Motion Application Program
CNC

User Manual

Table of Contents

Health and Safety.

1. The user should ensure that safe working practices are adhered to and that correct Personal Protective Equipment (PPE) is used at all times relevant to the nature of work undertaken and the type of machinery being operated.
2. Before powering up or using MAP CNC, the user should ensure that the machine being operated is clean, debris free and is being operated in a safe manner as required by the Laws, Rules, Regulations and Best Working Practices in force for the Country in which the equipment is being operated and that all safety devices are in undamaged and correct operating order.
3. This software (MAP) should not be used by personnel that have not been trained in the use of such software or the machinery it is used on.
4. TRM cannot be held liable for the miss-operation or incorrect programming of the system be it operating program or setup parameters.
5. Training and operation of machinery is solely the user's/owners responsibility and appropriate Risk Assessment records should be kept at all times.

Version Updates

Registers

Number of registers available increased from 32 to 256.

Sticky Menus

Menu Selections are now 'sticky' saving time on navigation and data entry.

Canvas

Canvas facility added to allow personalisation of screens.

AC Drive Delay

Timer delay added to allow drives to power up and avoid 'kick' in the motor.

User Interface

Overview

Motion Application Program (MAP) CNC is a new way to programme motion control systems. The programme consists of a database manager and a motion control kernel. The database is used to store commands and position data for the motion control command sequence. The end user is presented with a user friendly front end with which to manipulate the database. The database can contain up to 65,000 different commands and due to the structure of MAP, it is a simple matter to change the system to suit another application.

Map is a very flexible language written in 'C'. It is easily adapted to other applications, as each of the user functions is written as a case statement. Adding commands is a quick and easy matter and for custom requirements we can 'roll up' a number of commands into a single macro command.

The user interface consists of an icon driven interface with scroll bar menus. The motion editor has a user friendly interface and has a pull-down menu system. Upon accessing the commands menu for example, the user is presented with a list of possible commands, the up and down arrow keys are then used to highlight the commands the user wishes to insert into the programme. Once the command is highlighted it may be entered in to the database by pressing the Enter key, then by using the right or left Arrow keys the user simply moves the cursor to the information they wish to edit. The system is programmed to accept data in a strict format which prevents the end user from entering erroneous data. If the end user attempts to enter data incorrectly the system will prompt advising the user of the data limits.

From a users point of view, a major benefit of using MAP is that no technical knowledge is required in order to programme the system. Using the MAP database principle any advanced application programme may be written such that the program can be run from a simple operator interface.

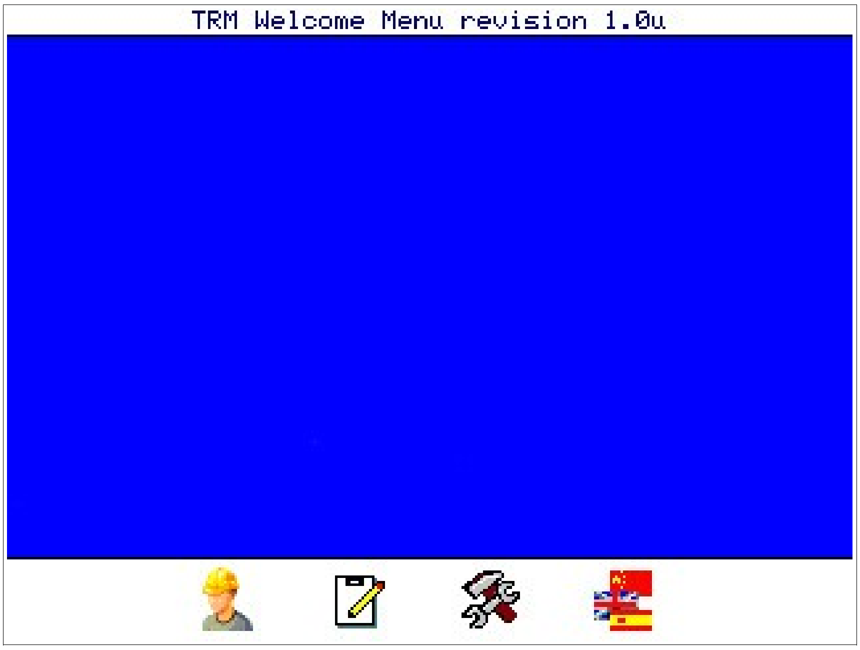
A MAP system requires calibration information in order to operate to standard measurements. The installer must programme the limits for the system which include motor and encoder details, axis dimension limits and operational options.

