#### Programming Concepts

For each FP1600 board, its 16 zones may be mapped individually to groups on the same board (to

provide an N of N or logical AND function), to (system wide) universal variables (to provide a 1 of N or logical OR function), and directly to that board's ancillary outputs (which generally operate using a logical OR of the conditions mapped to them).

Depending on each zone's programming, the zone status will map to system conditions such as common fire, common defect, bell ringing, brigade calling, etc.

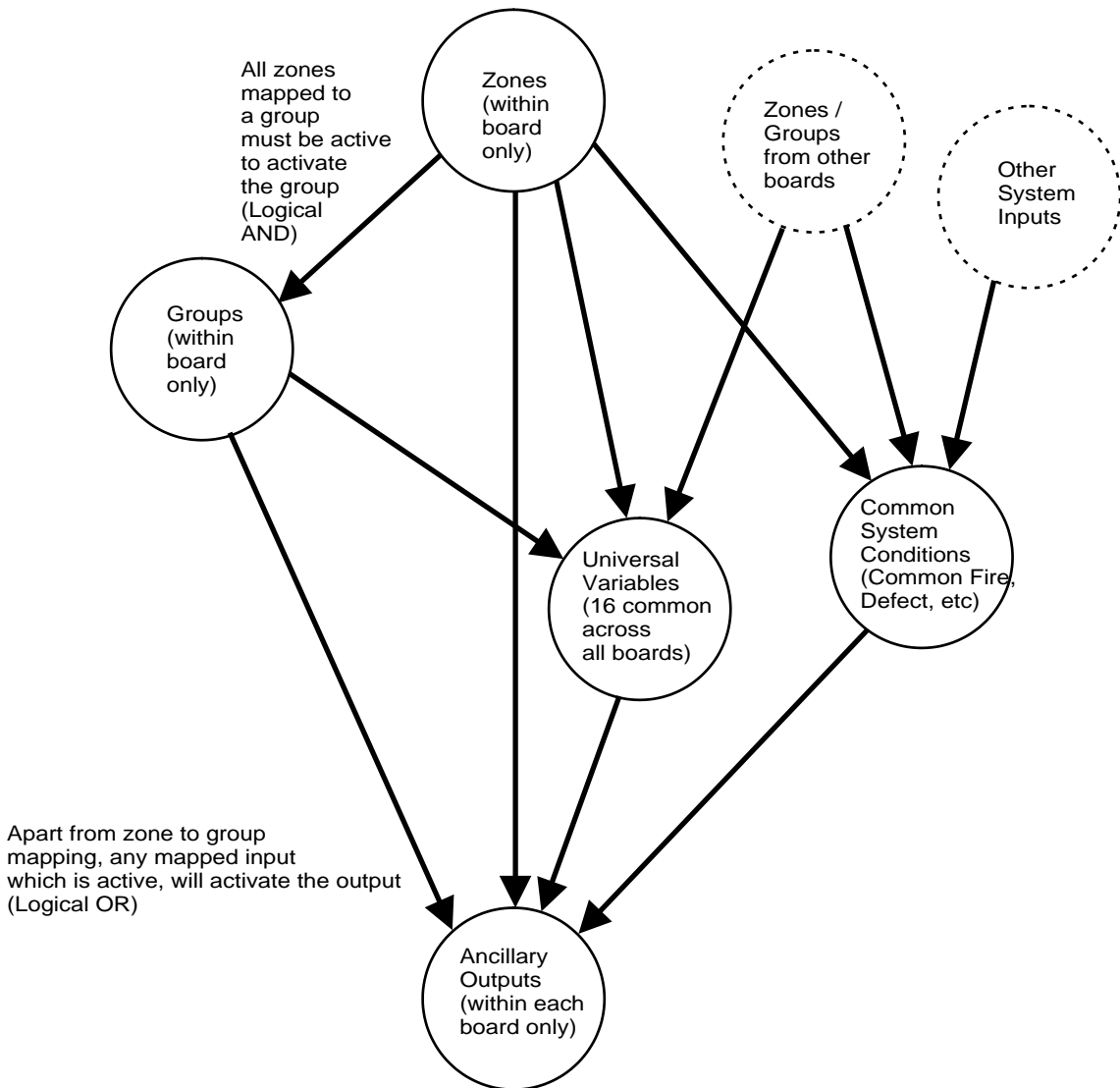
Groups may be mapped to universal variables and directly to ancillary outputs on the same board.

Universal variables can be mapped to ancillary outputs on each board..

Additionally some system states can be programmed to activate (or deactivate) ancillary outputs.

The universal variables are common across the (up to 6) FP1600 boards in a system and they provide the only way of transferring data across boards (apart from common system states which are automatically calculated).

The following diagram shows these relationships :



## 4.2 FP1600 Version Differences

### Overview

Two distinct versions of FP1600 are supported by SmartConfig:

1. Version 4 software.
2. Version 5 software and later.

Version 5 software was introduced for compliance with NZS4512:2003, which required the definition of new circuit types.

When a panel is configured, the [System Config](#) <sup>[28]</sup> page must be set with the correct software version for each board. (Mixed versions are permitted in a multiboard system.)

The circuits supported by version 4 software will now be referred to as legacy circuits within SmartConfig and may only be programmed for a version 5 board if legacy circuits are enabled for that board on the [System Config](#) <sup>[28]</sup> page.

### Circuits

The four default circuit types for version 5 are:

- Detector (dEt)
- Residential (rES)
- Flowswitch (FL)
- Evacuation Control (Ec) (Board 1 only)

The letters in the brackets indicate the display on the FP1600 board 7 segment display display.

All of these new circuit types register Defect on open or short circuit, as per the new requirements of the standard.

The legacy circuits are displayed on the FP1600 board 7 segment displays as follows -

Legacy Circuit Type	Display on Version 4 board	Display on Version 5 board
Smoke	S	LS
Thermal	th	Lth
Combined	Co	LCo
Residential	rE	LrE
Flowswitch	FL	LFL
Evacuation Control	Ec	LEc

## 4.3 FP1600 Getting Started

Start with [File | New](#) <sup>[6]</sup>. Select the type of panel you want to configure.

If the number of boards is higher than 1, enter the number of boards on the [System Config](#) <sup>[28]</sup> page.

Check and adjust the settings on the [Zones](#) <sup>[22]</sup> and [System Config](#) <sup>[28]</sup> pages as required.

If you have special output requirements, program the relationships between zones, groups, universal variables, and ancillaries, as per the previous section ([FP1600 Overview](#) <sup>[19]</sup>). But note that the default programming is adequate in many cases.

Download the database to the system as described in [Communicating with the FP1600 Panel](#) <sup>[29]</sup>.

## 4.4 FP1600 Tables

This describes each of the "tables" used for FP1600 Configuration.

### 4.4.1 Zones

Allows the programmable settings for each zone to be configured.

#### **Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

#### **Circuit No.**

The circuit number for the zone input wiring on the board (1-16) - fixed.

#### **Zone No.**

The logical zone number (1 - 96) (16 x (Board No. - 1) + Circuit No.) - fixed.

#### **Zone Name (optional)**

Enter a name for the zone here. Shown on subsequent pages for configuring zones.

#### **Circuit Type**

The types of circuits which may be selected depend on whether the board is Version 4, or Version 5, and if it is Version 5, whether Legacy mode is enabled. Refer to [FP1600 Version Differences](#)

For version 4 you can select from Disabled, Smoke (legacy), Thermal (legacy), Residential (legacy), Combined (legacy) or Flowswitch (legacy). Board 1 also has Evacuation Control (legacy) as a selection.

For version 5 you can select from Disabled, Detector, Residential, or Flowswitch. Board 1 also has Evacuation Control as an options. If "Legacy mode" on the System Config page is enabled, all the version 4 (legacy) circuit types are also available.

Default characteristics for each circuit type are loaded whenever a change is made.

#### **Callpoint**

Select this option if there is/are Call Point(s) with contact conversion modules on the circuit. (Smoke (legacy) or Residential (legacy) circuit types only).

#### **Gated**

Select if AVF gating is to be used for smoke detectors. (Detector, Residential, Smoke (legacy), Residential (legacy) or Combined (legacy) circuit types only).

#### **Latching**

Select for latching of the alarm state (until a Panel Reset occurs).

#### **Brigade Calling**

Select alarm types for which the brigade is called - none, MCP (option for Residential circuits only) or All.

#### **Bell Ringing**

Select alarm types for which the bells ring (i.e. evacuation output activates) - none, MCP (option for Residential circuits only) or All.

**Power**

Flowswitch, Evacuation Control, Thermal (legacy), Flowswitch (legacy) and Evacuation Control (legacy) circuits can be set for Low or High Power. (High power provides better immunity from interference but draws more quiescent current).

**Residential Smoke Alarm on zone LED**

Residential circuits can be set to display smoke alarms on their zone LED by selecting the 'indicated' option.

**Notes**

Use this space to include additional information about each zone.

**4.4.2 Zone Map Ancil**

Allows mapping of the zones of each board to the ancillary outputs 1 - 21, or the ancillary relay of that board.

**Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

**Circuit No.**

The circuit number for the zone input wiring on the board (1-16) - fixed.

**Zone No.**

The logical zone number (16 x (Board No. - 1) + Circuit No.) - fixed.

**Zone Name**

The zone name as entered on the Zones page.

**Ancil Relay, A20, A21, A1 - A19**

Set the alarm conditions for each zone for which each ancillary output/relay on the same board will be activated

Selection	Meaning	Notes
No	No conditions	
All	All conditions	
MCP	MCP only	Only available for Residential circuits
Smk	Smoke only	Only available for Residential circuits

**Notes**

Use this space to include additional information about each mapping.

**4.4.3 Zone Map Groups**

Allows mapping of the zones of each board to the groups 1 - 8 of that board.

**Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

**Circuit No.**

The circuit number for the zone input wiring on the board (1-16) - fixed.

### Zone No.

The logical zone number (16 x (Board No. - 1) + Circuit No.) - fixed.

### Zone Name

The zone name as entered on the Zones page.

### Grp 1 - 8

Set the alarm conditions on each zone for which each group on the same board will be activated  
-

Selection	Meaning	Notes
No	No conditions	
All	All conditions	
MCP	MCP only	Only available for Residential circuits
Smk	Smoke only	Only available for Residential circuits

Remember that a group will become active only when **all** zones mapped to it are active.

### Notes

Use this space to include additional information about each mapping.

## 4.4.4 Zone Map Variables

Allows mapping of the zones to the universal variables 1 - 16. The universal variables are shared across all boards.

### Board No.

The board number in the system (1-6) - fixed. Board 1 is the Master.

### Circuit No.

The circuit number for the zone input wiring on the board (1-16) - fixed.

### Zone No.

The logical zone number (16 x (Board No. - 1) + Circuit No.) - fixed.

### Zone Name

The zone name as entered on the Zones page.

### U1 - U16

Set the alarm conditions on each zone for which each variable will be activated

Selection	Meaning	Notes
No	No conditions	
All	All conditions	
MCP	MCP only	Only available for Residential circuits
Smk	Smoke only	Only available for Residential circuits

Note that the variables are universal and so do not belong to any particular board. Any activated zone or group condition on any board will activate a variable it is mapped to.



**Notes**

Use this space to include further information about the mapping.

**4.4.5 Group Config**

Allows the settings for each group to be programmed.

**Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

**Group No.**

The group number on the board (1 - 8) - fixed.

**Mapped From**

Lists zones that map to this group.

