CHAMELEON

PROGRAMMABLE CONVERTER

Table of Contents

1.0 Unpacking the Instrument

The instrument comes with the following items:

1 Programmable Function Converter.
1 User Guide.
1 Set of connectors.

The mini manual contains a full hardware specification, and connection details, and a list of all the functions. This provides all the information required for wiring purposes. The manual is designed to complement the full manual provided in the starter pack, but not replace it.

The connectors are standard 2 part connectors. Thus allowing the instrument to be removed without disturbing the wiring.

It is advisable to program the instrument, ensuring that all ranges have been configured before connecting any inputs or outputs to the unit.

2.0 The Chameleon Package

The full Chameleon package contains all the accessories to facilitate using the unit straight away. The contents are as follows:

1 Manual.
1 IBM PC communications cable.
1 PC hosted communications/spreadsheet program.
1 User Guide.

The cable supplied will allow you to connect to the majority of IBM PC type computers. The cable is fitted with a female 25-way D-type connector which can be plugged into the COM port on the PC if it is fitted with a male 25-way D-type connector. However, if your PC is fitted with a 9-way D-type connector, you will require a 25 pin to 9 pin adapter.

If you wish to connect the instrument to a PSION organiser another adapter will be required. This is a 25-way male D-type to a 25-way male D-type (i.e. both ends of the adapter are the same) with the lines 2 and 3 swapped over at one end.

To connect to any other dumb terminal an adapter described for the PSION will probably be required. Please refer to Section 6.0 Communications, for further technical details.

The PC software can be run from either floppy or hard disc. Full instructions are available in Section 8.0 PC Software. Please make a working copy of the program, as described in Section 8.0, and carefully store the Master Disc.

2.1 Using the Unit for the First Time.

If you are unfamiliar with the instrument, it is advised that you read the entire manual. Before connecting any power to the device, ensure that the power supply is correctly rated (24V d.c.), and that no inputs or outputs are connected until the instrument has been configured to the appropriate ranges (voltage or current).

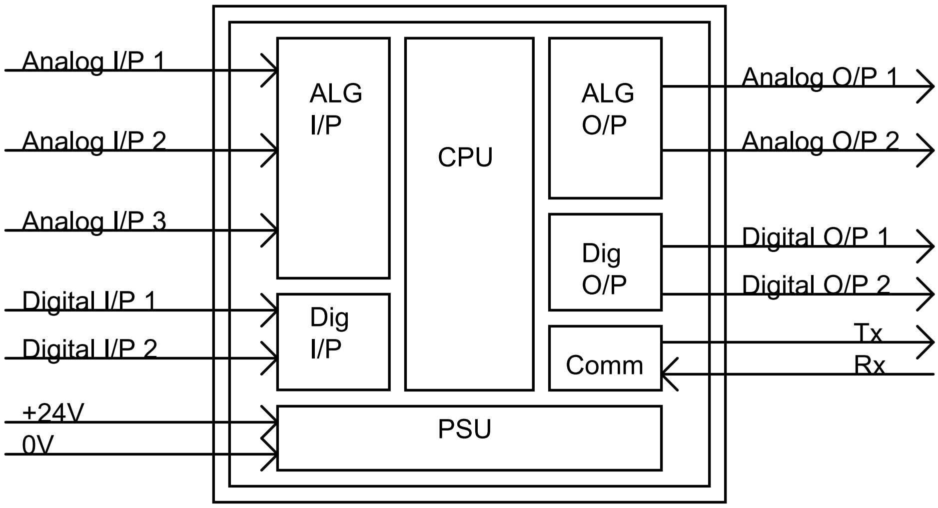
The instrument is programmed using a Spreadsheet. If you are unfamiliar with spreadsheets, then **Section 7.0 What is a Spreadsheet** gives a full description. This section also describes how the spreadsheet is implemented on the instrument, and how it differs from conventional spreadsheets such as LOTUS 1-2-3^{tm}.

If you are using the unit with an IBM PC or compatible computer, **Section 8.0** describes using the PC software. The software is a very good way to become familiar with the spreadsheet concept.

If you plan to use the instrument as a stand alone unit connected to a dumb terminal such as a Psion Organiser, then **Chapter 9.0 The Direct Communications Link** describes how to connect and program it using dumb terminals.

3.0 Introduction

The instrument is a microprocessor based signal conditioning unit which allows the linearisation and conversion of multiple input channels. The instrument has 3 analogue input channels, 2 digital input channels and 2 digital and 2 analogue output channels. The unit has a wide range of computational functions which are user-selected to generate the required outputs. The device can be factory set for dedicated applications or be configured by the end user. This configuration can be done using a variety of terminals, examples being any IBM PC compatible, a PSION II organiser, or any dumb terminal. The configuration is fully menu-driven with a spreadsheet style format, allowing the instrument to be very quickly and easily tailored to each application.



In the simplest configuration the unit could be used to linearise a single analogue input, invert it, re-transmit it in standard form and provide high and low alarm relay outputs. More complex operations are also possible. For example, the instrument could accept an input from a differential pressure transducer, extract the square root, linearise the result according to a second input (temperature for instance) and output this in both pulse and analogue outputs proportional to flow. A relay output would also be available for either a high or low alarm output. In addition, all this information can be relayed by RS485 or RS232 data communications channels for remote monitoring and configuration.

The functions are pre-programmed and structured so that they may be combined with others to provide a very powerful and flexible solution to process computing problems.

The instrument is housed in an ultra-compact custom enclosure which allows the device to be DIN-rail mounted.

4.0 Hardware Features

The unit features the following input/output capability:

3 Analogue Inputs

Current Ranges: 0-20mA, 4-20mA,
Voltage Ranges: 0-10V.

2 Analogue Outputs

Current Ranges: 0-20mA, 4-20mA,
Voltage Ranges: 0-10V.

2 Digital Inputs

Types: Logic Status, Frequency.

2 Digital Outputs

Types: Logic Status, Pulse Frequency.

Communication Channels.

Types: 1 RS232 or optional RS485 channel (address configurable).
The full hardware specification is included in Section 12.0.

4.1 Program Storage.

The instrument is fitted with a semiconductor non-volatile memory which does not rely on battery power. This will store the spreadsheet and configuration information for an unlimited period of time.

4.2 Status LED.

The instrument has a status LED which gives information on the current status of the instrument using different flashing modes.

1) Normal Healthy Programmed Operation.

Short Flash - Long Space:

...*..........*..........*....
Flash every 2 seconds.

2) Normal Healthy NO Program.

Long Flash - Short Space:

..******..*********..***
Space every 2 seconds.

3) Healthy Programmed, BUT Program NOT Saved

(Program WILL be lost if power fails)
Medium Flash - Medium Space:

*****.....*****.....*****.....
1 second flash, 1 second space.

4) Error Condition - Memory didn't store program correctly.

Short Flash - Short Space:

..*.*.*.*.*.*.*.*.*.*.*.*.*.*.
3 Flashes per second.

(Enter Edit mode, look at the contents of a cell and quit to main menu this will force another save to memory. Repeat this if the error condition persists. Or in terminal mode type SAVE at the main menu prompt. - See Section 9.0 for details.)

5.0 Software Features

The instrument has been designed to allow the user to program it easily and quickly, with the minimum amount of additional equipment.

All the software required to program the unit is contained within the unit. The only additional equipment that is required is a dumb terminal. e.g. PSION Organiser with communications pack, VT100, IBM PC with terminal program or the PC programming software.

The device has a full-featured spreadsheet stored within it. Using standard spreadsheet functions and the added speciality functions, it is possible to design a vast range of input/output functions on both digital and analogue signals and a mixture of them both.

The special functions include ones to output analogue values, interrogate the analogue inputs, and to input and output digital signals in a variety of formats. Also included is a PID function which can be used to form two standard PID control loops; a range of TIMER functions to help produce pulse width modulated signals, pulse outputs or simply to time a process; an INTEGRATOR function to integrate the analogue inputs; a PULSE function to produce pulses in conjunction with the integrator; a RATE LIMITING function to set the maximum rate of change of a variable; an INTERPOLATION function to help produce look-up tables; a SAMPLE function to sample values only when a specified condition is true; and SCALING and CLIPPING functions to help ensure that values are kept to meaningful ranges.

In addition, the device is now provided with a program for an IBM PC computer which provides the user with a simple method of programming, showing the spreadsheet in a standard format. The software allows the user to save the spreadsheets, test their functionality to a limited extent, read the program from the unit and access the standard in-built menu structure.

6.0 Communications

The instrument has been equipped with a serial communications port which will produce a standard RS232 signal when used with the special communications lead. In addition it is possible to specify the RS485 communications option. This will produce the RS485 signals ready for wiring into a standard RS485 multi-drop system.

The communications link is made through the 10 way IDC connector on the front panel.

In RS485 mode the unit is ideally suited to multi-drop applications with up to 32 units in any one loop. Each unit can have its address set from 1 to 32 accordingly.

In this way very complex distributed control systems can easily be implemented with the knowledge that there is an 'intelligent' unit at each node. The unit can also be connected in an RS485 multi-drop communications network alongside many other manufacturers' units for a versatile network configurable to almost any requirement.

6.1 Communications Protocols

The communications channels use the following protocols.

Baud Rate:	9600 Baud
Parity:	None
Bits:	8
Stop Bits:	1
Start Bits:	1
Handshake:	None
Echo:	Local

End of line character: <CR> ASCII value 13 (Hex 0D)

Most communications programs would show the above information in the following format: 9600 8N1 (i.e. 9600 Baud, 8 bits, No parity, 1 Stop Bit).
Character translation may be required for the end of line character.

