Introduction

Flex32 is the software used to program the Colter Products' range of programmable controllers including the FMT-100, FMT-200 and BIS-100.

The main features include:

- Graphical Ladder Programming
- Text based high level language
- Source level debugging
- Syntax highlighting in both programming languages
- Graphic mimic screens
- Facility Useage screen
- Easy Project documentation
- Site License

Minimum System requirements:

- Microsoft[®] Windows[™] 95/98/NT.
- SVGA monitor supporting 800 x 600
- 5MB Hard Disk space
- 16MB Ram
- Serial Communications port for connection to controller
- Mouse
- CD-Rom

High Level Instruction Language

The High Level text based instruction language is designed to be a simple high level language easily learned by anyone familiar with 'C', Basic, Pascal or any other programming language.

- The text editor is used to write instruction language programs and features colour syntax highlighting and multiple undo steps.
- For debugging the on-line mode of operation can be selected. Break points can be set and the current state of facilities such as inputs, outputs, registers etc can be displayed.
- Individual modules can be stopped, started, single stepped, or run to breakpoints.



Screen shot from the Text Editor



Pop-up menu in instruction editor

When in the instruction editor if the right mouse button is pressed you will be presented with a pop-up menu:



```
else if Tag State = WAIT FOR TAG
```

- If 'Symbol Names' is selected then a list of all the symbol names assigned for the project is displayed. Double clicking on an entry will paste that name into the editor at the current cursor location.
- If 'Functions' is selected from the pop-up menu then a list of all the instruction language functions will be displayed. Double clicking on an entry will paste that function into the editor at the current cursor position.
- If 'Keywords' is selected from the pop-up menu then a list of all the instruction language keywords will be displayed. Double clicking on an entry will paste that key word into the editor at the current cursor position.

Total Recall

Total Recall is a new feature in Flex32 it allows you to store your entire project in your FMT / BIS memory when downloading the executable code:

- Provides you with the option to store your entire project in your FMT / BIS memory alongside the controller's executable code. This means that at a later date the project can be uploaded from the FMT / BIS, into a computer with Flex32, even though the project is not stored on the computer.
- Optional password protection of project stored on FMT / BIS to prevent unauthorised uploading of stored project.
- Entire project is stored using Total Recall project configuration, source code, ladder code, symbol names etc.



Group

Ladder Logic

The Ladder Logic editor enables you to write and debug programs in the familiar style of ladder logic.

The ladder editor enables access to all of the FMT's or BIS-100's advanced maths and communications facilities as well as the simple contacts, coils, flags, counters etc. The ladder editor features graphical ladder display with colour highlighting of the facility type being used. The On-line mode showing current ladder condition to enable easy debugging.

Ladder functions include:

- Inputs: on, off, edge, fast edge.
- Outputs: on, off.
- Timers: on-delay, off-delay, pulse.
- Counters: count-up, count-down, pre-set, clear.
- Comparators: greater, less-than, equal to.
- 16-Bit and 32-Bit operations:

Add, Subtract, Multiply, Divide, Square, Square Root.

And, Or, Exclusive Or. Negate, Logical shift, Rotate.

- Binary to BCD, BCD to Binary.
- Move:

inputs, outputs, and flags and analogues to/from registers.

- 16-Bit or 32-Bit registers to/from 16-Bit or 32-Bit registers.
- Data Handling with flags and registers: Shift Registers, Stacks, and FIFOs.
- Serial Communications:

Send out Text string.

Receive number, text, data.

Compare received text with stored text.

🖳 Flex32: Demo	_0×
File Edit Search Bookmark View Compile Diagnositos Options Window Help	
Project Module Run., Print., Exit Project Text Symbol Jacobs Manitor Minic	
Project Configuration	
New Save Saves Pint Lice	Save Save4s Print Cut Copy Paste Delete Change Compile On-Line Off-Line Close
328ITOPS.LAD	0
Serve SaveAs Print Cut Copy Paste Delete Charge Compile On Line Olificine Dese	1Smple
0 - *** Examples of using the 32-BIT OPERATION functions ***	
2 Multiply seconds by minutes, put the answer in W10	10002 Q0001 F0000
3 Source Minute Dest. W0000	
4	7 F0000 40000 4000 0000
5 Ste A W000	FOOT
6 - Dest W0010 -	
7 Find the square of seconds times minutes, put the answer in W11	
A Straight A	10 Not to simple
STACK LAD	
	14
Save SaveAs Print Cut Copy Paste Delete Change Compile On-Line Off-Line Close	
	臣田記井山 程川上下 ③·123 ~18 (1·18·1w·雪 つつ涼 9
10 Source Second	Normally Closed Contact
11 Num. K0001	
12Start R0300	
臣田記書書 四四王下 ③·123 三田 ()·R·W·9 00※ 9	
Move	
Prinet Demo Statu	

Screen shot from the Ladder Editor



Page 3.

Symbol names for all facilities

Flex32 allows names to be assigned to all facilities such as inputs, outputs, registers, timers etc. A short name of 6 characters is used in the Ladder diagrams while the instruction modules can use the short name or a long name of up to 12 characters. Symbol names are assigned to facilities using the symbol name editor.