Please note that most menu's have a line number associated with that item, However that number is not shown for clarity. The top item will be 0.

For example bring up the commands menu and press 3 and then 6, the system will insert a 'Print_Reg' command. This can be a great time saver for frequently used commands.

Opening Screen

As shown below the opening screen for MAP is a very simple icon driven interface. At the top of the screen the revision number is shown so that when upgrades become available the user can check to see if there are any new features that may be useful and can then decide to upgrade if necessary.

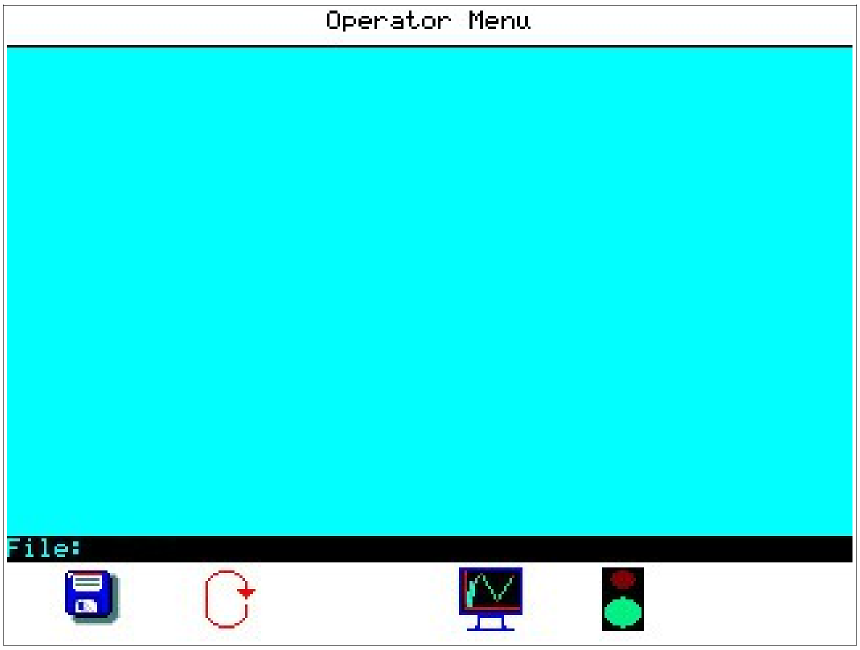

TRM Welcome Menu revision 1.0u

At the bottom of the screen are four Icons which take you into the Operator Interface, Edit/Create Screen, System Tools Menu and the Language Picker. On non-touchscreen controllers, pressing the corresponding button below or to the side of the Icon will activate that option.

The central blue area can display a JPEG image 320 x 200 pixels of the machine, company logo or similar.

Operator Menu

When the Operator menu is selected a simplified run screen is displayed. This gives the operator the information required to run the machine without the ability to edit and change machine settings or program flow.

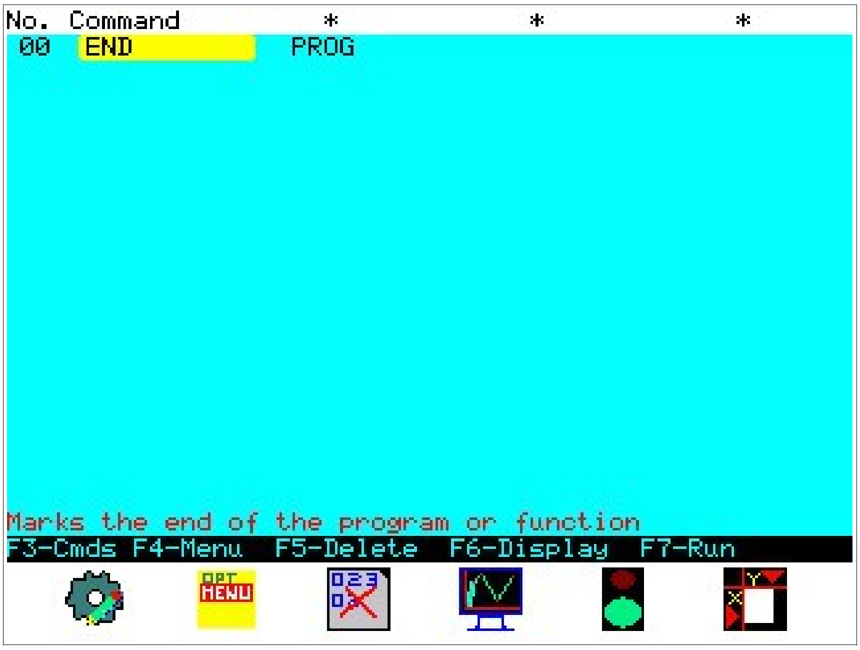


Operator Menu

File:

Edit/Create Menu

When the Edit/Create menu is selected the controller will display the screen below:



```
No.  Command        *            *            *
 00   END          PROG
```

Marks the end of the program or function

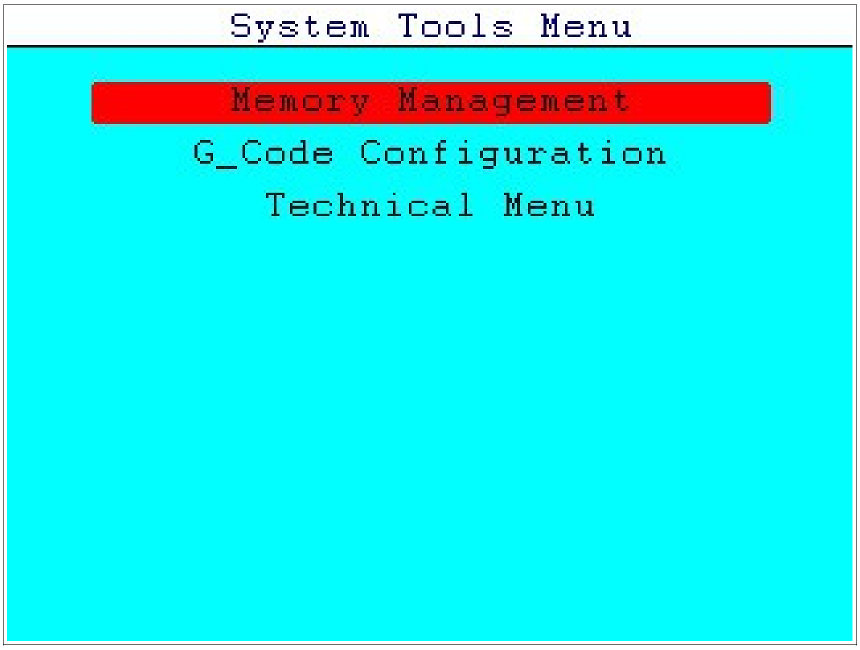
F3-Cmds F4-Menu F5-Delete F6-Display F7-Run

The text in the box at the top of the screen is intended to help with data entry as each command has it's own header. 'No.' represents the line number of the program, 'Command' is the name of the MAP command used and *,*,* are the three heading that will change based on the command. Where a valid command is selected and a column displays '*', no data should or can be entered.

The menu has a hint which is displayed in the information line, this hint changes to indicate to the operator what information is required.

System Tools Menu

All the technical information the controller needs to run the machine is stored here and most menus are password protected for security. For more information please review the MAP CNC Installation Manual.



Language Menu

Selecting the Language option will display a pop-up box on screen with a list of languages available. Should your language not appear in the list please check with TRM's sales or support teams for updates or electronic manuals in that particular language.

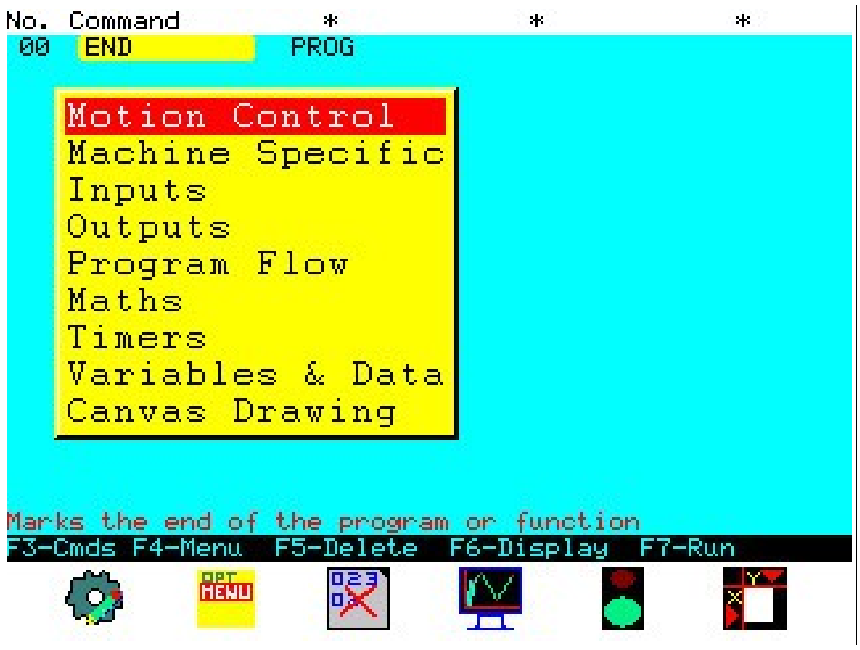
Scroll to the language required and press enter, the box will disappear and the system will now work in the chosen language. There is no need to re-boot the controller.