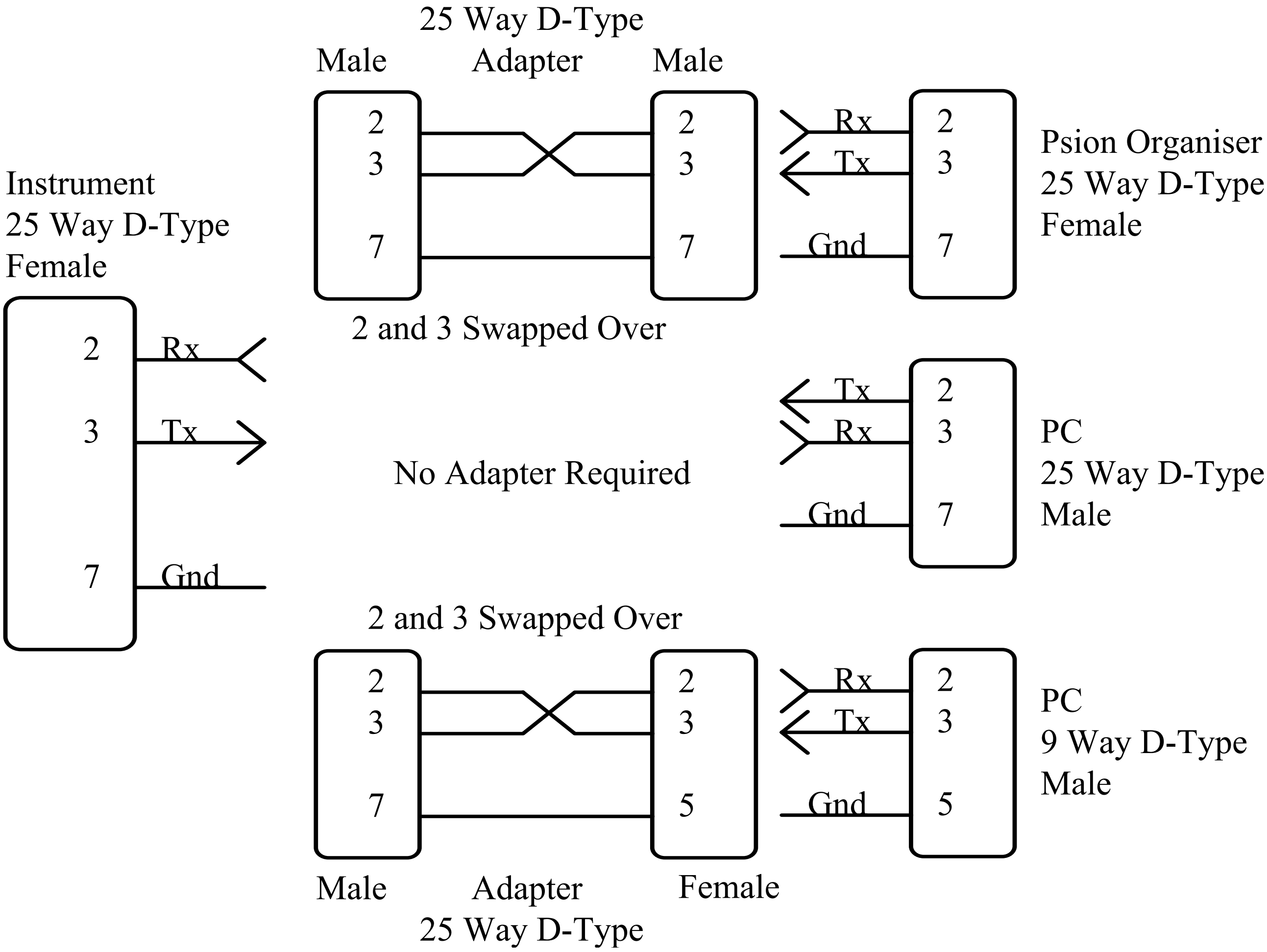
On the PSION organiser the following must be set:

Baud	9600
Parity	None
Bits	8
Stop	1
Hand	None
Protocol	None
Echo	LOCAL
Width	20
Timeout	None
Reol	<CR>
Reof	None
Rtrn	None
Teol	<CR>
Teof	None
Ttrn	None

Please refer to the PSION Communications Manual for details on how to change the above values.

6.2 Hardware.

Depending on the type of equipment that the unit is connected to, the 25 way D-type connector at the end of the communications lead may have to be adapted. The cable only uses 3 wires: the transmit signal, the receive signal and the signal ground. These are shown below:



The standard arrangement allows direct connection to the PC 25-way male D-type connector.

If you wish to connect the instrument to a PSION organiser another adapter will be required. This is a 25-way male D-type to a 25-way male D-type (i.e. both ends of the adapter are the same) with the lines 2 and 3 swapped over at one end.

To connect to any other dumb terminal an adapter described for the PSION will probably be required. The basic rule is that the Rx line on the instrument connector (pin 2) is connected to the terminal's Tx line, and the Tx line on the instrument connector (pin 3) is connected to the terminal's Rx line. Also the signal ground on both the instrument and the Terminal are connected.

7.0 What is a Spreadsheet

7.1 A Traditional Spreadsheet

Spreadsheets are used to manipulate data in an easy to understand fashion. Originally, a spreadsheet was literally a piece of ruled paper (usually ledger paper) on which columns of numbers could be written. Obviously headings would be added so that the sheet could be understood. A typical usage would be for cash flow analysis:

	Cash flow Forecast for Widgets Ltd		
	2/1/1900		
	Jan	Feb	March
Sales	1400	1500	1600
Expenses:	190	250	300
Salaries	250	250	250
Rent	20	20	20

When spreadsheets were computerised, the screen was divided into boxes called cells. The user could type either text or numbers into each cell, just as was done on paper. The computer version was extended to allow formulae to be entered as well. This then took much of the hard work out of checking calculations, or finding errors.

Using our cash flow example, it would be typical to add a 'Totals' column at the end. In here a summation formula would be placed for each row:

	A	B	C	D	E
1	Cash flow Forecast for Widgets Ltd				
2	2/1/1900				
3		Jan	Feb	Mar	Total
4	Sales	1400	1500	1600	B4+C4+D4 (4500)
5					
6	Expenses	190	250	300	B6+C6+D6 (740)
7	Salaries	250	250	250	B7+C7+D7 (750)
8	Rent	20	20	20	B8+C8+D8 (60)

The final column also shows figures in brackets. This is because when a formula is typed in, the computer calculates the result of the formula and stores it. If required, the display can be changed to show all the calculated values. So on our example the computer would remove the formula B4+C4+D4 from cell E4 and put the result, 4500 on the screen.

The clever thing about the computer format, was that if it was decided to change the sales in Feb to 1600, as soon as 1600 had been typed in, the computer would recalculate all the formulae, and thus the contents of E4 would change to 4600.

The spreadsheet has developed with the addition of many functions.

7.2 The Instrument's Spreadsheet

The Instrument's spreadsheet was designed around the basic concept described above.

The spreadsheet was designed with 7 columns, A,B,C,D,E,F,G, and 30 rows, 1 to 30. Thus there are 210 (7*30) cells. Each cell can contain:

Text,	to label the spreadsheet.
A Value,	A constant value.
A Formula,	A formula consists of text, (the actual formula) and a value (the result of the formula).

Thus the spreadsheet is similar to standard PC spreadsheets.

However a few changes were required.

Firstly, the instrument would be working continuously, monitoring inputs, recalculating formulae, and setting the outputs to the required values. This meant that the spreadsheet would have to recalculate itself at regular intervals.

A timer function was added so that real time control could be implemented.

The standard financial functions were replaced by control functions, such as Proportional-Integral-Derivative (PID) control, Integration, Analogue and Digital input and output functions, etc.

And finally, since the unit didn't have a screen, another form of communication was required, and preferably with the ability to talk to an instrument which might be situated in a remote location with no easy access to a mains power supply. This problem has been solved by using a serial communications port, normally configured as an RS232 port, and by installing all the software onto the unit it was possible to make the instrument talk to standard dumb terminals, some of which can function from battery power (e.g. a Psion Organiser).

To make it easier to use, a PC hosted spreadsheet package has been developed to model the instrument and to allow quick and easy programming. This is described in the following chapter.

7.3 Spreadsheet Operation

7.3.1 Order of Recalculation

The instrument's spreadsheet starts at cell A1 and if this cell contains a formula, it will be recalculated, and the result of that operation stored in cell A1. Cell A2 is then examined, and again recalculated if it contains a formula. Then onto cell A3, A4 etc. to A30. After A30 is operated on, the instrument moves to column B and starts with B1,B2...B30 then C1,C2...C30 etc then finally to G30. At his point the entire spreadsheet has been recalculated, and the whole process is repeated from cell A1.

7.3.2 Speed of recalculation

The speed with which the instrument recalculates, depends entirely on the complexity of the spreadsheet. Typically a small spreadsheet of say 10 cells or so would recalculate in about 20mSec, and a large spreadsheet of about 50 cells would take about 100mSec to recalculate.

On average allow about 2mSec per formula cell.

7.3.3 Calculation Concepts

A standard formula will look at the value in certain cells and perform a calculation on them, with the result being stored in the formula cell.
For example:

	A	B
1	1.0	A1 + A2
2	2.0	B1 * 2

Cell B1 contains a formula 'A1 + A2'. This means it will look at the value in cell A1 and add it to the value in cell A2. The result of this calculation is placed in cell B1. So '1.0 + 2.0' equals 3.0 which is stored in cell B1.

Cell B2 contains the formula 'B1 * 2'. This takes the value of cell B1 and multiplies it by 2. So it takes the value 3.0 (placed there when cell B1 was calculated) and multiplies it by 2, giving 6.0 which is then stored in cell B2.

7.3.4 The '=' Equal function

The equal sign is used purely to test if 2 values are equal, and will not assign a value to a cell. Thus:

	A	B
1	ALG.IN1	IF(A1=A2,1,0)
2	ALG.IN2	

Here cell A1 contains a formula which gets the value of analogue input 1, and similarly, A2 gets the value of analogue input 2. Cell B1, takes the value stored in cell A1 and compares it to the value in cell A2. If they are equal to each other, then the value of cell B1 is made 1.0, otherwise it is set to 0.0.

The following example is a common mistake of the use of '='.

	A	B
1	ALG.IN1	IF(A1=A2, B2=1, B2=0)
2	ALG.IN2	

Here the user is trying to set the value of cell B2 to 1 if A1 is equal to cell A2. But as explained earlier, the equal sign **CANNOT** be used like this. The value in a cell can only be changed when the formula in that cell is calculated.

8.0 PC Software

8.1 Installing the PC Software

The software is supplied on a standard floppy disc. The first thing to do is to make a working copy. If your computer has a hard disc, you should copy all the files from the floppy disc into a directory on your hard disc.

The procedure for a **Hard Disc** is as follows:
Assumptions are: the floppy drive is "A:",
the hard drive is "C:",
all commands below are typed followed by the <Enter> key.

Insert the program floppy disc.
At c: prompt type:

```
md c-link
cd c-link
xcopy a: c: /s /e
```

The procedure for a **Floppy Disc** is as follows:
Assumptions are: the floppy drive is "A:",
there is only one floppy drive,
all commands below are typed followed by the <Enter> key.

Insert the Program floppy disc.
At c: prompt type:

```
diskcopy a: a: /s /e
```

This working copy should be the only one that is used, with the master copy stored safely, in case the software needs to be re-installed.