**Mapped To**

Lists ancillaries and/or variables that the group maps to.

**Group Name (optional)**

Enter a name for the group here. Shown on subsequent pages for configuring groups.

**Latching**

Select for the group's state to be latched (until a Panel Reset occurs).

**Brigade Calling**

Select for the group to call the brigade.

**Bell Ringing**

Select for the group to ring the bells (activate the evacuation output).

**Notes**

Use this space to include further information about the group.

**4.4.6 Group Map Ancil**

Allows the mapping of groups to ancillary outputs 1 to 21 or the ancillary relay to be set. (Mapping is only possible within a board).

**Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

**Group No.**

The group on the board (1 - 8) - fixed.

**Mapped From**

Lists zones that map to the group.

**Mapped To**

Lists variables that the group is mapped to, as entered on the Group Map Variables page.

**Group Name**

Name of the group as entered on the Group Config page.

**Anc Relay, A20, A21, A1 - A19**

Ancillary outputs or relay on the same board to be activated when the group is activated.

**Notes**

Use this space to include additional information about each mapping.

**4.4.7 Group Map Variables**

Allows the mapping of groups to the system-wide universal variables 1 to 16.

**Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

**Group No.**

The group on the board (1 - 8) - fixed.

**Mapped From**

Lists zones that map to this group.

**Mapped To**

Lists ancillaries that this group is mapped to, as entered in the Group Map Ancil page.

**Group Name**

Name of the group as entered on Group Config page.

**U1 - U16**

Universal variables to be mapped to by the group. Note that the variables are universal and so do not belong to any particular board. Any activated group or zone on any board will activate a variable it is mapped to.

**Notes**

Use this space to include additional information about each mapping.

**4.4.8 Ancil Config**

Allows the settings for each ancillary output on each board to be programmed.

**Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

**Ancil No.**

The designation of the ancillary on the board - fixed.

**Mapped From**

Lists zones, groups and variables that map to the ancillary output.

### **Latching**

Select for the output state to be latching (until a Panel Reset occurs).

### **Follow Trial Evac**

Select if ancillary is to be forced on by the External Evacuation Switch.

### **Follow Silence Alarms**

Select if ancillary is to be forced off by the Silence Alarms Switch.

### **Obey Services Restore**

Select if ancillary is to be forced off by the Services Restore Switch.

### **Follow Lamp Test**

Select if ancillary is to be forced on by Lamp Test.

### **Follow Evac Relay**

Select if ancillary is activated by the Evacuation (bells) relay.

### **Follow Common Fire**

Select if ancillary is activated by Common Fire.

### **Follow Common Defect**

Select if ancillary is activated by Common Defect.

### **Follow Common Normal**

Select if ancillary is activated by Common Normal.

### **Follow Charger Inhibit**

Select if ancillary is activated by Charger Inhibit.

### **Follow Panel Reset**

Select if ancillary is activated by Panel Reset.

### **Notes**

Use this space to include additional information about each ancillary.

#### **4.4.9 Ancil Map Variables**

Allows the mapping of universal variables to ancillary outputs.

Note that the layout of this table is different to other tables - in this case the columns are the inputs and the rows are the outputs.

#### **Board No.**

The board number in the system (1-6) - fixed. Board 1 is the Master.

#### **Ancil No.**

The number of the ancillary on the board (A1 - A21, Anc Relay) - fixed.

#### **Mapped From**

Lists the zones and groups that map to each ancillary.

### **U1 - U16**

Select to map the universal variable to each ancillary.

### **Notes**

Use this space to include additional information about each mapping.

## **4.4.10 Variables Notes**

Page for entering additional information about the universal variables.

### **Variable No.**

The universal variable U1 to U16. Fixed.

### **Mapped From**

A list of zones and groups (on various boards) that map to the variable.

### **Mapped To**

A list of ancillaries (on various boards) that the variable maps to.

### **Notes**

Use this space to include additional information about each variable.

## **4.4.11 System Config**

Allows the system configuration settings to be set.

### **Number of boards in system**

A system is made up from one master and up to 5 slaves. This specifies the total number of boards.

### **Brigade**

Specifies whether the system is connected to the brigade, or if it operates in Local Mode.

### **Master Door Open = Services Restore Active**

Select this option if the master cabinet door open is to have the same effect as operating the Services Restore Switch.

### **Global Flowswitch Delay**

Specifies the global flowswitch delay in seconds (0 to 25 in multiples of 5).

### **Battery Low volt adjust**

Specifies adjustment to the Battery Low Defect voltage threshold. Allows an offset from the default value (P0) up to 500 mV above (P5) or down to 700 mV below (-7).

### **Number of RZDUs in system**

Specifies the number of RZDUs that are connected to the system (0 up to 8). Note - include only **replying** RZDUs, non-replying RZDUs (such as an IONet controller) should not be included in this

number.

### **Evacuation Supervision**

Select separately for each board as to whether it supervises its evacuation (bells) output.

### **Firmware Versions**

Select the version of software that is running in each board being configured.

### **Enable Legacy Devices**

Select for any version 5 boards that are to be configured with legacy circuits.

### **Residential Circuit Delays**

Specifies the delay in seconds for each board (0 or multiples of 10 up to 250). A value of zero selects latching operation with no automatic reset of the residential smoke detectors. Otherwise this is the delay before a residential smoke detector alarm condition is automatically reset.

## **4.4.12 Change Log**

Information is requested each time the configuration .f16 file is saved. This is stored on the Change Log page. It records the date, name of the person who made the changes, and a description of the change made.

## **4.4.13 Comments**

Any information not already defined or described on earlier pages can be stored here for future reference.

## **4.5 Download / Upload**

This describes the procedure for downloading a database to the FP1600 panel, and uploading a database from the FP1600 panel.

### **4.5.1 Communicating with the FP1600 Panel**

A window is brought up by the Download to / Upload from Panel command on the File menu to control communications between SmartConfig and the FP1600 Panel.

#### **Overview**

To program or read the information in a multi-board (Omega 64) system, each board must be programmed or read individually. A Serial cable (LM0076) must be connected between the PC and each board in turn, and each board programmed or read from in turn.

It is often necessary to power a board down and up in order to get into "Programming Mode". You will be prompted when this is needed. When in Programming Mode, the display on the board will indicate 'ACO'.

When reading or writing the data in a board, you must select the board number, or select "Auto". If you select "Auto", SmartConfig will read the existing board number from the connected board and write or read the data for that board. If you select a board number other than Auto, SmartConfig will check that the existing number of the connected board matches the selected board number. If it does not match you can continue if you choose, and if you are writing the database, the board number of the connected board will be changed to the selected board number.

Note - A Master database cannot be written into a Slave board and a Slave database cannot be written into a Master board.

However the data from a Slave board can be read into the Master portion of SmartConfig's database, and data from a Master board can be read into the Slave portion of SmartConfig's database. SmartConfig will ask you to confirm this operation twice. You must check (and amend as required) the resulting data carefully. (At least the system data may be wrong.)

### Reading Configuration From a Panel to SmartConfig

1. Select Write / Read Panel Database from the File menu.
2. Select the board to be read from and the serial port to use.
3. Connect a LM0076 serial cable to the serial port on your PC and the programming port of the desired FP1600 board.
4. Click the Read Database button.
5. If prompted, turn the panel power off and on. (If the board is already displaying 'ACO', this will not be necessary).
6. SmartConfig will initiate communications with the board and read the EEPROM contents.

If there are more boards to read from, repeat from step 2 for each board. Otherwise, click "Close" to exit.

NOTE - when extracting databases from an FP1600/OMEGA64 panel using Smart Config, it is necessary to extract the database from the master board first, before the slave boards, so that the correct number of boards is assigned before reading any slave databases.

### Writing Configuration From SmartConfig To a Panel

1. Select Write / Read Panel Database from the File menu.
2. Select the board to be written to and the serial port to use.
3. Connect a LM0076 serial cable between the serial port on your PC and the programming port of the desired FP1600 board.
4. Insert the "Data Program Enable" link on the FP1600 board.
5. Click the Write Database button.
6. If prompted, turn the panel power off and on. (If the board is already displaying 'ACO', this will not be necessary).
7. SmartConfig will initiate communications with the board and write into the EEPROM contents.
8. Remove the "Data Program Enable" link.

If there are more boards to write to, repeat from step 2 for each board. Otherwise, click "Close" to exit.

#### Note

Always save the file on your PC (and back it up) as well as writing it to the panel. All the "Notes" and "Comments" are only saved on the PC file. They are not saved in the panel.

## 4.6 FP1600 Reference

This contains general reference about using SmartConfig to configure an FP1600.

### 4.6.1 Checks

Checking a database ([Check | Check Tables](#)<sup>[11]</sup>) checks the following and violations are reported as warnings

1. Any zones that have been mapped to a group that isn't mapped anywhere.
2. Any zones that have been mapped to a variable that isn't mapped anywhere.
3. Any groups that have been mapped to a variable that isn't mapped anywhere.
4. Any groups that have been mapped to variables or ancillaries, but have no zones mapped to them.
5. Any variables that have been mapped to ancillaries, but have no zones or groups mapped to them.

## 5 IO-Net

This describes SmartConfig's IO-NET compiler and facilities for programming an IO-NET Controller, and for extracting programs from an IO-NET Controller.

For further information refer to LT0115 VIGILANT IO-NET PROGRAMMABLE CONTROLLER USER'S MANUAL.

See the following pages (press next at the top of the page, or expand the table of contents), for more details.

### 5.1 Opening Files

SmartConfig can open both the old style of IO-NET files prepared by a text editor (usually \*.txt) and the new files saved by SmartConfig (\*.ion).

#### Old Style Files (\*.txt)

To open an old style file, click File then Open. With the "Files Of Type" drop-down control, choose the "Import text files (\*.txt)" option. Alternatively if the extension of your file is not .txt, choose the "All Files (\*.\*)" option. Browse to the file you want to import and click "Open".

Most of the file will be imported into the Logic page, however the parameters will be separated out and displayed in the Parameters page. A matching network profile will be chosen, or a new one created if the network parameters do not match a standard configuration. Note that as the standard network profiles for 2400 baud, 4800 baud, and 9600 baud are all the same, 2400 baud may be chosen even if the actual baud rate in the system is 4800 baud or 9600 baud. It is suggested that you change the Network Profile on the Parameters page to match the actual baud rate of the system so that the system will be documented correctly.

At the start of the logic, there may be a large number of Comment lines, i.e. lines starting with ;. These lines would have previously documented the parameters which have now been stripped out. They can now be deleted.

#### SmartConfig Files (\*.ion)

These files will load exactly as they were saved.

### 5.2 Parameters

This section describes the Parameters page.

The following parameters can be set on the Parameters page.

<b>System Name</b>	Enter a name for the system, for documentation purposes only.
<b>Network</b>	Select the required network profile. The choices available will be the entries in the <a href="#">Network Profiles</a> <sup>[50]</sup> table. Normally you only need to select a choice with the required baud rate and modem selection.
<b>Specify Address</b>	Refer to <a href="#">Address</a> <sup>[32]</sup> .
<b>Address</b>	Refer to <a href="#">Address</a> <sup>[32]</sup> .
<b>Max Stations</b>	Refer to <a href="#">Max Stations</a> <sup>[32]</sup> .
<b>Connects to RZDU</b>	Refer to <a href="#">Panel Number</a> <sup>[32]</sup> .
<b>Panel Number</b>	Refer to <a href="#">Panel Number</a> <sup>[32]</sup> .
<b>RZDU Protocol</b>	Refer to <a href="#">RZDU Protocol</a> <sup>[33]</sup> .
<b>TX Zone Data</b>	Refer to <a href="#">TX Zone Data</a> <sup>[33]</sup> .
<b>Allow Reduced RAM</b>	Refer to <a href="#">Allow reduced RAM</a> <sup>[34]</sup> .
<b>Total User RAM</b>	Refer to <a href="#">Off-line RAM Assumptions</a> <sup>[35]</sup> .
<b>Base Page RAM</b>	Refer to <a href="#">Off-line RAM Assumptions</a> <sup>[35]</sup> .