Save	Print	0 m Clear Find	🖙 🚟 🛄 Load Multi Close	
Inputs	Outputs] .	Analogue In	Analogue Out Registers Wide Regs. Flags Tin	ners Text S 💻
Facility	Short	Long	Comment	
R0100	Mes.No		Text message number to be sent out	
R0101	Addrss		Smarteye reader address	
R0102	Rep.Ln			
R0103	Er-Cnt		count of sucessive errors	
R0104	SEA-St		SEA status register	
R0105	D.Mode		Diagnostic mode	
	-		EX- Contraction of the Contracti	

Screen shot from the Symbol Name Editor

Facility Monitor

The facility monitor allows values of facilities to be displayed in decimal number, text, voltage*, current*, or ASCII values.

Facilities can also be set or forced to as part of the debugging process.

* Voltage and current display relates to 10v and 20mA analogue inputs.

acilities	Allocate	ed RAM	
다 Add	📼 Rem. Cl	F F ear All Force	F F T V · □ Frc All Style Close
acility	Short	Long	Value
20			Off
21			Off
5			Off
6			Off
7			Off
8			Off
9			Off
R400			00000
R401			00000
3402			00000
n kon	1	17	00000

Screen shot from the Facility Monitor window



Mimic Screen

The Mimic Screen shows a live, animated graphic of the current state of the controller you are connected to.

All analogue and digital I/O are displayed as is the built in display of the FMT-200.

Facilities can be set and/or forced from this screen.



Screen shot from the Mimic Screen (FMT-200D shown)

Flex32 Programming Softwar

Instruction module execution options

Modules can be executed in several different ways and these are set up in the instruction editor. The different ways of module execution are:

One instruction step per loop (default): In this mode, one 'step' of the instruction module will be carried out during each execution loop. This is the default mode of operation and will be the best option for the vast majority of modules.

One step at fixed time interval: In this mode you will be prompted to enter the time interval between steps. The time interval can be 10mS to 1second in 10mS intervals. There are several reasons for selecting this option...

- If you have an unimportant section of code that you want to execute slowly without using much processor time. Select only a few steps per second.
- If you have some code that is very important and must be executed frequently then select a high number of steps per second.

Fixed number of steps per second: In this mode you will be prompted to enter the number of steps per second to be executed from this module. The number of steps per second can be between 1 and 100.



Colter Group **Execute module on input interrupt:** In this mode the entire module will be executed at the exact moment that the specified input comes on. It should be remembered that executing large sections of code on an input interrupt will reduce the capacity of the FMT to process the other code within the project. You are limited to executing 20 steps of code in one interrupt before the firmware will raise an 'Input Interrupt overrun' error.

Screen shot showing the Instruction Module Control Options box



Setting the module execution: While in the instruction module editor click on the 'Control' button on the instruction editor toolbar. You will then be presented with a box as in the screen shot above from which you can select how your module is executed.

Text String Editor

Text strings to be sent from the ports are set up using the text string editor. The text, data in register etc. to be sent is entered next to the text string number which is to be controlled from the program. The text string is sent out using the 'text' command in your program. The text string can be assigned a symbol name which can be used in your program.



Colter Group

Screen shot from the Text Editor

Text S	tring Ec	litor								o	
Save	🞒 Print	 Control 	ASCII	ian ≁ Value	₿ • Buffer	୍ଦ ି Insert	Bester DateTime	∑ • Check	हता , Display	Close	
Text String	Short	Long		Text							
Tx0000				%#0CSn	narteye GE	M80 %#	0A Concer	itrator			
Tx0001				%rRdr%F	R0120R31	ag:%R0	118R5%IRc	r%R0121	R3 Error: %	R0119R2	
Tx0002											
Tx0003				%rReade	%rReader Num: %R0106R3%IT ag %R0107R5 Err %R0108R2						
Tx0004				%r Entry	%r Entry %R0130RZ %I Data %R0131RZ %s						
Tx0005				%n%DT7	%n%DT7 %DT0 Reader %R0120R3 Tag Number %R0118R5						
Tx0006				%n%DT7	%n%DT7 %DT0 Reader %R0121R3 Error %R0119R2						
Tx0007				%r Initiali	%r Initialising%IPass %R0122R1 S.E.A. %R0123R2						
Tx0008				%rReade	%rReader Num: %R0106R3%IS.E.A. Off-Line						
Tx0009				%rReade	%rReader Num: %R0106R3%IS.E.A. No-Resp.						
Tx0010				%#0C							
Tx0011											
•											×.

Facility useage screen

The facility useage screen shows you which facilities of the FMT are being used and in which module they are referenced. This information simplifies the process of keeping track of facilities that have been used and can be printed out for project documentation. Note that the Facility useage screen is only available once a 'test compile' or 'compile and download' has been completed.