8.2 Running the Software

To run C-LINK type C-LINK at the DOS prompt. A title screen will be displayed and pressing any key will then display the main screen.

```
                        Industrial Interface Research   c:\...\?.chm
 File  Edit  fOrmat  Recalc  Chameleon  Monitor  Password
A1 Empty                                        Insert
                A               B               C               D               E           F        G
1
2
3
.
19
20
 Cells 210    Stack  100%   |                                      | Values  |  AutoCalc
```

The 2nd line is the **MENU** line.
The 3rd line is the **DATA** line.
The next section of the display shows the actual **SPREADSHEET**.
The bottom line is the **STATUS** line. This shows how many cells are still available and how much of the calculation stack is left. The calculation stack can go negative on the PC software. However, the Stack value must be greater than zero before the spreadsheet is programmed into a Chameleon. Also shown on the status line is which mode the display is in ; Values or Formula, and also whether the spreadsheet will be automatically re-calculated after every edit.

8.3 The Menu Line.

The menu line features the basic options available. Each option leads to a drop down menu with more detailed options available. To access any of the options the <ALT> key must be pressed together with the letter for the option. So, for example, to access the **File** menu press the **<ALT>** and **'F'** keys at the same time. The other options are:

<ALT> + F	->	File menu.
<ALT> + E	->	Edit menu.
<ALT> + O	->	fOrmat menu.
<ALT> + R	->	Recalc menu.
<ALT> + C	->	Chameleon menu.
<ALT> + M	->	Monitor menu.
<ALT> + P	->	Password menu.

Each of these actions leads to a menu box with a list of items. Each item has only 1 capital letter which will be in a different colour on a colour screen. This is the letter used to select that item.

Example:

To select File - Open.
Press <ALT> + F at the same time.
The 'File' Menu box is displayed.
Press 'O' by itself.
This selects the 'Open' item.

The full list of menu options is shown below:

File	**Edit**	**fOrmat**	**Recalc**	**Chameleon**	**Monitor**
Open	Edit <f2>	Range	Autocalc	Program	Continuous update
New	Delete <del>	Column width	Recalc <f1>	Read	Single update
Save	Goto	Auto col width <alt w>		Clear	Open rs485
Print		Formula		Version	cLose rs485
eXit <alt x>				Terminal	seT rs485
Directory				pOrt	
Tab save				Local echo	
				vErify	

Password
Enter password
New password
enAble password
Clear password

After some of the options a **Shortcut** key is shown. This allows quick access to the function. For example on the fOrmat menu, Auto col width has <alt w> after it. This means that by pressing the <ALT> key and the 'w' key at the same time the columns on the spreadsheet will be automatically resized to the maximum width of that column. Similarly, pressing the function key <f2> will allowing editing of the current cell.

8.4 The File Menu.

The File menu is used to open, close, save, and print spreadsheets and to exit the program.

Open. — This opens a dialog box allowing the user to enter the name of a file which he wishes to load into the spreadsheet. If the file does not exist or cannot be found a message will be displayed.

New. — This will clear the current spreadsheet from memory. If it has not been saved the user will be asked if it should be saved before being removed.

Save. — This opens a dialog box allowing the user to enter a filename so the current spreadsheet can be saved to disk. If the file already exists the user is asked if he wishes to replace the existing copy with the current spreadsheet.

Print. — This option leads the user through a series of dialog boxes to produce a printout of the current spreadsheet. It allows printing direct to a printer or to a file on disk, with options of printing in 132 columns and whether the border should be printed.

eXit. <Alt X> — This option exits from the program. However if the current spreadsheet has not been saved, the user will be given a chance to save it before leaving.

Directory — This opens a dialog box allowing the user to enter a directory name and file specification. The program will then show a directory listing of all the selected files. The user can enter the wildcard symbol *. So for example to find all the spreadsheet files beginning with ‘p’ the user would type \p*.chm

Tab save — This option allows the user to save the file in Tab delimited format. This is a useful way to export the spreadsheet so that other applications can read the data. For example if the spreadsheet had to be included in an operational manual, it could be saved and imported into a wordprocessor.

8.5 The Edit Menu.

Edit. <F2> — The edit option allows the user to edit the current highlighted spreadsheet cell. The current contents will be displayed on the data line, with the cursor at the end of the data. See Section 8.9 for more details.

Delete. <Del> — This will delete the contents of the current spreadsheet cell.

Goto. — This option prompts for a cell identifier, e.g. 'A2'. This cell will then become the current highlighted spreadsheet cell, and the screen will be re-drawn with this cell in the top left if possible.

8.6 The fOrmat Menu.

Range. — This leads the user through several options to allow the formatting of the data in every spreadsheet cell.

Column width. — This prompts the user for the required width of the current column. This will allow the user to select a width to allow the full contents of cell to be seen.

Auto col width. <Alt W> — This option automatically sets the column width for all columns so that the entire contents of each cell are clearly visible.

Formula. <Alt V> — This option toggles between a view of the spreadsheet where the current values of each cell are visible and a view where the actual formulae in each cell are visible. The word '**Formulae**' will appear towards the right of the status line if the screen is in formula-view mode, and '**Value**' when in value-view mode.

8.7 The Recalc Menu.

Autocalc. — This option switches the automatic recalculation mode on and off, displaying '**AutoCalc**' at the right of the status line if the option is on. Automatic recalculation will recalculate the entire spreadsheet if the contents of any cell are altered.

Recalc. <F1> — This will force a recalculation of the entire spreadsheet. This option allows the sheet to be recalculated even when the AutoCalc option is switched off.

8.8 The Chameleon Menu.

Program. This option will clear the contents of the connected instrument and program it with the current spreadsheet contents.

Read. This option reads the entire contents of a connected instrument and displays the cells on the screen. If there is a spreadsheet loaded already then it will be written over. If the instrument is older than Version 3.0, then the user will be prompted to enter the last cell used in the instrument. This cell reference is the largest column, and the largest row used. For example if cells A20 and C1 are used, then the last cell is C20.

Clear. This will clear the contents of the connected instrument.

Version. This shows what software version number the connected instrument has been programmed with.

Terminal. <Alt T> This option puts a small window on the screen. This window acts just like a standard 'dumb terminal', enabling the user to access the standard instrument menu structure directly. All the appropriate communication parameters are preset. Pressing <ESC> will allow the user to leave Terminal mode'.

pOrt. This option allows the user to set which COM serial port the instrument is connected to. A choice of COM1, COM2, COM3 and COM4 is available.

Local echo. This option is used to toggle local echo on and off. Local echo can be a problem if the RS485 communications lead is being used to talk to an instrument which does not have the RS485 option fitted.

vErify This option will verify that the spreadsheet shown on the screen is the same as that in the connected unit. This verify procedure is automatically performed as part of the programming process.

8.9 The Monitor Menu.

Continuous update.	This option allows the user to continuously monitor the values in a connected unit. A dialog box will appear asking the user to enter a delay period in seconds between each update. The minimum value is 1 second. Then the continuous monitoring will begin and continue until the user presses another key. It is sensible to change the screen display to 'Value' mode by pressing <alt v> before starting the continuous update, so that all the values are clearly shown on the display.
single Update <alt u>	This option will perform a single update of the values currently in a connected instrument.

The next 3 options are designed to work with units that have the RS485 communications option fitted.

Open rs485	This option asks the user to enter a channel number. The channel number must be from 1 to 32. If a connected unit has the selected address then the channel will be opened.
cLose rs485	This option asks the user to enter a channel number. The channel number must be from 1 to 32. If the currently selected unit has that address, then the channel will be closed.
seT rs485	This option asks the user to enter a channel number. The channel number must be from 1 to 32. The currently selected unit will have its address changed to the required value.

8.10 The Password Menu.

The password options are available on all units Version 4.0 and later. If an instrument is password protected it cannot be read, cleared or programmed.

Enter password.	This option allows the user to enter a password to unlock a connected unit which has been password protected. The user is told if the unit is not password protected. Otherwise a dialog box is displayed asking the user to enter the password. The password is sent to the instrument. The user is then informed whether the password was correct or not.
New password.	This option allows the user to enter a password to lock a connected unit which has not been password protected. If the connected unit is already password protected, then a dialog box appears prompting the user to enter the current password. If this password is correct or the unit was not locked then a dialog box is displayed asking the user to enter the new password. The password is entered twice and dialog appears to confirm that the unit should be password protected. The new password is then programmed into the unit
enAble password.	This option allows the user to enable the password protection of a connected unit. If the connected instrument does not have a current password or the password protection is currently enabled then a dialog box will appear to say that the password protection failed. Otherwise the protection will be activated.

Clear password. This option allows the user to clear the password protection of a connected unit. If the connected instrument does not have a current password a dialog box will inform the user that this is the case. If the unit is currently locked then a dialog box will appear asking the user to enter the current password. If the password is correct then the password protection will be cleared.

8.11 Using the PC Program.

Initially the spreadsheet is empty. At this point there are 3 options:

(i) Open a spreadsheet file which has already been saved to disc.
(ii) Read in the contents of a connected instrument.
(iii) Start entering a new spreadsheet.

The first 2 options are performed using the menu bar as detailed above. The final option, entering a new spreadsheet, is very straightforward.

Initially the current cell is A1. On the data line this is shown on the left, i.e. A1 Empty.
Pressing any letter or number on the keyboard immediately starts the editing process. Again on the data line you will see what has been typed. This line can be altered using the <delete>, <backspace>, <home>, <end> and the <left> and <right> cursor keys, and when the contents are correct simply press <Enter>. The data line contents will now appear in the current cell.

If the cell is correct you can use the cursor keys, <page up>, <page down>, <home> and <end> keys to move to other spreadsheet cells and enter data in them in the same fashion.

If the contents of a cell are incorrect, then there are a few options available.

(i) Simply type the correct version, this will automatically replace the incorrect information.

(ii) Use the Edit-Edit menu option, or press <f2>, to edit the contents of the cell.

(iii) Delete the cell by using either the Edit-Delete menu option or simply press the <delete> key. Either of these options will leave the cell empty, so that it can be edited again.

Once the entire spreadsheet is programmed it should be saved to disk using the File-Save option. An instrument can then be attached to the PC and programmed using the instrument menu.

8.12 Program Limitations

The PC program is limited in its imitation of the instrument spreadsheet.

Firstly, it does not have access to real input or output channels, and will normally read the inputs as 0.0. Likewise, it does not have access to timers, and thus the timer values will be returned as 0.0. Any functions which rely on time are also limited. To help in this matter the functions INTEG, PID, PULSE, and RATE will assume that 1 second has elapsed since the last time it was calculated.

The program will show the message ERROR in a cell which has a calculation error. On the instrument, this situation is handled so that the instrument continues to function until the error has corrected itself. For example:

	A
1	ALG.IN1
2	25
3	A2/A1

If the analogue input 1 is read as zero, there will be a divide by zero error. On the instrument cell A3 would be given a value of 1.0e30. However when the analogue input increases in value the calculation would become valid again.