See the following pages (press next at the top of the page, or expand the table of contents), for

more details.

### 5.2.1 Address

The address of the target IO-NET controller into which the source file is to be programmed can be specified in the range 0 to 127 as the Address in the Parameters page. This must be the same number as assigned to the address dipswitch on the IO-NET Controller.

Normally it is not **essential** to specify the address of the target controller in the user source file. Instead select the radio button "Use DIP Switch". However specifying the address may reduce the RAM requirements of the program.

The parameter to assign the address should be included only if a "Ram limit exceeded" error is produced by the compiler. In this case select "Specify here also to reduce RAM required" and specify the Controller address in the "Address" box. It is important that the EPROM be installed in a target controller with **the same address set on the DIP switches**. If not, the controller may display an error cadence on its status LED after power up and not run (see LT0115 Section 7).

This parameter was referred to as ADDR in LT0115 version 2.

### 5.2.2 Max Stations

Max Stations should be assigned a value of 0 to 255 to match the number of the highest controller address you are ever likely to connect to the network. Any controller connected to the network with an address higher than this will not be given access to the network. A reasonable upper limit value should be used for Max Stations. Too large a value can reduce the efficiency of the network and increase the start-up time of the network or increase the access time when adding a new station to the network. Too low a number means you may need to reprogram some controllers when adding to the network later.

The information in the following paragraph may be useful if adding new stations to an existing network and the new stations have addresses greater than the value of the Max Station parameter on the existing controllers.

It is not essential for all stations to have the same value for Max Stations, but is probably desirable. For a given station N, it gets access to the network only if the next lowest numbered controller present on the network has a Max Stations greater than or equal to the value N. For example, if controllers 5, 7 and 12 are the only controllers physically present on the network then the Max Stations value of controller 7 usually determines whether controller 12 gets access to the network. BUT, note that if controller 7 is disconnected from the network for a while it would then be the Max Stations value of controller 5 which determined controller 12 access.

Note that in non programmed mode a default value of 31 is assigned for Max Stations.

This parameter was referred to as MXST in LT0115 version 2.

### 5.2.3 Panel Number

The IO-NET Controller program can use the zone status transmitted by an RZDU compatible FIP (F3200, MX4428/F4000, MX1, Sigma5, FP1600, NDU, NLDU) via a local RZDU data connection or from zone data received on the IO-NET network from other controllers that have their own RZDU data connection.

Each controller which has a local RZDU data connection must have the "**Connects to RZDU**" box ticked and a unique panel number (0-63) assigned to the fire panel in the box "**Panel Number**".



Each controller with a local RZDU connection must be given a unique panel number, so that the particular panel's data can be identified on the IO-NET network by other controllers. These should be carefully recorded in the documentation of the system. The panel number could often be the same as the panel's SID number, unless some of the panels connecting to the IO-NET network are on different panel-link networks and happen to have the same SID numbers. Alternatively you could make each panel number equal to the IO-NET address that the panel connects to.

Refer to [Zone Data](#)<sup>[40]</sup> for details of how to access data from RZDU connections in your logic.

#### Notes

SmartConfig is not able to detect duplication of panel numbers. However an IO-NET controller is able to detect such duplication and will set the ZNF operand to TRUE if it receives a message of zone data from the network where the panel number is the same as its own RZDU connection panel number.

The local RZDU data connection may be enabled even if none of the zone data from it is accessed in the control program - this could happen if it was desired for the controller to transmit the zone data onto the network. A controller may access zone data from its own RZDU data connection (if any) or zone data received from the network.

Refer to LT0115 Section 8 for information on RZDU zone data refresh rates and the delay from a zone change of state occurring to the new state being received by an IO-NET controller.

This parameter was referred to as RZDU in LT0115 version 2.

### 5.2.4 TX Zone Data

A controller may be programmed to transmit all of the zone data it receives on its local RZDU data connection onto the IO-NET network for use by other controllers. To enable the transmission of local RZDU data onto the network, tick the box "**TX Zone Data**". The Panel Number described above must also be included. The panel number will be included in every message of zone data transmitted onto the IO-NET network so that other controllers can identify the zone data.

NOTE - for a network using 1200 baud modems the transmission of zone data takes up a significant amount of time on the network and if a controller does not get an opportunity to transmit often enough, it may have to continually discard some of the zone status data that it receives from its RZDU input - which could result in the status of some zones NEVER being transmitted on the network. Refer to LT0115 section 8.4 for further information.

This parameter was referred to as TXZD in LT0115 version 2.

### 5.2.5 RZDU Protocol

The data received on the local RZDU connection from a fire panel can be in one of two different formats or "protocol types". Older versions of panels transmit data in a format referred to as "LED protocol" and newer versions of panels can be programmed to transmit using this or the newer "LCD protocol". The older LED protocol transmits only led/zone status and the newer LCD protocol also sends event text and information for display on a remote LCD and allows more functions for isolating/ resetting zones from remote RZDUs/RDUs. Refer to LT0115 Section 8 for information on how to configure some of the fire panels for use with IO-NET.

Select the correct RZDU protocol as being used by the panel.

If an IO-NET controller is connected to a fire panel which is transmitting the wrong version of the RZDU protocol, then the IO-NET controller will not operate correctly.

This parameter was referred to as RZDP in LT0115 version 2.

## 5.2.6 Allow reduced RAM

If the compiler produces a "RAM storage limit exceeded" error, the "**Allow Reduced RAM**" tickbox may need to be ticked to reduce the size of the RZDU buffer (if any) allocated in the controller. The size of the buffer is set by the compiler and affects the capability of the controller to transmit zone data onto the IO-NET network. The "Allow Reduced RAM" parameter has no effect (and is greyed out) if the controller is not programmed to transmit zone data onto the network.

If the controller is programmed to transmit zone data onto the IO-NET network then the compiler will attempt to allocate an RZDU receive buffer size of at least 60 bytes - this is twice the size of the maximum sized message which can be received on its local RZDU data connection. The controller receives a message of zone data on its RZDU data connection once every 2 seconds, but is not able to transmit this data onto the network immediately - it must store it briefly until it gets an opportunity to transmit it onto the network.

The average and maximum lengths of time the controller must wait before it can transmit this zone data onto the network is dependent on the baud rate of the network, the number of controllers connected to the network, also the "activity level" of the network. Some approximate figures are given in Section 8.4 of LT0115 for the length of time a controller may have to wait before it is able to transmit on the network.

When the controller gets an opportunity to transmit on the network it will immediately transmit all of the zone data that it currently has stored in its buffer (which could require multiple messages being transmitted - and it may also transmit the status of its own input points) before relinquishing transmission right to the next controller.

The controller must therefore have a large enough RZDU receive buffer to be able to store a reasonable amount of zone data while it is waiting for the opportunity to transmit it. On a typical system a controller would get an opportunity to transmit more often than once every 2 seconds so that the message of zone data which it receives from the RZDU data connection every 2 seconds has probably been completely transmitted by the time the next RZDU message is received.

However, there may be times when the controller has not transmitted the last data received from the RZDU input. The size of the buffer set by the compiler determines how much zone data the controller can store while it is waiting to transmit it. Since the controller has only a relatively small amount of total RAM storage space, some flexibility has been provided by the compiler to allow some control of the size of the RZDU buffer allocated.

If the controller has its local RZDU data connection enabled but is not programmed to transmit zone data onto the IO-NET network then the compiler will allocate an RZDU receive buffer size of minimum 30 bytes - this is all that is needed to store the maximum size message that can be received from the RZDU input and the controller never needs to store more than one message at a time. (In actual fact the compiler will allocate all of the free RAM it has available to the RZDU buffer which may result in a buffer size of more than 30 bytes).

If the controller is programmed to transmit zone data onto the network then the compiler will initially attempt to allocate a minimum of 60 bytes to the buffer - which it will probably be able to do for a lot of control programs. However some larger programs may not have enough free RAM for a 60 byte buffer. By ticking "Allow Reduced RAM" the controller will allow the buffer size to be reduced to a minimum of 30 bytes.

The compiler will ALWAYS allocate all of the free RAM it can to the RZDU receive buffer, so the buffer size could be any size from 30 to 60 bytes when the "Allow Reduced RAM" is selected.

This parameter was referred to as MRAM in LT0115 version 2.

## 5.2.7 Off-line RAM Assumptions

When compiling **offline** (see [Checking a Program](#)<sup>[52]</sup>), SmartConfig by default assumes the chip will be a C8/C8A and the controller software version 2.01. (In this case there are 141 bytes of total RAM and 61 bytes of base page RAM available in the target.)

If you know the chip will be a C9A and the software Version 2.01 (i.e. the software is SF0239), you can set "Total User RAM" to 189 and "Base Page RAM" to 77.

Other software versions may have different amounts of RAM available. You can check that your program will fit by compiling **online** (see the Compile Only command in [Programming an IO-NET Controller](#)<sup>[54]</sup>).

## 5.3 Output Logic

This section describes the text you can type in the Logic page.

See the next few pages (press next at the top of the page, or expand the table of contents), for more details.

### 5.3.1 General Format

The Logic page may contain any number of output logic expressions (limited only by the available space in the target IO-NET Controller) which assign the value of an expression to a local output or internal variable or timer.

e.g.  $O4 = I7/9O \text{ OR } I15/1A \text{ AND } V3$  ; to control output 4

and may also assign values to logic variables

e.g.  $V5 = V4 + I3/26A$

and to timers

e.g.  $T1[15,25] = I1/4S$

At run time the target controller will evaluate all of the logic expressions sequentially in exactly the order they appear in the logic page.

When the controller gets to the end of the logic expressions it returns to the start and cycles through forever.

Expressions which assign values to outputs do NOT have to appear in numerical order of the output number. They may be in any order - however, the order that the expressions appear in the source file does NOT determine the order in which the PHYSICAL output changes. The physical state of the outputs is only updated at the completion of each pass through the entire list of logic expressions (and also at intervals of 100 milliseconds) so that the actual state of the physical output may not necessarily follow the logical state of the output generated by the logic expressions if the logical state changes quickly.

There does not have to be a logic expression for every output.

The same output should be controlled by ONLY ONE EXPRESSION and the compiler will display a warning message if more than one expression is given for the same output.

e.g.  $O5 = V1 \text{ OR } V2$   
 $O6 = I1/1A$   
 $O5 = O6 \text{ OR } I29/3S$

is controlling output 5 with two different expressions and the actual state which appears at the physical output will be intermittent.

Similarly, the program should not contain more than one expression defining the input condition and start/stretch delays of the same timer.

The compiler will display a warning if this is done. The state of a timer may be accessed as many times as desired in multiple equations on the "right hand side" of a logic expression.

Expressions which assign values to variables may be in any order with more than one expression for the same variable being allowed if desired. The variable will then keep that value until the next occurrence of an expression which assigns a value to that variable. It is recommended that the same variable is not used for two different purposes, except perhaps by an experienced user or if the compiler produces a "RAM limit exceeded" error.