Screen shot from the Facility useage screen

🚽 Facilit	ty Useage								×
	96 - 311 - 94 - 94 - 94 - 94 - 94 - 94 - 94 -	හි Find	Print	Exit					
Facility	Short	Long			Multiple	Module	Function	Read/Write	
W0033					1	ins-demo	Rotate Left (Write	
W0034					Rotate Righ	nt (32-bit)		Write	
W0051					2	ins-demo	Move	Write	
W0060					2	ins-demo	Move	Write	
W0060					1	ins-demo	RAM Write	Read	
W0060					1	ins-demo	Flash Write	Read	
W0060				_	2	ins-demo	Move	Write	
W0061					1	ins-demo	RAM Read	Read	
W0061					1	ins-demo	Flash Read	Read	
T0000					1	TIMERS	On-Delay Tim	Write	
T0000					1	TIMERS	Load	Read	
T0001					1	TIMERS	Off-Delay Tim	Write	
T0001					1	TIMERS	Load	Read	
T0002					1	TIMERS	Pulse Timer	Write	
T0002					1	TIMERS	Load	Read	
T0010					1	ins-demo	Wait For	Read	
T0010					1	ins-demo	On Delay Tim	Write	



Easy project documentation

Flex32 allows you to easily document you project using it's print center. You can select exactly which parts of the project to print:

- Project Summary
- Ladder Modules
- Instruction Modules
- Symbol Names (selected by facility)
- Facility Useage
- Text Strings

	Printer Setup	1	
Summary	Symbol Names Input Output	Ladder Modules	
Facility Useage	In Indiana Indian	COMP SETRESET JUMP MOVES	
Text Strings	Analogue Input Analogue Output Internal Register Internal Flag Text String	Instruction Modules	
Internal Registers	Label Comparator Set/Reset Latch Input Status Register Output Status Register Comparator "Greater-Than" Flag Comparator "Less-Than" Flag Comparator "Equal-To" Flag		

Screen shot from the Print centre

Simple project configuration

Flex32 allows you to configure you project with the greatest of ease. Various configuration options can be set from within the project configurations screen. These options include:

- Communications ports setup
- High speed inputs setup
- Flash card setup
- I/O update setup
- Fieldbus module setup (BIS-100 only)
- Allocated RAM



ex32 Programming Software

🚊 Project Configura	tion			
New Save Save	As Print Close			
Project Name: Dem	0			
Project Preserve Co	omms High Speed Flash	Card RAM 1/0 Update	e Fieldbus I.R.	
FMT Type	Station Numbe	ar		
FMT 200D	· [÷			
Description				
Demonstration prog	gram			
Available Ladder	Project Ladder	Available Instruction	Project Instruction	
	STATES	INS-DEMO IST	modules	
32BITOPS.LA	Add 32BITOPS		Add	
COMP-TEX.L/	SHIFT			
FIFO.LAD	Remove STACK	F	emove	
JUMP.LAD	D GET-NUM		в	
MOVES.LAD	New COMP-TEX		New	
Apply C	ancel			
	1			

Screen shot from the Project configuration screen

Project Name: De	emo			
Project Preserve	Comms High Port 0 Port 1	Speed Flash Card RAM	1/0 Update Fieldbus I.R.	
Baud Rat C 75 C 110 C 300 C 600 C 1200	 2400 4800 9600 19200 	Data Bits Parity © 5 © None © 6 © Odd © 7 © Even Stop Bits Catoon Num- © 1 0	Protocol © User Code © Modbus RTU Slave © LinkLine © PC-Manager © Saab-Link © Baumer © LinkLine Plus © C24 Protocol 1	

Screen shot from the Project configuration screen showing the communcations settings

Allocated RAM: The FMT contains battery backed memory which is used to store your program. If you wish you can use some of this RAM to hold data instead of program code, but obviously the maximum size of your program will be smaller. The amount of RAM to be used is setup in the RAM page of the configuration screen.

I/O Update: In this page you can set the number of digital inputs and outputs used and the time between updates. You can also set the number of analogue inputs and outputs used and the update time. The less inputs and outputs set to be used and the longer the update time then the more instruction per that second will be executed. This can be useful when code within the FMT is required to be executed at great speed. For more information on optimising FMT performance please see the separate data sheet.

Fieldbus setup: (This is only applicable to the BIS-100). In this page of the project configuration screen you can set the start of the block and the number of registers that are read from and written to the Fieldbus that the BIS-100 is connected to. You can also set the update time of these registers. Again the longer the time between updates the more instruction per second that will be executed.



Colter Group Data Sheet Issue: 1.21 Date: 10 May 2005

Order Codes

Part Number FLEX32

COLTER GROUP COLTER PRODUCTS LIMITED

UNIT 7, ZONE C CHELMSFORD ROAD INDUSTRIAL ESTATE DUNMOW ESSEX CM6 1HD

Telephone: + 44 (0) 1371 876887 Fax: + 44 (0) 1371 875638

E-Mail: sales@coltergroup.co.uk Web Site: www.coltergroup.co.uk

© Copyright 1999

Colter Flex32 is designed and manufactured in Great Britain by Colter Products Ltd. Colter Products reserve the right to amend these specifications and the user is asked to check the validity of the data sheet prior to use



Group

Page 10.