On the PC, cell A3 would be shown as ERROR.

8.13 Hints

The PC spreadsheet can be used to prototype and test the operation of a instrument before actually connecting a instrument in a real situation.

The easiest way of doing this is to replace the analogue input statements (e.g. ALG.IN1) by actual values, and likewise digital input values. If the Autocalc option is switched off all the input values can be set to a test condition, and then the Options-Recalc, or <f1> option selected. The cells that have the output functions (i.e. ALG.OUT(...), DIG.OUT(...)), can then be looked at to see what the spreadsheet did.

There are several key shortcuts available. These are detailed on the menu options. For example, it is possible to quit the program by pressing the ALT and X keys simultaneously. This is marked on the File menu as:

"Exit <alt x>"

Other menus show similar shortcut options.

8.14 Program Conventions

All spreadsheets should be saved with an extension of .CHM to ensure compatibility with future versions.

For example a spreadsheet which programs the instrument to be a timing unit could be saved as TIMER.CHM.

9.0 The Direct Communications Link

9.1 System Overview

The instrument software includes a full function spreadsheet with all the standard mathematical functions and also includes a number of specialist functions. These include such features as P+I+D control algorithms, control timers, logic functions such as AND, OR, NOT etc., and range selection and scaling functions. A full list of the functions available is included in Section 10.0.

The instrument can be programmed using a Dumb Terminal by using the built-in menu. This is accessed through the RS232 port.

If any changes are made, the spreadsheet is automatically saved to non-volatile storage when editing is finished or the spreadsheet is cleared. It takes approximately 2 to 3 seconds to complete the store operation and **it is imperative that the power supply is maintained during this time otherwise the spreadsheet information will be lost.**

9.2 Programming The Spreadsheet

If the dumb terminal is connected, and power to the instrument is connected, the main menu should appear on the screen. If nothing happens, pressing <CR> (Carriage Return) should bring the menu up. If this does not happen refer to the communications section to ensure that the cable and all parameters have been set correctly.
The main menu on the instrument looks like this:

```
** CHAMELEON 4.0 **
  Edit          Display
  eRase        Clear
Select Option:_
```

The instrument spreadsheet is programmed from this main menu, using the commands Clear, eRase, and Edit. The current values in the spreadsheet can be examined with the Display command.

9.2.1 Clearing The Spreadsheet

When changing the functionality of the device or programming it for the first time, it is wise to clear the entire contents of the spreadsheet memory. To clear the spreadsheet select <u>Clear</u> (type "C<CR>") from the main menu. A prompt will appear as a double check of your intentions. Typing "Y<CR>" or "y<CR>" will continue with the operation and after a slight pause the message,

```
Saving Data...
**DO NOT SWITCH OFF **
```

will appear followed 2 seconds later by

```
Cleared
```

The Main Menu will then reappear. This sequence of display messages may not be appreciated when connected to a four line PSION organiser as the lines will scroll out of view.

9.2.2 Erasing A Single Spreadsheet Cell

If only one cell is to be cleared, selecting the "eRase" option (typing "R<CR>") will prompt the user for the cell location. Once entered the cell will be cleared. Note that there is no double check of your intentions.

9.2.3 Editing A Cell

When a spreadsheet is cleared, all the cells are set to the "Empty" attribute. Programming the spreadsheet effectively consists of entering formulae and values. A typical formula may be:

(A10+B10)/2

If this function were required the following procedure would be used:-

Select "Edit" (type "E<CR>") from the main menu
A prompt appears:

Edit which cell?
>A1
:

The value A1 is the current active cell. Typing <CR> will always select the current cell.

Alternatively typing any other cell reference, e.g. "D4<CR>" will make that cell the current cell and enter the edit mode. Assuming we decided to edit cell A1, on an empty spreadsheet we would see the following message:

A1 Empty
>
:

The first line shows the active cell, and what is in it (in this case it is empty), the second line is showing what the contents are (in this case no characters are present). The third line is the prompt for the user to type in the required value or formula.

So in our example, at the prompt we could type:-

(A10+B10)/2<CR>

This would then be examined by the spreadsheet program to assess whether it is text, a formula or value. The entire spreadsheet is then recalculated using the new information just entered, and the user is then prompted to enter another cell location. This procedure is followed until all the formulae and values have been entered.

If a long formula has been entered, it may be wise to check that the spreadsheet has interpreted it properly. To do this simply Edit that cell and ensure the attribute of the cell is correct, i.e. Text, Formula, or Value.

For our example above the display would show:-
A1 Formula
>(A10+B10)/2
:
If it is correct type <CR>, otherwise re-enter the formula correctly.

A point to note is that the spreadsheet will try to simplify all formulae to a value, if possible, in order to reduce the recalculation time, so the formula.

SQRT(9.00)
would become a value

3.00000
Similarly:
DIG.OUT1(1,0)
would reduce to
0.00000
although in this instance digital output 1 would be set to logical 0.

Thus, the entire spreadsheet can be programmed quickly and simply using just the three functions Clear, eRase and Edit as detailed above.

When the required cells have been edited it is necessary to return to the Main Menu. This is achieved by typing 'q<CR>' at the 'Which Cell?' prompt.

The display will then show

Saving Data...
** DO NOT SWITCH OFF **

Two seconds later

Data has been Saved

will appear followed by the Main menu and the Spreadsheet contents will be safely stored in the unit.

9.2.4 Displaying The Spreadsheet Contents

Once the spreadsheet has been programmed it is possible to examine the contents of every cell. In this way the input values and calculated values can be examined or data logged remotely down the communications channel.

In order to display the current value of a cell select Display from the main menu (type "D<CR>"). A prompt will appear to allow the selection of the required cell. Typing <CR> will automatically select the current active cell. The instrument will then output the cell name, attribute, formula if there is one, and either the calculated value, or the textual information stored in it.

A typical message could be:

B4 Formula	- cell and attribute
(A10+B10)/2	- formula
52.16597	- current value

Typing <CR> will then return to the prompt asking for the next cell to be displayed. In this way, any one cell, or any collection of cells, could be repeatedly interrogated by the user.

10.0 Function Reference

10.1 Using the Function Reference

The function reference describes each function in detail.
Firstly, the function name is given.

Next the list of arguments is presented showing which type of argument is valid.

The possible options for the arguments are as follows:

<Cell> means a cell reference by itself. i.e. the value of a cell is used as the value for the argument.
e.g.
PID(1,A1)
Uses the value in cell A1 to input to the PID channel 1.

<Number> means a pure number.
e.g.
PID(1,A1)
Uses the number 1 to select the PID channel.

<Result> means the result of a calculation.
e.g.
PID(1,A1*2.5)
Uses the result of the calculation A1*2.5
i.e. it takes the value from cell A1, multiplies it by 2.5 and uses the result to feed to the channel 1 PID loop.

The Function Description follows which gives a brief description, and if necessary describes the actual arguments of the function.

The typical function usage gives a few examples of the syntax of using the function.

Finally an example spreadsheet is shown which includes the usage of the function.

10.2 Precedence of Calculation

The precedence of the operators is:

1.	-	negate
2.	( )	parentheses
3.	* /	Multiply and Divide
4.	+ -	Add and Subtract
5.	Logic Functions:	AND NAND OR NOR XOR XNOR
6.	Comparison Functions:	< <= > >= = != <> ><
7.	All Other Functions	

So for example:

1+2*3	evaluates to 7. That is 2*3 equals 6, 1+6 equals 7.
(1+2)*3	evaluates to 9. That is (1+2) equals 3, 3*3 equals 9.
SQRT(a1+2*3)	evaluates to SQRT(a1+6).
5+-2	evaluates to 3. That is 5 + (-2) equals 3.

10.3 Function List

-	Negate
+	Addition
-	Subtraction
*	Multiply
/	Divide
>	Greater Than
<	Less Than
>=	Greater than or equal to
<=	Less than or equal to
=	Equals
<> >< !=	Not Equal to
:	Summation
ABS( )	Absolute Value
ALG.IN1, ALG.IN2, ALG.IN3	Value of Analogue Inputs (Channel 1, 2 or 3).
ALG.OUT( )	Output Analogue value
AND	Logical AND
CLIP( )	Clips the value
DIG.IN1, DIG.IN2	Value of Digital Inputs (Channel 1 or 2).
DIG.OUT( )	Output Digital Value
HALT.TIMER( )	Halts the timer
IF( )	If...then...else...
INTEG( )	Integrator
INTERP( )	Interpolate function
NAND	Logical NAND
NOR	Logical NOR
NOT( )	Logical NOT
OR	Logical OR
PID( )	Perform PID control loop
PULSE( )	Produce a pulse
RATE( )	Limit rate of change
ROUND( )	Rounds value to nearest integer
RST.TIMER( )	Reset timer value
RUN.TIMER( )	Start the timer
SAI( ), SET.ALG.IN( )	Set up analogue input mode
SAMPLE( )	Sample a value
SAO( ) SET.ALG.OUT( )	Set up analogue output mode
SCALE( )	Re-scale a value
SDI( ) SET.DIG.IN( )	Set up digital input mode
SDO( ) SET.DIG.OUT( )	Set up digital output mode
SET.PID( )	Set up PID loop
SQRT( )	Take square root
STEAM( )	Steam Tables
T1.STATUS, T2.STATUS	Timer status value (Channel 1 or 2).
TIMER1, TIMER2	Current timer value (Channel 1 or 2).
TRUNC( )	Round a value down
XNOR	Logical Exclusive NOR
XOR	Logical Exclusive OR

10.4 Variable List

FREQ	Frequency mode for digital inputs and outputs
LOGIC	Logic mode for digital inputs and outputs
MA0.20, MA4.20	0-20mA and 4-20mA scale values
VOLT0.10	0-10V scale values

Function Name: Algebraic Functions,

-, +, *, /

Arguments:

<Cell>
<Number>
<Result>

Function Description:

+ Addition,
- Subtraction,
* Multiplication,
/ Division,
- Negation.

Typical Function Usage:

A1 + B2
1.02 * B1 + C3
-SQRT(B2) / (A5 + B6)

Example Spreadsheet:

	A	B	C
1	ALG.IN1		
2	A1 * 10.0		
3			
4			
5			

Function Name: Comparison Functions,

>, <, >=, <=, =, <>, ><, !=

Arguments:

<Cell>
<Number>
<Result>

Function Description:

These are general comparison functions, which are normally used with the 'IF' function, although they can be used alone. When the comparison is true, the result of the operation is 1.0 otherwise the result is 0.0.