Comments may be included anywhere in the source file. Any text on the same line following a semicolon ; is treated as a comment with the exception of the semicolon used in the NA function.

Be careful not to unintentionally convert a real logic expression into a comment by accidentally placing a semicolon at the start of, or part way through, a logic expression.

Blank lines may appear anywhere in the source file.

Each logic expression must fit entirely on a single line, but long lines are allowed- up to 299 characters. Variables can be used to simplify complex or long expressions. Following the last logic expression in the file, the keyword END must appear on a new line to terminate the list of expressions. Any lines following the END statement may contain comments only.

Upper or lower case characters are allowed everywhere and are treated identically.

A logic expression does not have to start in column one, it may start in any column ie. a logic expression may be preceded by any number of spaces on that line.

In general, individual operands such as I15/3A should not contain any embedded space characters (in some cases this is not strictly enforced and embedded spaces are allowed), but spaces may be used freely between operands and operators.

It is not essential to have any spaces between operands and operators but they should be included for clarity.

Each logic expression is of the form :

Ox = Operand {Binary operator Operand} ...

to assign the state of output x (x = 1 to 32)

or

Vx = Operand {Binary operator Operand} ...

to assign a value to logic variable x (x = 1 to 256).

or

Tx[mmm,nnn] = Operand {Binary operator Operand} ...

to define the input condition and timer delays (start, stretch) for timer x (x = 1 to 128). The start delay (mmm) and stretch delay (nnn) are required and may be any value from 0 to 255 seconds.

The curly brackets {} which are not part of the expression indicate that their contents are optional and may be repeated any number of times. The right hand side of each expression consists of

one or more operands with each pair of operands separated by a binary operator.

All operands return a logical value of TRUE or FALSE. If the right hand side of an expression evaluates to TRUE then the specified output on the left hand side will be turned ON (and FALSE = OFF) or if the left hand side is a variable, then the variable will be assigned a value of TRUE or FALSE. See [Timers](#)<sup>[45]</sup> for more details on the operation of timers.

If the output or variable which appears on the left hand side of the = also appears as an operand on the right hand side then the value used on the right hand side is the value of the operand before the new assignment.

Any operand may be preceded by the unary operator NOT (the symbol ^ may also be used) which complements the value of the operand.

Each operand may itself consist of an expression enclosed in parentheses () with a maximum nesting level limited by the amount of stack space required for evaluation at run time. The nesting level limit is 16 which is more than enough for any foreseeable requirement.

See also [Example Programs](#)<sup>[51]</sup>.

### 5.3.2 Operators

The binary operators which may be used are:

AND - logical AND (alternatively use .)  
 OR - logical OR (alternatively use +)  
 XOR - exclusive OR

The one unary operator which may be used is:

NOT - logical complement (alternatively use ^)

The single character "+" (plus) may be used instead of OR, the character "." (fullstop) may be used instead of AND, and the character "^" may be used instead of NOT.

The truth tables for AND, OR, XOR and NOT are (F = FALSE, T = TRUE)

A AND B	<u>A</u>	<u>B</u>	<u>RESULT</u>
	F	F	F
	F	T	F
	T	F	F
	T	T	T

A OR B	<u>A</u>	<u>B</u>	<u>RESULT</u>
	F	F	F
	F	T	T
	T	F	T
	T	T	T

A XOR B	<u>A</u>	<u>B</u>	<u>RESULT</u>
	F	F	F
	F	T	T
	T	F	T
	T	T	F

NOT A	<u>A</u>	<u>RESULT</u>
	F	T
	T	F

Example NOT A OR NOT B

<u>A</u>	<u>B</u>	<u>RESULT</u>
----------	----------	---------------

F	F	T
F	T	T
T	F	T
T	T	F

### 5.3.3 Order of evaluation

The order of evaluation of operands is important and can affect the result of the expression.

When more than one operand appears in an expression, the order of evaluation of the operands is not necessarily left to right but depends on some rules as follows:

- A. The contents of parentheses are evaluated first.

e.g.  $O3 = V1 \text{ AND } (V2 \text{ OR } V3)$

will cause "V2 OR V3" to be evaluated first with the result being "ANDed" with V1 resulting in the following truth table.

<u>V1</u>	<u>V2</u>	<u>V3</u>	<u>O3</u>
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	F
T	F	F	F
T	F	T	T
T	T	F	T
T	T	T	T

- B. Each of the operators AND, OR, XOR and NOT have a priority which determines the order of evaluation when not overridden with parentheses as follows:

The NOT operator has highest priority, AND has the next highest priority, and OR and XOR have equal and lowest priority. Where operators have equal priority the order of evaluation is left to right.

E.g. NOT V1 AND V2 has a truth table of:

<u>V1</u>	<u>V2</u>	<u>RESULT</u>
F	F	F
F	T	T
T	F	F
T	T	F

with the "NOT V1" being done first

but NOT (V1 AND V2) has a truth table of:

<u>V1</u>	<u>V2</u>	<u>RESULT</u>
F	F	T
F	T	T
T	F	T
T	T	F

E.g. V1 OR V2 AND V3 results in the "V2 AND V3" being evaluated first because the AND operator has a higher priority than OR.

Operators of equal priority result in evaluation of operands from left to right. E.g. V1 OR V2 XOR V3 OR V4 XOR V5 will result in the operands being evaluated in

the order they appear with V1 first and V5 last.

If unsure, use parentheses. The use of parentheses does not affect the size of object code generated.

### 5.3.4 Operand Types

This describes the 7 different Operand Types:

1. The condition of one of the 32 inputs on the local controller or inputs on a remote IO-NET Controller received via the network.
2. The state of a local output on the controller.
3. The condition of any zone on a compatible fire alarm panel received via an RZDU connection directly wired into the controller or from zone data received on the IO-NET network.
4. The state of a local variable V1 to V256.
5. The "State" of an NA function.
6. System conditions and constants including the "scan status" of any IO-NET controller or fire panel RZDU data link.
7. The state of a timer.

#### 5.3.4.1 IO-Net Inputs

The four conditions of a circuit input may be accessed as follows:

Ix/yO (open circuit)  
Ix/yN (normal)  
Ix/yA (alarm)  
Ix/yS (short circuit)

Where x is the address of an IO-NET Controller 0 to 255, y is the number of a circuit input 1 to 32 and O, N, A, S refer to the four voltage bands where S is the lowest band (short circuit) and O is the highest (open circuit). The operand returns true if the circuit input x/y is in the specified condition O, N, A or S at the time the logic expression is evaluated.

NOTE - currently the maximum address which can be physically set (with dipswitches) on a controller is 127, but the compiler allows a controller number of 0-255 to be used in expressions.

The circuit input x/y can be any input on any IO-NET controller including the inputs of the controller running this program.

E.g. O1 = I3/15A OR I0/32S results in output 1 being on if either input 15 on controller 3 is in Alarm or if input 32 on controller 0 is in short circuit.

When the controller is first powered up the initial state of all inputs will be normal until their true state is received either from the network for remote controller inputs or from input boards for local controller inputs. If the input termination board is not connected the associated inputs will all be in an open circuit condition.

### 5.3.4.2 IO-Net Outputs

The local outputs can appear as an operand on the right hand side of an expression as:

OxF or O<sub>x</sub>N where x is the number of an output 1-32

NOTE It is not possible to use output states of other controllers.

OxF returns TRUE if the output is OFF, otherwise FALSE.

O<sub>x</sub>N returns TRUE if the output is ON, otherwise FALSE.

E.g. O2 = O1F OR O32N; will result in output 2 being ON if output 1 is off or output 32 is on.

After power-up or re-start of the controller all outputs will initially be off.

### 5.3.4.3 FIP Zone Data

The condition of a zone on a compatible fire panel may be accessed as follows:

Zpp:nnnA	returns TRUE if the zone is in alarm and NOT isolated
Zpp:nnnF	returns TRUE if the zone is in fault and NOT isolated
Zpp:nnnI	returns TRUE if the zone is isolated
Zpp:nnnN	returns TRUE if the zone is normal (not isolated, not in fault, and not in alarm)

where pp is the panel number, 0-63, nnn is the zone number 1-528. Zone data from multiple fire panels may be accessed but the total number of panels accessed should be 8 or less.

Each IO-NET controller which has an RZDU data link directly connected is programmed with a panel number and whether the zone data should be transmitted over the IO-NET network or not. The zone data which it transmits onto the IO-NET network will include the panel number for identification by other controllers. Any controller may access zone data from both its own directly connected RZDU data link and from zone data which it receives via the IO-NET network.

e.g. O32 = Z3:1A AND Z4:528N  
results in output 32 being on if zone 1 on panel 3 is in alarm (and not isolated) and zone 528 on panel 4 is normal.

When the zone corresponds to an Ancillary Control Zone on an MX4428/F4000 or a relay on an F3200 FIP then the conditions A,I,F,N are

Zpp:nnnA	returns TRUE if zone activated and NOT isolated
Zpp:nnnF	returns TRUE if zone in fault and NOT isolated
Zpp:nnnI	returns TRUE if the zone is isolated
Zpp:nnnN	returns TRUE if the zone is normal (not isolated, not in fault, and not activated)

After power-up or restart of the controller the state of all zones will initially be normal until the current actual state is received. This could take up to 7 or 8 minutes for a system with 528 zones depending on which version of software is being used in the fire panel. For example, in the latest version of MX4428 panel software the time to refresh the state of all 528 zones is approximately 70 seconds.

### 5.3.4.4 Variables

Variables may be used for storage of logical values.

A variable is accessed with Vnnn where nnn is a value 1-256.

There are up to 256 variables available, the actual number will be limited by available RAM storage in the controller. A variable has a logical value of either TRUE or FALSE.



E.g. V1 = Z1:6A OR Z3:2F  
 V2 = I0/1A OR I0/2S  
 V3 = V1 OR V2  
 V4 = V1 AND V2

Variables should be used "contiguously" starting with V1, then V2, V3 .... etc, to minimise RAM usage in the IO-NET Controller.

The compiler will display an error message if the RAM storage limits of the controller are exceeded.

When the controller is first powered up or restarted, the initial value of all variables is FALSE.

A variable may be useful for holding the result of an expression which is common to several different logic expressions. This can result in a saving of EPROM space.

Variables are also useful in NA functions because an NA function is not allowed to use an expression in its operand list. This can be overcome by assigning the expression to a variable first and accessing the variable in the NA function.

Variables are also useful for other things. For example, the expression V1 = NOT V1 will toggle the value of V1 each time it is executed and produce an oscillating value.

The expression  
 V1 = V1 OR ZNF  
 will latch the occurrence of a zone network fault. (A mechanism to clear this will probably be required.)

**5.3.4.5 New Alarm Function**

The NA function may be used as an operand in any logic expression and has the form:

NA (reset input; operand list)

where "operand list" consists of a list of fundamental operands separated by commas and includes:

OnnF	local output nn (1-32) off
OnnN	local output nn (1-32) on
Ix/yO	input y (1-32) on controller x (0-127) open circuit
Ix/yN	input y on controller x normal
Ix/yA	input y on controller x alarm
Ix/yS	input y on controller x short circuit
Vnnn	variable nnn (1-256)
Zpp:nnnA	zone pp:nnn (1-528) alarm (and not isolated)
Zpp:nnnF	zone pp:nnn fault (and not isolated)
Zpp:nnnI	zone pp:nnn isolated
Zpp:nnnN	zone pp:nnn normal (not A,I or F)
Tnnn	timer nnn (1-128) state
IRnnn	IO-NET controller responding status
PRnn	Fire Panel responding status

and "reset input" may be any one of the above fundamental operands, i.e., it is a single operand such as I5/I2S or V15 or Z1:42A, etc.