>	Greater Than,
<	Less Than,
>=	Greater Than or Equal,
<=	Less Than or Equal,
=	Equal,
<>, ><, !=	Not Equal.

Typical Function Usage:

A1 > B2
ALG.IN1 <= 12.0
IF(B2=100,1,0)
IF(ALG.IN2>12.0,B1*10,B1/12)

Example Spreadsheet:

High Signal Select, with Trip function.

	A	B	C
1	ALG.IN1		
2	ALG.IN2		
3	IF(A1>A2, A1, A2)		
4	IF(A2>=14.5,1,0)		
5	DIG.OUT(1,A4)		

Function Name: Summation, :

Arguments:

<Cell>: <Cell>

Function Description:

This function sums the values in all the cells between the first and last cells, either column-wise, or row-wise.

The start cell and end cell must be in the same column or in the same row.

Typical Function Usage:

A1:A10 Adds together all the values in cells A1,A2,A3,...A9,A10.
B2:F2 Adds together all the values in cells B2,C2,D2,E2,F2.

Example Spreadsheet:

Cell A5 would equal 1+2+3+4 = 10.0

	A	B	C
1	1.0		
2	2.0		
3	3.0		
4	4.0		
5	A1:A4		

Function Name: Absolute Value ABS( )

Arguments:

<Cell>
<Number>
<Result>

Function Description:

This function finds the absolute value of an expression.
i.e. it makes a number positive, if it was negative.

e.g. ABS(-1.0) = 1.0
ABS(2.0) = 2.0

Typical Function Usage:

ABS(A1)
ABS(1.02 * B1 - C3)
ABS(-SQRT(B2))

Example Spreadsheet:

	A	B	C
1	ALG.IN1		
2	ABS(A1)		
3			
4			
5			

Function Name: Analogue Input Variable

ALG.IN1, ALG.IN2, ALG.IN3.

Arguments:

No Arguments.

Function Description:

These return the current value of the analogue inputs.

All scaling is taken into account.

Typical Function Usage:

ALG.IN1
ALG.IN2
ALG.IN3

Example Spreadsheet:

	A	B	C
1	ALG.IN1		
2	IF(A1>10,100,0)		
3			
4			
5			

Function Name: Analogue Output Function ALG.OUT( ).

Arguments:

ALG.OUT(*<Number>, <Expr>*)

where *<Expr> can be <Cell>, <Result>*

Function Description:

ALG.OUT(Channel, Value)

Sets the analogue output to the given value. The scaling of the output is taken into account. The value will be clipped between the maximum and minimum values of the analogue output scales.

Typical Function Usage:

ALG.OUT(1, A1)
AlG.OUT(2, A1*A2)

Example Spreadsheet:

Re-transmits analogue input 1 if digital input 1 is logic 1, otherwise re-transmits analogue input 2. Digital input 1 is inverted and put out on digital output 1.

	A	B	C
1	ALG.IN1	ALG.IN2	
2	DIG.IN1		
3	IF(A2=1,A1,B1)		
4	ALG.OUT(1,A3)		
5			

Function Name: Clip Function CLIP( )

Arguments:

CLIP(*<Cell>, <Expr>, <Expr>*)

where *<Expr> can be <Cell>, <Number>, <Result>.*

Function Description:

This clips a value between a minimum and a maximum value.

CLIP(Cell, Minimum Value, Maximum Value)

Typical Function Usage:

CLIP(A1,A2,A3)
CLIP(A1,0.0,100.0)

Example Spreadsheet:

Re-transmits analogue input 1 on analogue output 1 but clipped between the values of 4.0 and 16.0.

	A	B	C
1	ALG.IN1		
2	CLIP(A1,4,16)		
3	ALG.OUT(1,A2)		
4			
5			

Function Name: Digital Input Variable

DIG.IN1, DIG.IN2.

Arguments:

No Arguments.

Function Description:

Returns the current value of the digital inputs.

If the input is configured as a logic input, the returned value will be either logic 1.0, or logic 0.0.

If the input is configured as a frequency input, the returned value will be scaled accordingly.

Typical Function Usage:

DIG.IN1
DIG.IN2

Example Spreadsheet:

Re-transmits analogue input 1 if digital input 1 is logic 1, otherwise re transmits analogue input 2.

	A	B	C
1	ALG.IN1	ALG.IN2	
2	DIG.IN1		
3	IF(A2=1,A1,B1)		
4	ALG.OUT(1,A3)		
5			

Function Name: Digital Output Function DIG.OUT( ).

Arguments:

DIG.OUT(*<Number>, <Expr>*)

where *<Expr> can be <Cell>, <Result>*

Function Description:

DIG.OUT(Channel, Value)

Sets the digital output to the given value.

If the output is configured as a logic output, the output will be energised if the given value is non-zero, otherwise the output will be switched off.

If the output is configured as a frequency output, the output will be set to the required frequency.

Typical Function Usage:

DIG.OUT(1, A1)
DIG.OUT(2, A1*A2)

Example Spreadsheet:

Re-transmits analogue input 1 if digital input 1 is logic 1, otherwise re transmits analogue input 2. Digital input 1 is inverted and put out on digital output 1.

	A	B	C
1	ALG.IN1	ALG.IN2	
2	DIG.IN1		
3	IF(A2=1,A1,B1)		
4	ALG.OUT(1,A3)		
5	DIG.OUT(1,NOT(A2)		

Function Name: Halt Timer HALT.TIMER( ).

Arguments:

HALT.TIMER(*<Number>, <Expr>*)

where *<Expr> can be <Cell>, <Result>*

Function Description:

HALT.TIMER(Channel, Halt Condition)

If the Halt Condition is logic 1, the timer will be stopped at the current value. It will resume as soon as the halt condition is removed.

There are 2 timers available.

Typical Function Usage:

HALT.TIMER(2, A10) Halts timer 2 when cell A10 is logic 1.

Example Spreadsheet:

This sets a continuously running 20 second timer, which is halted when digital input 1 is logic 1.

	A	B	C
1	20.0	TIMER1	
2	DIG.IN1	T1.STATUS	
3		RST.TIMER(1,B2=3)	
4		RUN.TIMER(1,A1)	
5		HALT.TIMER(1,A2)	

Function Name: IF...THEN...ELSE

Arguments:

<Cell>
<Number>
<Result>

Function Description:

This is a standard if then else function. If an expression is TRUE i.e. logic '1', then the second expression is the result otherwise the third expression is the result.

Again the function takes any non-zero value to be TRUE.

This function is typically used with comparison and logic functions.

The function is constructed as follows:
IF(expr1, expr2, expr3)
If (expr1 is true) (then the result is expr2) (else the result is expr3).

NOTE: Both expr2 and expr3 are calculated before the final value is returned. This means that functions like analogue output functions should not form the basis of expr2 and expr3.

Typical Function Usage:

IF(A1, 1.0, 0.0)
IF(ALG.IN1>ALG.IN2, 1, 0)
IF(B1<12, 12, 20)

Example Spreadsheet:

High Signal Selector.

	A	B	C
1	ALG.IN1		
2	ALG.IN2		
3	IF(A1 > A2, A1, A2)		
4	ALG.OUT(1,A3)		
5			

Function Name: Integration INTEG( )

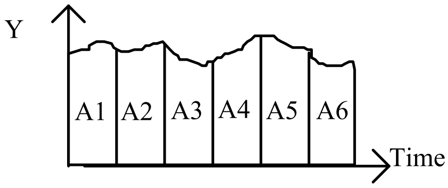
Arguments:

INTEG(*<Number>, <Cell>, <Cell>*)

Function Description:

This function performs integration of a value in real time.
INTEG(Channel, Value, Wrap Value)

In effect, INTEG( ) calculates the area under a graph:



i.e. INTEG( ) returns the sum: A1+A2+A3+A4+A5+A6

For example, if the flow rate of a substance was being measured, the function would integrate this flow rate to produce the total amount of the substance to have passed the flow sensor. Typically a flow sensor could be calibrated in units of Kg/hour. This value should then be divided by 3600 to give a value in Kg/sec. The INTEG( ) function would then produce a total in units of Kg.

The integral function has a maximum limit of 9,999,999, at which point it wraps back to 0, and the integration begins again. However, the user can set the point at which the integral sum is wrapped, by setting a 'Wrap Value'.

There are 5 integrators available, selected using the Channel number, which must be a 1, 2, 3,4 or 5.

Typical Function Usage:

INTEG(1, A1, A3)

Example Spreadsheet:

Analogue input 1 is integrated using integrator channel 1. The Integral total is in cell A2. When this value reaches 1000 cell A3 sets the Wrap Value to 1000, thus on the next integral calculation the integral sum will subtract 1000 from itself, effectively resetting the sum to zero plus the carry.

	A	B	C
1	ALG.IN1		
2	INTEG(1, A1, A3)		
3	IF(A2>=1000,1000,0)		
4			

Function Name: Interpolation Function INTERP( )

Arguments:

INTERP(*<Cell>, <Cell>, <Cell>*)

Function Description:

This functions examines a range of cells and interpolates a value from the table.

INTERP(Value, Start Cell, End Cell)

The table of values must be arranged in two columns. The first column contains the 'x' values and the second column contains the 'y' values. The function finds where Value fits into the 'x' range, clipping where necessary, and then interpolates the result from the 'y' values.

Typical Function Usage:

INTERP(A1, A2, B5)

Example Spreadsheet:

This function takes the analogue input value and interpolates a new value from the look up table in cells A1 to B5. The result is in cell C2.

	A	B	C
1	0.0	0.0	ALG.IN1
2	1.0	10.0	INTERP(C1,A1,B5)
3	2.0	40.0	
4	3.0	90.0	
5	4.0	160.0	

Function Name: Logic Functions,

AND, OR, NAND, NOR, XOR, XNOR

Arguments:

<Cell>
<Number>
<Result>

Function Description:

These are general logic functions, which are normally used with the 'IF' function, although they can be used alone. When the logic function is true, the result of the operation is 1.0 otherwise the result is 0.0.

All the logic functions take a non-zero value as being logic '1' or TRUE and a value of 0.0 to be logic '0' or FALSE.

AND	Logical AND function,
OR	Logical OR function,
NAND	Logical NAND function,
NOR	Logical NOR function,
XOR	Logical XOR function,
XNOR	Logical XNOR function.

Typical Function Usage:

A1 AND B2
A1 OR B1 OR C3
(A1 NAND B1) OR (B3 AND C2)
IF(A1ANDB2,1,0)

Note: Spaces are optional in the above examples.

Example Spreadsheet:

	A	B	C
1	DIG.IN1		
2	DIG.IN2		
3	IF(A1 OR A2, 1, 0)		
4	DIG.OUT(1,A3)		
5			

Function Name: Logic NOT( )

Arguments:

<Cell>
<Number>
<Result>

Function Description:

This complements the value of the expression.
A logic '1' becomes logic '0' and vice versa.

Typical Function Usage:

NOT(A1)
A1 OR NOT(B2)
IF(NOT(A1)ORB1, 1, 0)

Example Spreadsheet:

	A	B	C
1	DIG.IN1		
2	DIG.IN2		
3	IF(A1 OR NOT(A2), 1, 0)		
4	DIG.OUT(1,A3)		
5			

Function Name: PID Control Loop PID( )

Arguments:

PID(*<Number>, <Expr>*)

where *<Expr> can be <Cell>, <Result>*

Function Description:

The PID function implements a standard Proportional-Integral-Derivative control loop.

There are two channels of PID control available.

The parameters to set up the PID loop are configured using the SET.PID( ) function.

The PID control system expects the process variable to be scaled 0 to 100. The output from the PID control loop is likewise scaled 0 to 100. If an input is being used for the PID input, then it can be scaled to 0 to 100 using the SAI( ) function. Alternatively, the SCALE( ) function can be used.

Typical Function Usage:

PID(1,B1)

Example Spreadsheet:

	A	B	C
1	Setpoint:	ALG.IN2	SAI(1,MA4.20,0,100)
2	Proportional Band:	20.0	SAI(2,MA4.20,0,100)
3	Integral:	60.0	SAO(1,MA4.20,0,100)
4	Derivative:	0.0	
5	Set Up PID:	SET.PID(1,B1,B2,B3,B4)	
6	Input:	ALG.IN1	
7	PID Control:	PID(1,B6)	
8	Output:	ALG.OUT(1,B7)	

Function Name: Pulse Function PULSE( )

Arguments:

PULSE(*<Number>, <Expr>, <Expr>, <Expr>, <Expr>*)

where *<Expr> can be <Cell>, <Number>, <Result>*

Function Description:

PULSE(Channel, Value, Wrap Value, Pulse Step, Pulse Length)

The pulse function is designed to monitor an increasing value and produce a pulse when the value reaches pre-defined targets: the Pulse Step. It is primarily used in conjunction with the INTEG( ) function described earlier.

Typically the INTEG( ) function could be monitoring flow rate and calculating the total mass flow. The pulse function could then be used to produce for every 10Kg. When the PULSE( ) function is used with the INTEG( ) function, the Wrap Value should ideally be set to the same amount.

The length of the pulse produced can also be set using Pulse Length.

The PULSE( ) function will ensure that no pulses are lost. For example, if the pulse length is set to 1 sec, but the input value has increased quickly, requiring 2 pulses per second, then PULSE( ) keeps a record of the number of pulses required and will catch up when the rate of change of the input value has slowed down sufficiently.

The pulse function can be fed to a digital output function to create a pulse stream on one of the digital outputs.

There are 3 channels of the pulse function.

Typical Function Usage:

PULSE(1, A1, 1000, 50, 0.5)
PULSE(2, A1, A2, A3, A4, A5)

Example Spreadsheet:

This spreadsheet integrates the analogue input. The integral sum wraps around at 1000. The pulse function also wraps at 1000 to keep in synch with the integral total, and it produces a 0.5 second pulse every time the integral sum increases by 50.

	A	B	C
1	ALG.IN1		
2	INTEG(1,A1,A3)		
3	IF(A2>=1000,1000,0)		
4	PULSE(1,A2,1000,50,0.5)		
5			

Function Name: Rate Limiting Function RATE( )

Arguments:

RATE(*<Number>, <Expr>, <Expr>*)

where *<Expr> can be <Cell>, <Result>*

Function Description:

This function limits the rate of change of value to a predetermined rate of change, expressed as units per second.

RATE(Channel, Value, Maximum Rate of Change)

There are 3 channels of rate of change available.

If the Value changes suddenly, this function will only change its value to the new value at the pre-defined rate.

E.g. Assume the maximum allowed rate of change is 5 units per second.
If the input Value changed from 0 to 100 instantaneously, the output of the Rate function could only change at the maximum rate of 5 units per second. Thus it would take 100/5 = 20 seconds for the rate function to finally reach the value of 100.

Typical Function Usage:

RATE(1, A1, B2)
RATE(2, A1, 5.0)

Example Spreadsheet:

This example sets a value to 100 if the analogue input increases above a value of 10. This is fed to the analogue output after being Rate Limited to ensure a 'bumpless' output.

	A	B	C
1	ALG.IN1		
2	IF(A1>10,100,0)		
3	RATE(1,A2,5.0)		
4	ALG.OUT(1,A3)		
5			

Function Name: Rounding ROUND( )

Arguments:

<Cell>
<Number>
<Result>

Function Description:

This function rounds the result to the nearest integer value.
e.g. ROUND(1.4) = 1.0
ROUND(1.7) = 2.0

Typical Function Usage:

ROUND(A1)
ROUND(1.02 * B1) + C3

Example Spreadsheet:

	A	B	C
1	ALG.IN1		
2	ROUND(A1)		
3			
4			
5			

Function Name: Reset Timer RST.TIMER( ).

Arguments:

RST.TIMER(*<Number>, <Expr>*)

where *<Expr> can be <Cell>, <Result>*

Function Description:

RST.TIMER(Channel, Reset Condition)

If the Reset Condition is logic 1, the timer will be reset. This will then allow the timer to be started.

There are 2 timers available.

Typical Function Usage:

RST.TIMER(2, A10) Resets timer 2 when cell A10 is logic 1.

Example Spreadsheet:

This sets a continuously running 20 second timer.

	A	B	C
1	20.0	TIMER1	
2		T1.STATUS	
3		RST.TIMER(1,B2=3)	
4		RUN.TIMER(1,A1)	
5			

Function Name: Start Timer RUN.TIMER( ).

Arguments:

RUN.TIMER(*<Number>, <Expr>*)

where *<Expr> can be <Cell>*

Function Description:

RUN.TIMER(Channel, End Value)

If the timer has been reset, this function will start the timer with the finish value (in seconds) given.

There are 2 timers available.

Typical Function Usage:

RUN.TIMER(2, A10) Starts a timer with the number of seconds in cell A10.

Example Spreadsheet:

This sets a continuously running 20 second timer.

	A	B	C
1	20.0	TIMER1	
2		T1.STATUS	
3		RST.TIMER(1,B2=3)	
4		RUN.TIMER(1,A1)	
5			

Function Name: Set Analogue Input SAI( ) & SET.ALG.IN( ).

Arguments:

SAI(*<Number>, <Mode>, <Number>, <Number>*)

where *<Mode>* can be MA0.20, MA4.20, VOLT0.10.

Function Description:

SAI(Channel, Mode, Low Scale, High Scale)

These are used to set the analogue output range.

VOLT0.10 Variable used for 0-10V operation.
MA0.20 Variable used for 0-20mA operation.
MA4.20 Variable used for 4-20mA operation.

The low scale must be a lower value than the high scale value. If the low scale must be higher than the high scale, the output value can be re-scaled using the SCALE( ) function described earlier.

Both SAI( ) and SET.ALG.IN( ) have the same effect, but it is recommended that SAI( ) is used to maintain compatibility with future releases.

Typical Function Usage:

SAI(1,MA0.20,0,20)

Example Spreadsheet:

Re-transmits analogue input 1 (4-20mA) onto analogue output 1 (0-10V).

	A	B	C
1	SAI(1,MA4.20,4,20)	ALG.IN1	
2	SAO(1,VOLT0.10,4,20)	ALG.OUT(1,B1)	
3			
4			
5			

Function Name: Sample Function SAMPLE( )

Arguments:

SAMPLE(*<Cell>, <Expr>*)

where *<Expr> can be <Cell>, <Result>*

Function Description:

This function samples and stores the value of cell when the sample condition is TRUE.

SAMPLE(Cell, Sample Condition)

If the Sample Condition is a logic '1', the sample function reads the value of the cell. However, when the Sample Condition goes to logic '0', the last value sampled is stored and held constant.

Typical Function Usage:

SAMPLE(A1, B2)
SAMPLE(A1, IF(B2>10,1,0))

Example Spreadsheet:

This example samples the analogue input value when the digital input is TRUE, and stores the value when the digital input goes low.

	A	B	C
1	ALG.IN1		
2	DIG.IN1		
3	SAMPLE(A1,A2)		
4			
5			

Function Name: Set Analogue Output

SAO( ) & SET.ALG.OUT( ).

Arguments:

SAO(*<Number>, <Mode>, <Number>, <Number>*)

where *<Mode>* can be MA0.20, MA4.20, VOLT0.10.

Function Description:

SAO(Channel, Mode, Low Scale, High Scale)

These are used to set the analogue output range.

VOLT0.10 Variable used for 0-10V operation.
MA0.20 Variable used for 0-20mA operation.
MA4.20 Variable used for 4-20mA operation.

The low scale must be a lower value than the high scale value. If the low scale must be higher than the high scale, the output value can be re-scaled using the SCALE( ) function described earlier.

Both SAO( ) and SET.ALG.OUT( ) have the same effect, but it is recommended that SAO( ) is used to maintain compatibility with future releases.

Typical Function Usage:

SAO(1,MA0.20,0,20)

Example Spreadsheet:

Re-transmits analogue input 1 (4-20mA) onto analogue output 1 (0-10V).

	A	B	C
1	SAI(1,MA4.20,4,20)	ALG.IN1	
2	SAO(1,VOLT0.10,4,20)	ALG.OUT(1,B1)	
3			
4			
5			

Function Name: Scaling Function SCALE( )

Arguments:

SCALE(*<Cell>, <Expr>, <Expr>, <Expr>, <Expr>*)

where *<Expr> can be <Cell>, <Number>, <Result>*

Function Description:

This function re-scales a value to fit a straight line.

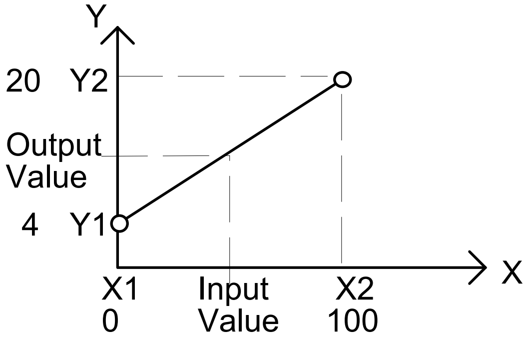
SCALE(Cell, X1, Y1, X2, Y2)

The Value of Cell is clipped to the range X1 to X2, and the corresponding value from the straight line, (X1,Y1) to (X2,Y2) is calculated..