The operand list may not include expressions, i.e., operators NOT, AND, OR, XOR may not be used in an NA function. If it is desired to use an expression in an NA function then the expression should first be assigned to a variable and the variable included in the operand list.

The NA function returns a value of TRUE or FALSE as follows. Whenever any operand in the list has a change of state from FALSE to TRUE then the function returns a value TRUE and continues to return a TRUE value (forever regardless of any further changes of state in the operand list) until the specified reset input has a change of state from FALSE to TRUE. At this point NA returns FALSE and continues to return FALSE until the next change of state from FALSE to TRUE of any operand in the list.

The same operand may appear any number of times in the list.

e.g. NA(I5/I2S; O3N, I7/25A, Z0:124A, V15)

This function is set to TRUE whenever one of the operands in the list becomes TRUE, e.g. if local output 3 changes state from off to on, or if input 25 on controller 7 changes state into alarm, etc. It is set to FALSE when input 12 on controller 5 has a change of state into a short circuit condition.

After power up or restart of the controller the initial state (value returned) by all NA functions is FALSE.

The NA function may be used as often as desired (within the limits of RAM storage) in any logic expression.

e.g. O5 = NA(Z12:42F; I2/1A, I2/2A) OR NA(I1/2S; Z1:52A) OR I3/7A

Each NA function uses some RAM storage in the controller, which sets a limit on the number of new alarm functions which can be used. (A function with 6 or less operands in the operand list uses one byte of RAM storage, 7-14 operands uses 2 bytes of RAM).

If there is a large number of operands in the operand list then multiple NA functions can be used to achieve the same thing.

e.g. V1 = NA(I1/1S ; Z1:1A, Z1:2A)  
O32 = NA(I1/1S ; V1, Z1:3A, Z1:4A)

is the equivalent of

O32 = NA(I1/1S ; Z1:1A, Z1:2A, Z1:3A, Z1:4A)

Multiple reset operands can be achieved by using two NA functions as shown in the following example.

V1 = NOT V1  
V2 = NA(V1; I1/1S, I1/2S, I1/3S)  
O32 = NA(V2; I2/1S, I2/2S, I2/3S)

When any of I1/1S, I1/2S, I1/3S have a transition from FALSE to TRUE, the first NA function becomes TRUE so V2 goes from FALSE to TRUE which will provide a reset to the second NA function. The first NA function is being continually reset on every second pass of the equation (since the value of V1 is continually alternating). However, if any of the operands I1/1S, I1/2S, I1/3S become TRUE the NA function will always become TRUE because the operands in the "set list" override the reset operand.

### 5.3.4.6 System Conditions

- INF** IO-NET network fault  
This returns TRUE if this controller is not regularly receiving input status information from any controller that is required by the output equations or from any controller accessed with the IR operand. If no input status information is required from any other controller and no IR operands are used, then INF will be TRUE if no valid messages of any kind are being received from the network, otherwise INF will be FALSE. INF is prevented from returning TRUE during the first 60 seconds after power-up.
- Each controller transmits its input status information on to the network at regular intervals, but will not necessarily include the information in every message it transmits. The rate at which it transmits input information is set by a network parameter and must be often enough to prevent an INF fault occurring at other controllers requiring that data. The rate at which a controller requires to receive input information is also set by a network parameter (DRQT).
- ZNF** Zone data receive fault.  
This returns TRUE if the controller is not receiving valid data from all of the fire panels that it requires. Valid data must be received at least every 32 seconds from each of the panels being accessed with zone operands Zpp:nnnC (where C is A,I,F or N) or being accessed with PRnn (panel responding) - or ZNF will be set TRUE. After being set true the value of ZNF will be updated every 32 seconds and will be set FALSE if it is found that valid data has been received from all required panels. This includes panels for which zone data is received via the IO-NET network and also any direct connection RZDU data link.
- ZNF is prevented from returning TRUE during the first 60 seconds after power-up.
- ZNF will also be set TRUE if a message is received from the IO-NET network which contains zone data and has a panel number the same as the panel number that has been assigned to the Controller's own RZDU data connection. ZNF will be cleared to FALSE at the next 32 second update time.
- ZNF is always FALSE (except for the situation of duplicated panel numbers just mentioned) if the logic expressions do not access any zone status and no PRnn operands are used - even if the local RZDU data link is enabled (by including the RZDU parameter). I.e. if the local RZDU data link is enabled but no zone status or PR operands are used then ZNF will still always return FALSE even if the local RZDU data link fails.
- If it is necessary to detect this fault then include a PRnn operand (see below) in the program. ZNF is a non-latching value, however, the occurrence of a fault can be latched with an expression  $V1 = V1 \text{ OR } ZNF$ , for example, or by an NA function.
- IB1** Input termination board 1 status. This returns TRUE if the board is present.
- IB2** Input termination board 2 status. This returns TRUE if the board is present.
- OB1** Output termination board 1 status. This returns TRUE if the board is present.
- OB2** Output termination board 2 status. This returns TRUE if the board is present.
- IRnnn** IO-NET controller responding status. This may be used to detect a loss of communications with controller nnn. It is necessary to use a timer in conjunction with the IRnnn operand since IRnnn actually returns a value which OSCILLATES between TRUE and FALSE when controller nnn is responding normally. The value of IRnnn operand is set TRUE whenever a message containing input status data (i.e. the status of the 32 input points on controller nnn) is received from controller nnn, but at periodic intervals of several seconds IRnnn is set FALSE and remains FALSE until the next point status data is received from controller nnn.
- IRnnn is forced to return TRUE for the first 60 seconds after the controller powers up to allow time for communication with the network to be established.

The rate at which controller nnn transmits its point status data is set by a programmable parameter (DRFT) in the control program running in controller nnn when it is in programmed mode and for a controller running in slave mode the rate is one of 2 fixed values (6 seconds for non modem mode and 12 seconds for modem mode - refer LT0115 section 6.1). Also, whenever a change of state on an input point occurs then that controller will immediately transmit its point status data. The IRnnn operand is set FALSE at periodic intervals according to the value of the programmed parameter DRQT and remains FALSE until the next point status data is received from controller nnn. Hence it is necessary to use a timer to detect the condition of IRnnn returning FALSE continuously for longer than a certain length of time - a time of 30 seconds or more is recommended.

As an example, the following logic will result in output 32 turning on if a period of 30 seconds occurred during which no point status data was received from Controller 16. Output 32 will turn off immediately any point status data is received from Controller 16.

```
T1[30,0] = NOT IR16
O32 = T1
```

As a further example, the following logic results in output 32 turning on if, for 30 seconds, no status data is received from Controller 16. The output will stay on, even if point status data is subsequently received, until reset by input I1/1S.

```
T1[30,0] = ^IR16
O32 = T1 + (O32N.^I1/1S)
```

When input I1/1 is put into short circuit, output 32 will turn off if the fault has cleared, i.e. if T1 is false. Output 32 will always be on if the fault is still present (T1 TRUE).

The value of DRQT also determines how often the value of the INF operand is updated and is normally set to about twice as long as the value of DRFT. It is not recommended to reduce the value of DRQT to anything less than the default value supplied in the supplied network profiles.

**PRnn** Fire panel responding status. This is similar to the IRnn operand described above, but may be used to detect a loss of communications with panel nn. It is necessary to use a timer in conjunction with the PRnn operand since PRnn returns a value which OSCILLATES between TRUE and FALSE when panel nn is responding normally. The value of PRnn is set TRUE whenever a valid communication is received from panel nnn (either from the local RZDU input or from zone data received on the network) but at periodic intervals (fixed at 32 seconds) the value of PRnn is set FALSE and remains FALSE until the next valid communication is received from panel nn. Normally one communication (a message) is received from each fire panel every 2 seconds, either from the local RZDU input or from zone data received on the network.

PRnn is forced to return TRUE for the first 60 seconds after power up of the controller to allow time for communication with the network to be established.

As an example, the following logic will result in Output 32 turning on if no message is received from Panel 5 for approximately 60 seconds. The output will turn off as soon as a message is received.

```
T1[60,0] = NOT PR5
O32 = T1
```

The following modification shows how the output can be latched on until reset by input I1/1S.

```
T1[60,0] = ^PR5
O32 = T1 + (O32N.^I1/1S)
```

**5.3.4.7 Constants**

TRUE may be used to return the value TRUE.  
 FALSE may be used to return the value FALSE.

e.g. V1 = FALSE sets variable V1 to FALSE  
 O3 = TRUE turns output 3 on.

**5.3.4.8 Timers**

A timer can be used to detect a particular condition being continuously TRUE or FALSE for a set length of time. Up to 128 timers may be used but may be limited to less than this depending on the RAM storage limits in the controller. An input condition, start delay and stretch delay must be specified for a timer with a statement as follows.

OR Tnnn[xxx,yyy] = input condition  
 Tnnn(xxx,yyy) = input condition

where nnn is the timer number 1 to 128,  
 xxx is the start delay, 0-255 (seconds)  
 yyy is the stretch delay, 0-255 (seconds)  
 and input condition is any valid expression and may include any number of operands, NA functions etc.  
 Either [] or () may be used.

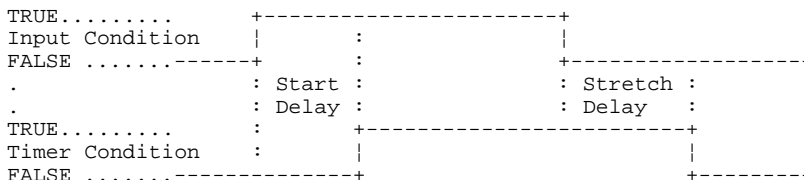
The current "state" of a timer is either TRUE or FALSE and may be accessed in an expression with the operand Tnnn, where nnn is the timer number 1-128.

Timer numbers used in a control program should be used "contiguously", starting with 1, to minimise the usage of RAM in the controller. E.g.. if a program needs two timers, then use timers 1 and 2 - don't use timers 5 and 25 as this will waste RAM memory.

The state of a timer is set to TRUE or FALSE as follows:

If the current state of timer nnn is FALSE, then when the input condition has a transition from FALSE to TRUE, the start delay (xxx) starts. If the input condition remains continuously TRUE for the entire length of time specified by the start delay, then the current state of the timer will be set to TRUE – i.e. Tnnn will return a value of TRUE. The current state of the timer will then remain TRUE until the input condition becomes FALSE (and remains FALSE) for the length of time specified by the stretch delay. Ie. if the current state of timer is TRUE, then when the input condition has a transition from TRUE to FALSE, the stretch delay (yyy) starts and if the input condition remains continuously FALSE for the length of time specified by the stretch delay then the current state of the timer will be set to FALSE - i.e. Tnnn will return a value of FALSE.

When the timer is false, if the input condition does not remain continuously true for the full length of the start delay, then the timer will not become true. Similarly when the timer is true, if the input condition does not remain continuously false for the full length of the stretch delay then the timer will not become false.



**Timer Operation**

As the input condition is evaluated only once every pass through the entire control program, short lived changes in the input condition may not cause a timer to start or reset. The time to execute

one pass of the control program is dependent on the size of the program i.e. the number and size of the expressions it contains. Typically this could be 100 to 300 milliseconds, but could be as long as 500 to 600 milliseconds for large programs.