There are no restrictions to the use of the X,Y pairs, thus an analogue input could simply be re-scaled before re-transmission, or it could be inverted.

Typical Function Usage:

SCALE(A1,A2,A3,A4,A5)
SCALE(A1, 0, 100, 100, 0)
E.g. SCALE(A1, 0, 4, 100, 20)



Example Spreadsheet:

This takes a 4 to 20 mA signal in scaled 4 to 20, and re-scales this to 0 to 100 before re-transmitting on analogue output 1, and also inverts the signal before re-transmitting on analogue output 2.

	A	B	C
1	SET.ALG.IN(1,MA4.20,4,20)	ALG.OUT(1,A3)	
2	ALG.IN1	ALG.OUT(2,A4)	
3	SCALE(A2,4,0,20,100)		
4	SCALE(A2,4,20,20,4)		
5			

Function Name: Set Digital Input SDI( ) & SET.DIG.IN( ).

Arguments:

SDI(*<Number>, <Mode>, <Number>, <Number>*)

where *<Mode>* can be LOGIC, FREQ.

Function Description:

SDI(Channel, LOGIC)
SDI(Channel, FREQ, Low Freq, High Freq, Low Scale, High Scale)

These are used to set the digital inputs range.

LOGIC Variable used for on / off operation.
FREQ Variable used for frequency operation.

In frequency mode, the low scale must be a lower value than the high scale value. If the low scale must be higher than the high scale, the input value can be re-scaled using the SCALE( ) function described earlier.

Both SDI( ) and SET.DIG.IN( ) have the same effect, but it is recommended that SDI( ) is used to maintain compatibility with future releases.

Typical Function Usage:

SDI(1,LOGIC)
SDI(2,FREQ,0.01,5,0.01,5)

Example Spreadsheet:

Sets Digital Input 1 to be a frequency input with a range of 0.5Hz to 10Hz, scaled to give a value of 0 to 100.

	A	B	C
1	SDI(1,FREQ,0.5,10,0,100)		
2	DIG.IN1		
3			
4			
5			

Function Name: Set Digital Output SDO( ) & SET.DIG.OUT( ).

Arguments:

SDO(*<Number>, <Mode>, <Number>, <Number>*)

where *<Mode>* can be LOGIC, FREQ.

Function Description:

SDO(Channel, LOGIC)
SDO(Channel, FREQ, Low Freq, High Freq, Low Scale, High Scale)

These are used to set the digital outputs range.

LOGIC Variable used for on / off operation.
FREQ Variable used for frequency operation.

In frequency mode, the low scale must be a lower value than the high scale value. If the low scale must be higher than the high scale, the output value can be re-scaled using the SCALE( ) function described earlier.

The Low frequency cannot be less than 1.75e-4Hz.
The High frequency cannot be more than 50Hz for relay outputs.

Both SDO( ) and SET.DIG.OUT( ) have the same effect, but it is recommended that SDO( ) is used to maintain compatibility with future releases.

Typical Function Usage:

SDO(1,LOGIC)
SDO(2,FREQ,0.01,5,0.01,5)

Example Spreadsheet:

Sets Digital Output 1 to be a frequency input with a range of 0.5Hz to 10Hz, scaled to give a value of 0 to 100.

	A	B	C
1	SDO(1,FREQ,0.5,10,0,100)	SAI(1,MA4.20,0,100)	
2	DIG.OUT(1,A3)		
3	ALG.IN1		
4			
5			

Function Name: PID Loop Parameter Set Up SET.PID( )

Arguments:

SET.PID(*<Number>, <Expr>, <Expr>, <Expr>, <Expr>*)

where *<Expr> can be <Cell>, <Result>, <Number>*

Function Description:

This function sets the parameters of a PID control loop.

SET.PID(Channel, Setpoint, Proportional Band, Integral Time, Derivative Time)

There are two PID channels available.

The range for the Proportional Band is 0.5% to 1000%.

The Integral Time can be set between 6.0 secs and 6000 secs.
It can also be set to OFF using a value of 0.0.
The Integrator has a Wind Up Inhibit feature, i.e. the Integral term is set to zero when the signal is outside of the proportional band.

The Derivative Time can be set between 0.6 secs and 600 secs.
It can also be set to OFF with a value of 0.0.

Typical Function Usage:

SET.PID(1, A1, A2, 0.0, 0.0)

Example Spreadsheet:

	A	B	C
1	Setpoint:	ALG.IN2	SAI(1,MA4.20,0,100)
2	Proportional Band:	20.0	SAI(2,MA4.20,0,100)
3	Integral:	60.0	SAO(1,MA4.20,0,100)
4	Derivative:	0.0	
5	Set Up PID:	SET.PID(1,B1,B2,B3,B4)	
6	Input:	ALG.IN1	
7	PID Control:	PID(1,B6)	
8	Output:	ALG.OUT(1,B7)	

Function Name: Square Root Function SQRT( )

Arguments:

<Cell>
<Number>
<Result>

Function Description:

This function will find the square root of any positive number.
If the number is negative the result is 0.0.

Typical Function Usage:

SQRT(A1)
1.02 * SQRT(B1 + C3)
-SQRT(B2) / (A5 + B6)

Example Spreadsheet:

	A	B	C
1	ALG.IN1		
2	10.0 * SQRT(A1)		
3			
4			
5			

Function Name: STEAM Function STEAM( )

Arguments:

STEAM(*<Number>, <Cell>, <Cell>)*

Function Description:

STEAM(Output Required, Pressure, Temperature)

The STEAM function performs the task of calculating from standard steam tables various properties of steam at different conditions of pressure and temperature. Typically the function is used to compensate flow meters for pressure and temperature variations, but the function can be used to convert any standard volumetric or mass flow meter into a fully compensated mass and energy flow meter.

The output values are as follows:

STEAM(0, P, T)	gives the specific volume of the steam
STEAM(1, P, T)	gives the enthalpy of the steam
STEAM(2, P, T)	gives the saturated temperature of the steam

all at the conditions of P, pressure in bar absolute, and T, the temperature of the steam. If the saturated temperature of the steam is not known, it may be determined from the pressure by using a dummy temperature as below:

STEAM(2, P, T_D) returns the saturated temperature of the steam at pressure P

NOTE: P, T and T_D all refer to cell references which contain the required values.

Example Spreadsheet:

This example calculates the various properties of the steam being measured using a pressure transmitter on analog input 1 and a temperature transmitter on analog input 2. Cell C1 is a superheat alarm which operates if the actual temperature of the steam is 5°C above the calculated saturated temperature at the pressure input.

	A	B	C
1	Pressure:	ALG.IN1	(B2 - B5) > 5
2	Temp:	ALG.IN2	DIG.OUT(1,C1)
3	Specific Volume:	STEAM(0,B1,B2)	
4	Enthalpy:	STEAM(1,B1,B2)	
5	Saturated Temp:	STEAM(2,B1,B2)	

Function Name: Timer Status Values

T1.STATUS & T2.STATUS.

Arguments:

No Arguments.

Function Description:

These variables return the current status of the timers.
The value is 0, 1, 2 or 3.

0 means the timer is reset and ready to run.
1 means that the timer is running.
2 means that the timer is currently halted.
3 means that the timer has reached its end value.

The timer will not respond to the run command unless it is in state 0.

When the timer is run it will enter state 1.

If he timer is halted, it will enter state 2, and will return to state 1 when the halt condition is removed.

Typical Function Usage:

T1.STATUS Returns the status of timer 1 in seconds.
T2.STATUS Returns the status of timer 2 in seconds.

Example Spreadsheet:

This sets a continuously running 20 second timer, which is halted when digital input 1 is logic 1.

	A	B	C
1	20.0	TIMER1	
2		T1.STATUS	
3		RST.TIMER(1,B2=3)	
4		RUN.TIMER(1,A1)	
5			

Function Name: Timer Values TIMER1 & TIMER2.

Arguments:

No Arguments.

Function Description:

These variables return the current values of the timers in seconds.

Typical Function Usage:

TIMER1	Returns the value of timer 1 in seconds.
TIMER2	Returns the value of timer 2 in seconds.

Example Spreadsheet:

This sets a continuously running 20 second timer, which is halted when digital input 1 is logic 1.

	A	B	C
1	20.0	TIMER1	
2		T1.STATUS	
3		RST.TIMER(1,B2=3)	
4		RUN.TIMER(1,A1)	
5			

Function Name: Truncation TRUNC( )

Arguments:

<Cell>
<Number>
<Result>

Function Description:

This function truncates the result of an expression to the lower integer value.
i.e. It rounds a number down.
e.g. TRUNC(1.65) = 1.00

Typical Function Usage:

TRUNC(A1)
TRUNC(1.02 * B1) + C3

Example Spreadsheet:

	A	B	C
1	ALG.IN1		
2	TRUNC(A1)		
3			
4			
5			

Variable Name: Input and Output range variables.

VOLT0.10, MA0.20, MA4.20.

Arguments:

No Arguments.

Function Description:

These are used to set the analogue inputs and outputs to the required range.

VOLT0.10 Variable used for 0-10V operation.
MA0.20 Variable used for 0-20mA operation.
MA4.20 Variable used for 4-20mA operation.

Typical Function Usage:

SAI(1,MA0.20,0,20)

Example Spreadsheet:

Re-transmits analogue input 1 (0-20mA) onto analogue output 1 (0-10V).

	A	B	C
1	SAI(1,MA0.20,0,20)	ALG.IN1	
2	SAO(1,VOLT0.10,0,20)	ALG.OUT(1,B1)	
3			
4			
5			

Variable Name: Digital Input and Output range variables.

LOGIC, FREQ

Arguments:

No Arguments.

Function Description:

These are used to set the digital inputs and outputs to the required range.

LOGIC	Variable used for on / off operation.
FREQ	Variable used for frequency operation.

Typical Function Usage:

SDI(1,LOGIC)
SDO(2,FREQ,0.01,5,0.01,5)

Example Spreadsheet:

Re-transmits analogue input 1 (0-20mA) onto digital output 1 (frequency 0.1-10Hz).

	A	B	C
1	SAI(1,MA0.20,0,20)	ALG.IN1	
2	SDO(1,FREQ,0.1,10,0.1,10)	DIG.OUT(1,B1)	
3			
4			
5			

11.0 Example Spreadsheets

11.1 Application Examples

Application examples listed below can be found on the C-LINK software disk.