If the start delay is zero, then when the input condition becomes TRUE the state of the timer will immediately be set TRUE. Note that the state of the timer changes only when the input condition is evaluated, which is once on each pass through the entire control program. I.e. even with a start delay of zero, short lived changes of the input condition may not cause the timer state to become TRUE. The timer state will become TRUE only if the input condition is TRUE at the time the input condition is evaluated.

Similarly, if the stretch delay is zero, then when the input condition becomes FALSE, the state of the timer will immediately be set FALSE, but note again that short lived changes of the input condition may not cause the timer to change state.

At power up, the state of all timers is FALSE.

The state of a timer, Tnnn, may be used anywhere in an expression and the same timer or any number of timers may be accessed any number of times in the same expression. A timer Tnnn, may also be used in the set list of an NA function and may also be used as the reset operand in an NA function.

#### Timer example 1

The following example shows how a timer can be used to give an oscillating output with fixed pulse widths of 30 seconds on and 20 seconds off.

Output 32 will be on for 30 seconds and off for 20 seconds.

```
T4[20,30] = NOT T4
O32 = T4
```

The input condition to timer 4 is NOT T4, hence if T4 is initially FALSE then the input condition will be TRUE, so after 20 seconds the state of the timer will then be set TRUE. This causes the input condition to become FALSE, so the stretch delay (30) starts and after 30 seconds the state of the timer will be set FALSE. The cycle then repeats.

#### Timer Example 2

This example shows the logic required to achieve the equivalent of the "pulse" timer provided on an F3200 fire panel.

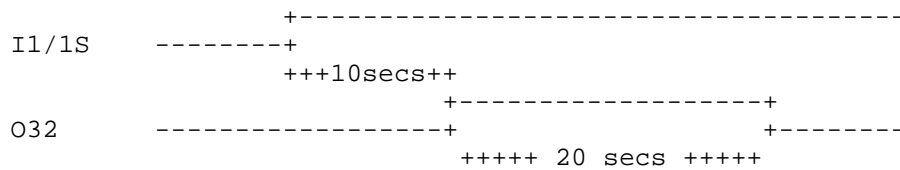
A timer can be used to give a single pulse and then turn off even if the "input condition" remains TRUE. Any input condition can be used - the example uses I1/1S but it may be any expression. When I1/1S has been TRUE continuously for 10 seconds, output 32 will turn on and stay on for exactly 20 seconds, after which it will turn off and stay off even if I1/1S is still TRUE. A transition of I1/1S from FALSE to TRUE is then required to start the process off again. Transitions of I1/1S while the timer is TRUE and output 32 is on are ignored.

```
V1 = I1/1S
V2 = NOT T1
T1[10,20] = NA(V2;V1) AND NOT T1 AND V1
O32 = T1
```

This is a fairly complex example. Suppose the state of the timer is initially FALSE and I1/1S is also FALSE. When I1/1S goes TRUE the NA function will be set TRUE and will remain TRUE until V2 has a transition from FALSE to TRUE (which will occur when timer T1 has a transition from TRUE to FALSE). When the NA function goes TRUE the start delay of the timer will start because the entire input condition is TRUE (V1 is TRUE and "NOT T1" is also TRUE).

After V1 has been TRUE for 10 seconds the state of the timer will become TRUE and output 32 will turn on. Now the input condition to the timer has become FALSE (because "NOT T1" is now

FALSE) so the stretch delay starts and after 20 seconds the state of the timer will become FALSE and output 32 will turn off. The transition of the timer from TRUE to FALSE will reset the NA function so that a new transition of V1 from FALSE to TRUE is then required to start the whole process off again.



The delay before output 32 turns on is set by the start delay and the length of time that output 32 stays on for is set by the stretch delay.

### Timer Example 3 - a Pulse Counter

"Pulse counter logic" may be used to create even longer time delays than the maximum of 255 seconds. The following example uses T5 (Timer 5) as the input condition to count pulses from, but the equation  $V131=T5$  may specify any input condition for which pulses are to be counted, E.g., a circuit input. (If pulses from a circuit input are being counted, short duration pulses may be missed.)

The following pulse counter logic relies on the fact that the equations are executed EXACTLY in the order shown - which is what an IO-NET controller does.

Note also that ideally a program should allocate variable numbers starting from 1 up contiguously. However, using variable numbers as shown makes the logic easier to understand.

```
T5[13,2] = NOT T5
V131 = T5
V132 = (V132 XOR (^V131.V151))
V151 = V131
V133 = (V133 XOR (^V132.V152))
V152 = V132
V134 = (V134 XOR (^V133.V153))
V153 = V133
V135 = (V135 XOR (^V134.V154))
V154 = V134
V136 = (V136 XOR (^V135.V155))
V155 = V135
```

The state of variable V132 will change every 15 seconds (13+2) at the time that timer 5 has a transition from TRUE to FALSE. I.e. V132 oscillates and is TRUE for 15 seconds and FALSE for 15 seconds.

Variable V133 will change every 30 seconds at the time that variable V132 has a transition from TRUE to FALSE.

Similarly, V134 will change state every 60 seconds, V135 every 2 minutes, and so on, with each successive timer doubling the time period.

### Timer Example 4, Long Output Pulses

The pulse counters of Example 3 can be used to generate long time delays. Suppose it is necessary to turn output 32 on for 15 minutes and off for 30 minutes continually. This can be done using the pulse counter equations above but with start and stretch delays in the base timer of 178 and 2. I.e.  $T5[178,2] = NOT T5$ .

This means that variable 132 changes state every 3 minutes, V133 changes every 6 minutes, etc.

The following table shows the state of the 4 variables V132-V135 after 3,6,9,12,15..... minutes.

Time (Minutes)	V132	V133	V134	V135	Binary Value of Counter
0	F	F	F	F	0
3	T	F	F	F	1
6	F	T	F	F	2
9	T	T	F	F	3
12	F	F	T	F	4
15	T	F	T	F	5
18	F	T	T	F	6
21	T	T	T	F	7
24	F	F	F	T	8
27	T	F	F	T	9
30	F	T	F	T	10

With the addition of a reset variable (V130) the pulse counter logic can be used to make Output 32 oscillate on for 15 minutes and off for 30 minutes.

$$T5[178,2] = \text{NOT } T5.V130$$

$$V131 = T5$$

$$V132 = (V132 \text{ XOR } (\wedge V131.V151)).V130$$

$$V151 = V131$$

$$V133 = (V133 \text{ XOR } (\wedge V132.V152)).V130$$

$$V152 = V132$$

$$V134 = (V134 \text{ XOR } (\wedge V133.V153)).V130$$

$$V153 = V133$$

$$V135 = (V135 \text{ XOR } (\wedge V134.V154)).V130$$

$$V154 = V134$$

$$V160 = ((\wedge V132.V133.\wedge V134.V135).\wedge V160) + (V160.\wedge (V132.\wedge V133.V134))$$

$$V130 = \wedge (V160 \text{ XOR } V161)$$

$$V161 = V160$$

$$O32 = V160$$

At startup of the controller, V160 and V161 will both be false so V130 will be set TRUE after the first pass and the pulse counter will start counting. When the state of variables V132-V135 reaches FTFT respectively, (10 counts of 3 minutes = 30 minutes), V160 will become TRUE.

V130 will then immediately be set FALSE for exactly one pass of all equations which will reset the pulse counter. V160 then remains TRUE until the state of variables V132-V134 reaches TFT respectively (15 minutes) at which point V160 will become FALSE and once again V130 will immediately become TRUE for exactly one pass of all equations and reset the counter. Note that the logic works for all cases of the on and off times being less, equal or greater than each other.

#### Timer Example 5 - Long Time Delay Input Monitoring

Suppose it is necessary to detect an input condition being continually true for 15 minutes at which point an output is to turn on and remain on until the input condition has been continually false for 6 minutes.

The following equations do this with the desired input condition being I1/I5. (The input condition could actually be any expression.)

$$T5[178,2] = \text{NOT } T5.V130$$

$$V131 = T5$$

$$V132 = (V132 \text{ XOR } (\wedge V131.V151)).V130$$

$$V151 = V131$$

$$V133 = (V133 \text{ XOR } (\wedge V132.V152)).V130$$



```

V152 = V132
V134 = (V134 XOR (^V133.V153)).V130
V153 = V133
V135 = (V135 XOR (^V134.V154)).V130
V154 = V134
V136 = (V136 XOR (^V135.V155)).V130
V155 = V135
V162 = I1/1S
V160 = ((V132.^V133.V134).^V160) + (V160.^(^V132.V133))
V130 = ^((^V162.^V160) + (V162.V160) + (V160 XOR V161))
V161 = V160
O32 = V160

```

The equation for V130 specifies the conditions which reset the counter. It is important that the length of the stretch timer is a small, non-zero value, otherwise the base timer will not get reset properly when the pulse counter is reset (V130 set false). A value of 2 should always be used for the stretch delay.

The counter is continuously reset while  
 "V160 is false and the input condition is false"  
 OR while  
 "V160 is TRUE and the input condition is also TRUE"

and the counter is reset for exactly one pass following any transition of V160.

An alternative equation for V160 which produces the same result is

$$V160 = ((V132.V134).^V160) + (V160.^(V133)).$$

i.e. it is possible to leave out some of the "least significant" variables which for which the "match condition" is FALSE.

#### Timer Example 6 - Long Time Delay Input Monitoring

The following example of pulse counter timing is a variation of the preceding example. Suppose an output is to turn on after an input condition has been TRUE continually for 15 minutes and stay on for exactly 6 minutes and then turn off and stay off until the input condition has a transition from FALSE to TRUE and the cycle starts again.

```

T5[178,2] = NOT T5.V130
V131 = T5
V132 = (V132 XOR (^V131.V151)).V130
V151 = V131
V133 = (V133 XOR (^V132.V152)).V130
V152 = V132
V134 = (V134 XOR (^V133.V153)).V130
V153 = V133
V135 = (V135 XOR (^V134.V154)).V130
V154 = V134
V136 = (V136 XOR (^V135.V155)).V130
V155 = V135

V162 = I1/1S
V160 = ((V132.^V133.V134).^V160) + (V160.^(^V132.V133))
V163 = ((^V160.V161) + V163).V162
V130 = ^((^V162.^V160) + (V160 XOR V161) + V163)
V161 = V160
O32 = V160

```

V163 is set true when V160 has a transition from TRUE to FALSE (i.e. the point at which the output turns off), and latches TRUE until V162 (the input condition) becomes FALSE.

The equation for V130 specifies the conditions which reset the counter, i.e. the counter is reset when:

"the input condition is FALSE and V160 is FALSE"  
OR while  
"V163 is true"  
and the counter is reset for exactly one pass following any transition of V160.

#### Timing Accuracy

For time delays requiring greater accuracy/resolution, smaller values of the start and stretch delays in the base timer of the pulse counter can be used but more equations and variables will be needed to achieve longer times. For accuracy, it is important that the stretch delay value in the base timer is a small non zero value - the value 2 is recommended. Also, the start delay should always be a non zero value - i.e. do not have a value of zero for the start delay.

There is an inaccuracy in the base rate of a timer. Due to the way the time is measured, the controller decrements every timer at a nominal rate of once every 1.008 seconds (1108 milliseconds) and this is also subject to a variation in the base crystal oscillator which is approximately 0.5% (half a percent).

Hence, with increasing a time delay, a variation of +1.3 percent to +.3 percent can occur. In a 24 hour period this can amount to 15 to 20 minutes.

Even with start and stretch delays both non-zero, there is potential for variation in the repeat rate of a time, e.g. T5[1,1] = NOT T5, will repeat exactly every 2 seconds in most cases except for programs where the time to execute one pass of the output logic is approaching 1 second or more.

For all but extremely large programs, the execution time of one pass of the output logic is normally less than 500 milliseconds.

If a timer is used to drive a pulse counter, any inaccuracy in the base timer has a multiplying effect through the pulse counter.

## 5.4 Change Log

The change log is a record of changes made to an IO-NET program. When you save a file, you will be prompted to fill in a summary of the changes you have made, and this will be maintained in the Change Log.

## 5.5 Notes

Enter any text you like here. It is for documentation and notes on operation of the system.

This information is not stored in the IO-NET controller, so it will not be extracted when [Extracting a program from an IO-NET Controller](#)<sup>55</sup>. Therefore, and as a backup, it is essential that you maintain a copy of the SmartConfig file.

## 5.6 Network Profiles

The network profiles for various baud rates, and with and without modems, are prepared by Tyco Safety Products and supplied with SmartConfig. They should not need changing or adding to by a user.

More information on the parameters is in LT0115 VIGILANT IO-NET PROGRAMMABLE CONTROLLER USER'S MANUAL.

## 5.7 Example Programs

Some example programs are included here.

See the next few pages (press next at the top of the page, or expand the table of contents), for more details.

### 5.7.1 Duplicating Non-Programmed mode

The following program duplicates the operation of non-programmed mode if it was running in a controller with address one.

```
O1 = I1/1A OR I1/1S
O2 = I1/2A OR I1/2S
:   :   :
O32 = I1/32A OR I1/32S
```

### 5.7.2 Passing Data onto the IO-Net network

Each controller transmits the state of all its inputs onto the IO-NET network. By wiring a local output to one of its own inputs it is possible for a controller to effectively pass output information onto the network for use by other controllers. This could include error conditions E.g.. INF or ZNF, or perhaps even the state of one of its own outputs.

With an output wired directly to an input, when the output is on, the state of the input will be EITHER alarm or short circuit (with low current pull-up it will definitely be short circuit but with high current pull-up it may be either alarm or short circuit), and when the output is off, the state of the input will be either normal or open circuit depending on whether an EOL resistor is connected or not.

E.g. If the controller at address 8 has output 32 wired directly to input 32 and included the statement O32 = INF OR ZNF, then if some other controller (say 5) included the statement

```
O1 = I8/32A OR I8/32S
```

then output 1 (on controller 5) would be on when controller 8 had a zone network fault (ZNF) or an IO-NET network fault (INF).

### 5.7.3 Accessing Zones

The following are simple examples of output logic using fire panel zone data.

```
O1 = Z3:1A ; output 1 on if zone 1 on panel 3 is in alarm
```

```
V1 = Z0:2A OR Z0:3A OR Z0:4A
```

```
O2 = V1 ; output 2 on if alarm on zones 2, 3 or 4 of panel 0.
```

```
O3 = NA (I1/1S; Z1:2A, Z1:3A, Z1:4A)
```

```
; output 3 on if any new alarm on zones 2, 3 or 4 of panel 1, latched until reset by short circuit on input 1 on controller 1.
```

```
O4 = Z2:156F OR Z3:157F OR V1
```

```
; output 4 on if fault on zone 2:156 or zone 3:157 or if
```

```
; alarm on zones 2, 3 or 4 of panel 0 (from expression for V1 above)
```

## 5.8 Checking a Program

Use the command Check | Check Output Logic.

This checks the logic without attempting to connect to an IO-NET programmer.

You may need to adjust the amount of available RAM in the Parameters page, Off-Line RAM assumptions. See [RAM Assumptions](#) <sup>[35]</sup>.

### Error Messages

#### Range Error

This occurs when an out of range numerical value is used. Examples are use of a controller circuit input number of 33 when the allowed range is 1 to 32 or a variable number of 0 when the range is 1 to 256.

#### Syntax Error

This covers many different types of error such as illegal characters or unknown words or brackets or operators missing in an expression.

#### Target Stack Error

This error is unlikely to occur but will if more than 16 levels of stack space are required at run-time in the IO-NET Controller to evaluate an expression. The compiler detects this and indicates an error.

E.g.  $O1 = V1 \text{ OR } (V3 \text{ OR } (V4 \dots$  with 17 levels of nested brackets would exceed this limit.

To overcome the error, reduce the number of brackets or separate the expression into two parts using a variable.

#### Missing Network Parameters

**Missing RZDP Parameter for Protocol Type**  
**Parameters must precede expressions**  
**Missing RZDU parameter**

These errors should now never occur. Contact Tyco Safety Products if they do.

#### Cant access own address with IR

The IR operand is not allowed to specify a controller number which is the same as the Address assigned on the Parameters Page.

#### No END Statement in Source File

An END statement must be included at the end of the Output Logic.

#### WARNING multiple expressions for the same output

If the same output is controlled by more than one expression then a warning message is displayed.

e.g.  $O12 = Z1A$   
 $O12 = Z1F$

is an example where output 12 is controlled by more than one equation and will result in a potentially intermittent and indeterminate output state. This warning will not prevent programming of the controller but, for normal operation, it should not be done.

#### WARNING multiple expressions for the same timer

If the same timer is controlled by more than one equation then a warning message is displayed.

e.g.  $T1[15,30] = I1/1S$   
 $T1[15,30] = I2/1S$

Control of a timer by more than one equation will give indeterminate operation and should not be done. Variables can be used if the input condition for a timer is long and complex.

**Base Page RAM Limit Exceeded**

If this error occurs it means that the output logic compiled with no syntax errors but that the total number of variables, NA functions, timers and remote IO-NET controllers accessed exceeded available base page RAM space in the IO-NET controller.

If the target Controller was not on line when the source file was compiled, then the logic should be re-compiled with the target Controller on line. (Refer to [Programming an IO-NET Controller](#))<sup>[54]</sup>.

If this error still occurs when the logic is compiled with the target Controller on line, then it is necessary to reduce the usage of:

1. The range of variables used. Always start with variable V1 and use variables contiguously (without any "gaps"). E.g. if you require 3 variables, use V1, V2, V3 - do not use V1, V100, V256.
2. NA functions or the number of operands in an NA function.
3. The range of IO-NET Controller numbers accessed in the control program. The difference between the highest and lowest controller number accessed determines the amount of base page RAM required. So, if possible, Controllers should be accessed contiguously but the lowest Controller accessed does not have to be 0.
4. The range of timers used. Always start with timer T1 and use timers contiguously.
5. The RANGE of fire panels accessed in zone operands - the difference between the highest and lowest fire panel number determines the amount of base page RAM required. Use consecutive panel numbers if possible.

Note, it does require fairly heavy use of variables, NA functions, etc., to exceed the base page RAM limit.

E.g. if the highest and lowest IO-NET Controller numbers accessed were 31 and 0, and all 256 variables (V1 to V256) were used, then approximately 144 NA function operands can be used (E.g. 24 NA functions with 6 operands each or 12 NA functions with 14 operands each). An NA function with 7 to 14 operands uses 2 bytes of RAM.

**Total RAM Limit Exceeded**

If this error occurs it means that the source file compiled with no syntax errors but the total amount of RAM required in the IO-NET Controller for storing zone data, IO-NET Controller point status, variables, NA functions, timers and RZDU receive buffer, if any, exceeded the available RAM.

If the problem occurred when checking a program, enter the correct numbers under "RAM Assumptions" (refer to [Off-line RAM Assumptions](#))<sup>[35]</sup>, or check compiling with the target Controller on line. (Refer to [Programming an IO-NET Controller](#))<sup>[54]</sup>.

If this error still occurs when the source file is compiled with the target Controller on line then tick the box [Allow reduced RAM](#)<sup>[34]</sup>, or reduce the usage of:

1. The range of variables used, the number of NA functions and NA function operands, the range or number of timers used, and the range of IO-NET Controller numbers accessed as discussed under the "Base Page RAM Limit" error above.
2. If any local circuit inputs are accessed on this Controller then specify the address of the target Controller. (Refer to [Address](#))<sup>[32]</sup>. If the address is specified then the number of circuit inputs on the local Controller which are accessed makes no difference to the amount of RAM required.
3. Reduce or re-arrange the number of circuit inputs accessed on remote IO-NET controllers. IO-NET controller circuit inputs are stored in groups of four. E.g. 1, 2, 3, 4 are stored as one group, 5, 6, 7, 8 are next, etc, and it is possible that reducing the number of "groups" which are accessed can reduce the required RAM storage. E.g. if 3 circuits are accessed

on controller 15 try to make them all from the same group such as circuits 1, 2, 3, rather than circuits from three different groups.

4. Reduce or rearrange the number of zones accessed. Less RAM is required if zones are contiguous rather than scattered through a wide range. E.g. Accessing zones 1, 2, 3, 4, 5, 6, 7, 8 requires less RAM than accessing zones 10, 20, 30, 40, 50, 60, 70 and 80.

## 5.9 Programming an IO-Net Controller

To program an IO-NET controller, or check the free space for a new program, use the command File | Download to / Upload from Panel.

You will be given the following options. Apart from "Select Comms Port" they all require the IO-NET programmer, with the proposed IO-NET Controller IC inserted, to be connected to the selected Comms port on your PC.

Note to check your program without downloading it and without being connected to an IO-NET programmer, see [Checking a Program](#)<sup>[52]</sup>.

### Compile Only

This compiles your program "online". I.e. it as well as checking for syntax errors it also checks the program will fit into the Controller in the programmer. For an explanation of possible error messages, refer to [Checking a Program](#)<sup>[52]</sup>.

### Compile & Program Chip

This compiles your program, and if there are no errors and it will fit into the Controller in the IO-NET programmer, downloads the program to the Controller in the programmer.

### Compile & Verify against Chip.

This compiles your program and verifies it against the contents of the Controller in the IO-NET programmer.

### Retrieve Space Info

This shows details of the Controller in the IO-NET programmer, including the amount of free EPROM.

### Extract Latest Program from Chip

This will allow you to extract the latest program from an IO-NET controller. (See [Extracting a program from an IO-NET Controller](#))<sup>[55]</sup>

### Extract Older Program from Chip

This will allow you to extract a selected program (not necessarily the latest) from an IO-NET controller. (See [Extracting a program from an IO-NET Controller](#))<sup>[55]</sup>

### Select Comms Port

This enables you to select the comms port on your PC that connects to the IO-NET programmer. The port you select will be remembered across SmartConfig sessions.

### LEDs on Programmer Board

When your PC is reading data from the Controller in the programmer, both the Verified and Programming LEDs will be on. When your PC is writing data to the Controller in the programmer, only the Programming LED will be on.

### Error Messages

#### Chip Not Responding

The PC cannot communicate with the Controller in the Programming Module. Check that the correct serial port on the PC is used, the programmer is connected, the IO-NET Controller is

correctly installed in the programmer, power is applied, and the Reset switch is in the RUN position.

Switch the programmer module Reset switch to RESET then back to RUN and try again.

If the problem persists, switch the programmer to RESET, turn the programmer power switch OFF, switch the power switch back ON, switch the reset switch to RUN and try the command again.