AIRFLOW.CHM
Compressed air mass flow compensation

ALGFREQ.CHM
Analogue to Frequency conversion

ANATROL1.CHM
Automatic Calibration Routine

COOLTOWR.CHM
Fan and Pump Control for cooling towers

DIFF.CHM
Difference of two analogue inputs + Trip

DUALBURN.CHM
High/Low fire burner application

FLOWLIN.CHM
Steam Flow P & T Compensation for Mass & Energy

FLOWSUM.CHM
Summation of inlet & outlet flows for pumping station

GASMETER.CHM
AGA 3 gas calculations for temp/pressure compensation

GLYCOL.CHM
Glycol detection and control algorithm
HILOWSEL.CHM
Selects high or low signal from two analogue inputs

INTEGRAT.CHM
Integrates an analogue input to provide a pulse output

LINEAR1.CHM
Linearises (or non-linearises) one or two analogue inputs

MATHS1.CHM
Example of Mathematics available

MAXNOTE.CHM
Remembers Maximum input value

NETENERG.CHM
Net energy flow computer for steam/condensate.

PAPERBK.CHM
Paperbreak system for continuos paper mills

PENSTOCK.CHM
Penstock valve control for sewage inlet valve control

PIDCON.CHM
PID control algorithm for control applications

PULSDIV.CHM
Pulse Divider or Frequency divider

RATETRIP.CHM
Rate of change Trip Amplifier

RATIOCON.CHM
Ratio Control for two gases

ROLLAVE.CHM
Calculates a rolling average over a variable period

SILOFILL.CHM
Automatic Silo filling Program

SQROOT.CHM
Square Root extraction application

SUMAVE.CHM
Calculates the sum and average of three analogue inputs

SUMTRIP.CHM
Trip Amplifier operating on summated inputs

TRIPLE1.CHM
Triple Validation of sensor inputs

VALVCON1.CHM
Controls analogue valve using raise/lower inputs + trip

11.2 Temperature Measurement System

The first example is a peak measurement system.

	A	B	C	D
1	SAI(1,VOLT0.10,0,100)	Temp Input:	ALG.IN1	C1>D2
2		Start Cycle:	DIG.IN1	IF(C2,C1,IF(D1ANDNOT(D5),C1,D2))
3	SAO(1,VOLT0.10,0,100)	Sensitivity:	10	IF(C2 OR D1, 10, 0)
4	SAO(2,VOLT0.10,0,100)			IF(C1<(D2-C3),1,0)
5		Temp Output:	ALG.OUT(1,C1)	IF(C2,0,IF(D6,1,D4))
6	SDO(1,FREQ,1.8E-4,1,0,10)	Peak Temp:	ALG.OUT(2,D2)	D5
7	SDO(2,LOGIC)	Freq Output:	DIG.OUT(1,D7)	NOT(D5)*10
8		End of Cycle:	DIG.OUT(2,D5)	
9	SDI(1,LOGIC)			

Column A has all the range setting functions included.

Analogue input 1 is set to the 0 to 10 volt range with a low scale of 0 and a high scale of 100.
Both analogue outputs are 0 to 10 volts with scales of 0 to 100.
The first digital input is set to frequency mode with a low frequency of 1.8e-4 Hz and a high frequency of 1Hz, relating to a low scale of 0 and a high scale of 10 respectively.
The second digital output is in logic mode.
The first digital input is set to logic mode.

Column B contains labels to help explain the functionality of the next column. Comments are to be recommended in all spreadsheets to assist in debugging and analysis of the functionality of the instrument.

Column C is used to control the input and output operations.

Cell C1 samples analogue input 1 and stores the result.
Cell C2 gets the status of digital input 1.
Cell C3 contains a value used to set the sensitivity of the system.
Cell C5 outputs the value in cell C1 on analogue output 1.
Cell C6 outputs the value in cell D2 on analogue output 2.
Cell C7 outputs a value of either 0 or 10 on digital output 1 depending on the state of cell D5.
Cell C8 outputs the value in cell D5 on digital output 2.

Column D is where the logic control algorithm has been implemented. The basic functionality is such that if digital input 1 becomes logic 1 a temperature measurement cycle is initiated. The temperature signal is input on analogue input 1 and re-transmitted on analogue output 1. At the same time the spreadsheet is comparing the current input value with a record of the maximum temperature received so far. This maximum temperature value is constantly output on analogue output 2. If the input temperature falls below the current maximum value by the amount set in cell C3 - the sensitivity - the cycle is ended and no further variation of the analogue input is acted upon until a new cycle is initiated by activating digital input 1. For the entire duration of the measurement cycle a 1Hz pulse train is output on digital output 1.

11.3 Dosing Application

This example is used in a dosing application.

	A	B	C	D
1	SAI(1,VOLT0.10,0,14)	pH Input Value:	ALG.IN1	310
2	SAI(2,VOLT0.10,0,14)	Setpoint:	ALG.IN2	TIMER1
3	SAI(3,VOLT0.10,0,600)	Gain:	ALG.IN3	T1.STATUS
4				D7 OR D10 OR D11
5	SAO(1,VOLT0.10,0,14)	PID Loop:	PID(1,C1)	RST.TIMER(1,D4)
6	SAO(2,VOLT0.10,0,14)	Low Time:	300-(2.9*C5)	RUN.TIMER(1,D1)
7		High Time:	1+(0.198333*C3)	(D14=0)AND(D2>C6)
8	SDO(1,LOGIC)			(D14=1)AND(D2<=C7)
9	SDO(2,LOGIC)	pH Output:	ALG.OUT(1,C1)	
10		Setpoint Output:	ALG.OUT(2,C2)	(D14=1)AND(D2>C7)
11	SET.PID(1,C2,4,0,0)	Alarm Condition:	(C1<4.0)OR(C1>10.0)	(D3=3)OR(D2>300)
12		Alarm Output:	DIG.OUT(2,C11)	
13				
14			Pulse Mode:	IF(D7 OR D8,1,0)
15			Pulse Output:	DIG.OUT(1,D14)

Three analogue inputs are used to get the pH value of a substance, the setpoint and the gain of the system. An alarm is output on digital output 2 if the pH value is less than 4 or greater than 10. The pH value and the setpoint are re-transmitted on analogue outputs 1 and 2 respectively. Digital output 1 is used to control the rate of dosing. If digital output 1 is logic 1 dosing will occur. The amount of dosing is controlled using a PID control loop and a timer. The PID loop acts upon the current input pH value and produces an output of 0% to 100%. This is then used in conjunction with the input gain value to set the number of seconds that the dosing should last.

Column A contains all the initialisation of analogue and digital channels, and also the SET.PID function which sets PID loop 1 to have a setpoint set by analogue input 2, a proportional band of 4 and integral and derivative times of 0.

Column B contains comments.

Column C is used to gather the analogue input data, set the analogue values, calculate the PID loop value and the dosing times, and check for the alarm conditions.

Column D sets the timer 1 to run for up to 310 seconds depending on the various timings calculated using the PID in column C. The actual digital output used for the dosing is set in column D also.

12.0 Hardware Specification

Operating Temperature 0 to 40°C

Specification @ 20°C.

12.1 Input Channels

12.1.1 Analogue

Ranges:	0-10V 0-20mA, 4-20mA	
Input Impedance:	Voltage Current	100K ohms 300 ohms
Resolution:	10 bit or 0.1% of F.S.D.	
Accuracy:	± 0.15% of F.S.D.	
Temperature Stability:	100ppm / °C over operating range.	
Transmitter Power Supplies:	20V ±15% dc @ 22mA for each channel.	

12.1.2 Digital

Modes:	Logic/Status, Frequency Measurement.
Working Frequency Range:	0 - 1kHz.
Time Period / Frequency Resolution:	1.333 μsec.
Resolution Error:	0.0013% @ 10Hz. 0.13% @ 1kHz.
Temperature Stability:	50ppm over operating range.
Wetting Voltage:	22V dc at 5mA.

12.2 Output Channels

12.2.1 Analogue

Ranges:	0-10V 0-20mA, 4-20mA.
Resolution:	11 bit or 0.05% of F.S.D.
Accuracy:	± 0.2% of F.S.D.
Temperature Stability:	150ppm / °C over operating range.
Load Impedance:	For Current ranges maximum load is 500 ohms. For Voltage range minimum load is 1000 ohms.

12.2.2 Digital

Modes:	Logic/status or Frequency.
Relay Rating:	240 V a.c. at 3A.
Minimum Frequency:	1.75e-4 Hz.
Maximum Frequency:	50 Hz (Relay Output). 1000Hz (Open Collector Output).
Time Period / Frequency Resolution:	2.666 μsec.
Resolution Error:	0.0026% @ 10Hz. 0.26% @ 1kHz.
Temperature Stability:	50ppm over operating range.

12.3 Communication Channels

1 RS232 (1 RS485 optional).

Baud Rate:	9600 Baud.
Data Bits:	8.
Parity:	None.
Start Bits:	1.
Stop Bits:	1.

12.4 Isolation

Full 3-port galvanic isolation to 500 V d.c.
(Inputs and RS232 comms / outputs / power supply / RS485 Comms.)

12.5 Power Requirements

24V d.c. +/- 10%. at up to 350mA (maximum load conditions).

12.6 Dimensions

140 x 142 x 32 mm (H x D x W).

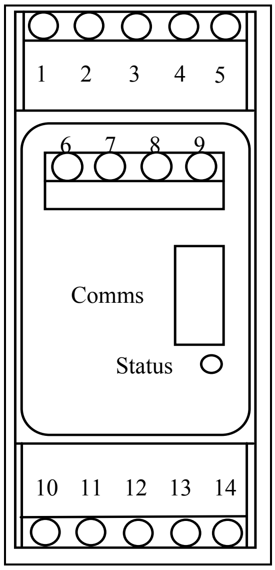
12.7 Mountings

The instrument is DIN rail mounting.

It should be fitted with the slats at the top to allow for airflow through the unit.

13.0 Connection Details

13.1 Digital Outputs



1. Digital Output 1 Common
2. Digital Output 1 N/O
3. Not Connected
4. Digital Output 2 Common
5. Digital Output 2 N/O

13.2 Analogue Outputs

6. Analogue Output 1 -ve
7. Analogue Output 1 +ve
8. Analogue Output 2 -ve
9. Analogue Output 2 +ve

13.3 Digital Outputs

10. Digital Input 1 Ground
11. Digital Input 1
12. Not Connected
13 Digital Input 2 Ground
14 Digital Input 2

13.4 Analogue Inputs

15. Analogue Input 1 Ground
16. Analogue Input 1 +ve
17. 24V Transmitter Supply
18. Analogue Input 2 Ground
19. Analogue Input 2 +ve
20. 24V Transmitter Supply
21. Analogue Input 3 Ground
22. Analogue Input 3 +ve
23. 24V Transmitter Supply

13.5 Power Supply

24. Power Supply Input -ve
25. Power Supply Input 24V