#### **Chip Checksum Read Error**

This indicates that a response of some sort was received from the programmer module but there was an error in the data received (perhaps due to noise on the serial line). Power the programmer module OFF and then ON and try again.

#### **Invalid Data Read From Chip**

This error indicates that the Controller chip is not valid or of an unsupported software version. SmartConfig can program only controller V2.00 version or later.

#### **Unable to Program - EPROM is Full**

This indicates that there is not enough space left in the EPROM to program the source file. The Retrieve command can be used to show how much free space is remaining in an EPROM (in bytes).

The Compile command also indicates the size (in bytes) of the object data produced when the source file is compiled.

The Controller should either be thrown out or could be used in non-programmed mode. Also it is possible that the Controller could be used with a smaller source file.

#### **Error Reading File DUMPMEM.BIN**

#### **Error Reading File PROGSBC8.BIN**

#### **Error Reading File PROGSBC9.BIN**

All these files are installed with Smartconfig. If you get these messages, one of them must have been accidentally deleted. Re-install SmartConfig.

#### **Failure During Programming**

This error occurs if something goes wrong during the programming of a source file into the Controller. Use the Verify and Retrieve command to check if the Controller has been programmed correctly. If the Verify fails then it may be possible to try the programming operation again or it may be necessary to throw the Controller away.

#### **RAM Size Mismatch Error**

This error can occur during programming an Controller if the source file was compiled with a different target Controller on line than the Controller currently being programmed. The source file should be re-compiled with the target Controller on line.

## **5.10 Extracting a program from an IO-Net Controller**

To extract a program from an IO-NET controller, use the menu command File | Upload new from Panel | IONet. (Or if you already have an IO-NET file open, you can use File | Download to / Upload from Panel).

Then to extract the **latest** program in the Controller, click "Extract Latest Program from chip".

You can also extract an older program from the Controller, i.e., one that has been replaced by a newer version. Click "Extract Older Program from chip". You will then need to enter the program number to extract, where 1 is the first program that was ever programmed into the chip, 2 was the second program that was ever programmed into the chip, and so on. You will be advised the latest program number, so if you want the second latest for example, enter a number one less than the latest.

Every program you extract will create a new window in SmartConfig with a name such as NONAME1, NONAME2, etc. You can access these windows with the window menu.

Note 1 - when you extract the latest program from a chip, the CRC of the program will be checked against the chip contents including the remaining free space. When you extract an older program, the CRC will be checked against the older program plus the assumed erased state for what was the remaining free space at the time. It can therefore be much quicker to extract a selected program number than the latest program (even if the selected program number is the latest).

Note 2 - when extracting, the following items on the Parameters page are not extracted - [Address](#) <sup>[32]</sup>, [Allow reduced RAM](#) <sup>[34]</sup>, [Off-line RAM Assumptions](#) <sup>[35]</sup>. You may need to change these to suitable values if re-compiling the extracted program fails due to insufficient RAM.

### **LEDs on Programmer Board**

When your PC is reading data from the Controller in the programmer, both the Verified and Programming LEDs will be on.



## 6 Whats New

This section describes the changes with each new version of SmartConfig.

### 6.1 Version 1.6.2

#### FP1600

- The default panel firmware version (with File | New) has been changed from V4.xx to V5.xx.

### 6.2 Version 1.6.1

#### General

- When a file had a very long name with no spaces, attempting to close it when the data had been changed gave a garbled display in the dialog box asking whether to save the file. This has been fixed.
- If you closed the window that was active the first time you used File | PDF, then subsequent attempts to PDF failed. This has been fixed.
- The position and size of the terminal window is now saved across SmartConfig sessions.

#### FP1600

- Upload from a FP1600 panel to SmartConfig and download from SmartConfig to a FP1600 panel failed in version 1.6.0. This has been fixed.

### 6.3 Version 1.6

- **Support for IO-Net introduced.**
- Two lines of text per cell are now displayed in table view, so that long entries are now much more likely to be displayed in full in this view.
- If the change log gets full the oldest entries are now automatically removed.

### 6.4 Version 1.4

#### Installer

- The installer now optionally adds an Icon to the "Quick Launch" area of the taskbar.
- When installing this version of SmartConfig, removing or overwriting any older version of is now optional, so that new and old versions can easily be run in parallel. Each new version will now have separate shortcuts and separate entries in the Control Panel "Add / Remove programs" function. However bug-fix releases by default will continue to overwrite the older version (e.g. 1.4.1 will overwrite 1.4.0).
- The uninstall function (invoked either by the "Start Menu", or "Add / Remove programs") now gives the option of not removing SmartConfig's licence, so uninstalling this version some time after installing a newer version, will not necessarily wipe the licence.
- Existing templates that could have been modified and will be overwritten are backed up under a "SavedTemplates" folder in the installation folder.
- If an installation folder was selected using the "Browse" button, "\SmartConfig" was previously appended to the chosen folder. Now the chosen folder is used as is.
- It now removes any duplicate "check expiry on startup" task possibly left by a previous installation.

#### Data Files location

The default location for data files was previously the same folder as SmartConfig was installed into, typically "Program Files\Vigilant\SmartConfig". However now each version of SmartConfig is

by default installed into a separate folder within the above folder, and in this case the default folder for data files is now the parent folder of the installation folder, so that previous data files can easily be located.

However this default location can now be changed with a new menu command "File | Select Default Data File Location". You may want to use this command to make the default location something under "My Documents".

## Terminal Window

The terminal window has been improved:

- It can be used as a standalone terminal without an open database.
- Common Vigilant products can be selected by name so that it is not necessary to remember the baud rate and other settings for each product.
- It now has a 1000 line scrollbar buffer.
- It now has copy and print commands.
- The "send" command has been split into "send file" (which sends an arbitrary file) and "send database" (which sends the current database).
- The "capture" command has been split into "capture file" (which captures to an arbitrary file) and "capture database" (which extracts the database in the panel, into a new SmartConfig Window).
- Connection over TCP/IP is now possible. NOTE : this has had limited testing over wireless and other slow connections and its operation is not guaranteed. Further information will be supplied as it becomes available. See Terminal Window in the on-line help file for more information.

## Help

- The Help file has been split into individual products to make searching easier and reduce confusion.
- The help menu now includes a "What's New" section that describes the changes from the previous versions.

## Templates

- Supports multiple templates for each product. Multiple choices are included for each product under the File | New command based on the available templates found in SmartConfig's installation folder. Also when saving the file you are given the choice of creating a new template by using the template extension.

## Colours

- "Greyed out" cells were not displaying properly with the Windows XP "Silver" colour scheme - this has been fixed.

## Zone Labels

- Zone LED Labels of the required sizes and spacing can be printed for MX1 and MX4428.

## Dual Monitors

- Dialog boxes will now generally appear on the monitor you are using.
- On startup if the saved position of the main window is off the screen (due to a second monitor being no longer present), a default position will be used instead.

## Dialog Boxes

- Dialog boxes have been improved with the use of customised text on each button, instead of asking convoluted questions to use the standard buttons : OK, Yes, No, Cancel.

## Windows disappearing

- There was a problem with Windows XP re-arranging the display order of SmartConfig's windows when switching to another application and then back to SmartConfig. For example, if the terminal window was on top of the main window and you switched to another application and back to SmartConfig, the terminal window could disappear behind the main window. This

problem has been fixed.

## 6.5 Version 1.3

- PDF Command added.
- You can now cancel a save from the Change Log entry form.
- With the “Fill series” command if the increment was not 1 and the number in the second cell was preceded by letters, it now calculates the increment properly. (E.g. Point 1, Point 3 in the first two cells now continues with Point 5, Point 7)

# Index

## - A -

About 12  
 Accessing Zones 51  
 ADDR 32  
 Address 32  
 Allow reduced RAM 34  
 Ancil Config (FP1600) 26  
 Ancil Map Variables (FP1600) 27  
 As Form 9  
 As Table 9

## - B -

Back 9

## - C -

Change Log (FP1600) 29  
 Check 11  
 Check O/P Logic 11  
 Check Tables 11  
 Checking a Program 52  
 Checks (FP1600) 30  
 Close 6  
 Comments (FP1600) 29  
 Communicating with the FP1600 Panel 29  
 Compare 6  
 Compile & Program Chip 54  
 Compile & Verify against Chip 54  
 Compile Only 54  
 Constants 45  
 Copy 8  
 Copy with Header Row & Fixed Cols 8  
 CRC 11  
 Cut 8

## - D -

Database Converter 12  
 Default Folder for Load / Save 6

Delete 8  
 Download to / Upload from Panel 6  
 Download to / Upload from Panel (FP1600) 29  
 Duplicating Non-programmed mode 51

## - E -

Edit 8  
 Edit Cell 8  
 Error Messages 52, 54  
 Example Programs 51  
 Exit 6  
 Export 6  
 Extract Latest Program from Chip 54  
 Extract Older Program from Chip 54  
 Extracting a program from an IONet Controller 55

## - F -

F3200 / MX4428 Zone Data 40  
 File 6  
 Fill Down 8  
 Fill Series 8  
 Filter... 9  
 Find 9, 13  
 Find Next 9  
 Firmware, Programming 12  
 Form View 14  
 FP1600 Getting Started 21

## - G -

General Editing Instructions 13  
 General Format 35  
 Getting Started (FP1600) 21  
 Go to Related Table 9  
 Group Config (FP1600) 25  
 Group Map Ancil (FP1600) 25  
 Group Map Variables (FP1600) 26

## - H -

Help 12

**- I -**

Info 11  
 IO-Net Board Inputs 39  
 IO-Net Board Outputs 40

**- L -**

LCD Protocol 33  
 LED Protocol 33  
 LEDs on Programmer board 54  
 Licence 11  
 Licensing 15  
 Loading Files 31

**- M -**

Max Stations 32  
 Move Column Left 9  
 Move Column Right 9  
 MRAM 34  
 MXST 32

**- N -**

New 6  
 New Alarm Function 41  
 Next 13  
 Notes 50  
 Notes column in tables 14

**- O -**

Open 6  
 Operand Types 39  
 Operators 37  
 Order of Evaluation 38  
 Output Logic 35  
 Overview (FP1600) 19

**- P -**

Passing data onto the IO-Net network 51  
 Paste 8  
 PDF 6

Previous 13  
 Print All 6  
 Print Table 6  
 Program Firmware 12  
 Programming an IO-Net Controller 54

**- R -**

Renew 11  
 Retrieve Space Info 54  
 RZDP 33  
 RZDU 32  
 RZDU Number 32  
 RZDU Protocol 33

**- S -**

Save 6  
 Select 9  
 Select All 8  
 Select Comms Port 54  
 Show All Rows/Unsort 9  
 Show CRC 11  
 Show Used Rows 9  
 SmartConfig Help 12  
 Sort Ascending 9  
 Sort Descending 9  
 Status Display 13  
 System Conditions 43  
 System Config (FP1600) 28

**- T -**

Table view 13  
 Template Files 14  
 Terminal Window (File Menu) 6  
 Terminal Window (General) 16  
 Terminate 11  
 Timers 45  
 Tools 12  
 TX Zone Data 33  
 TXZD 33

**- U -**

Undo 8

Upload New from Panel 6

## **- V -**

Variables 40

Variables Notes (FP1600) 28

Version (FP1600) 21

View 9

## **- W -**

Window 11

## **- Z -**

Zone Data 40

Zone Map Ancil (FP1600) 23

Zone Map Groups (FP1600) 23

Zone Map Variables (FP1600) 24

Zones (FP1600) 22