MAP Commands

Overview

The commands that can be used for a MAP program are organised into categories to make them easier and quick to find. A new feature in MAP CNC is a 'sticky' cursor, This was implemented so that returning to the same command or category is quick and eliminates many key presses from the previous versions of MAP.

Press key to reveal the commands pull down menu.



```
No. Command        *            *            *
 00  END          PROG
```

Marks the end of the program or function

The desired command is selected by moving the cursor to the line required using the arrow keys. Once the desired command is highlighted pressing the ENTER key will cause the command to be entered on the current highlighted line. Any existing commands will be moved down one position. If the menu has been selected in error pressing the Esc key will exit the menu without changing the database.

After entering a command the controller will now display the new command in the database. Move the cursor to the first data location (I.E. to the right of the command) and enter data using the number keys followed by the Enter or right arrow key. Some commands require data to be entered for two axis or that other values be selected by using the enter key or +/- keys, this will be indicated by a heading above the column and an information line towards the bottom of the screen. If a value is entered that exceeds the system maximums a message box will be drawn, this will prompt the user with the maximum data value. When the user acknowledges the error, the system will enter the maximum value in the database.

If a typing error is made, the left hand ARROW key can be used to delete the last character typed.

Following are a list of the available categories:

Main Menu
Motion Control
Machine Specific
Inputs
Outputs
Program Flow
Maths
Timers
Variables & Data
Canvas Drawing
Data Recording

Selecting an item from the above menu will display one of the following lists of commands:**

Motion Control Commands
Arc
Do Move
Helix
Home
Index
Move
Origin
Rapid_Move
Revolve
Speed
Stop Axis
Torque Lim

** *Note: The actual list of commands will depend on the model of controller used.*

Machine Specific Commands
Air Op Drill
Coolant
Feed Rate
G-Code
Park & Power Down
Planes
Spindle
Tool

Input Commands
ADC (Analogue In)
Import
Input
Keypress
Question
Read File
YesNo Box

Output Commands
Analogue Out
Change Op
Export
Led
Message
Output
Print Register
PWM (PMC4 Only?)
Write

Program Flow Commands
Begin
Branch not true
Call MAP Program
Call Subroutine
Dec Jump
Do
End
Exit
If
Jump
Main_Section
Repeat_If_True (PMC4 only)
While

Maths Commands
Average
Cosine
Find Maximum
Find Minimum
Logic Operators
Math
Raise to the Power
Sine
Square Root
Sum
Tangent

Timer Commands
Halt
Timer
Wait

Variables & Data Commands
Copy Child Registers
Load Register
Load Counter
Load File Data
Load Multiple
Load Parameter File
Load System Vars
Parse File
Register Type
Rewind File Pointer

Canvas Drawing Commands
Draw Canvas
Clear Canvas
Canvas Draw Pointer
Filled Circle
Draw Rectangle
Set Colours
Draw Line
Draw JPEG Image
JPEG Image Size
Work Piece Size

Data Recording
Data Logger Task
Start Data Recording
Stop Data Recording

Motion Control Command Definitions:

ARC

This command allows the user to execute arcs or complete circles between X and Y axis. ARC must never be the first command in a program, always precede with a move, index or similar motion command. The user has to specify the centre point (X,Y) and the angle of the circle, a positive angle will be a counter clockwise arc and a negative angle will be a clockwise arc. The current position will be the starting point of the circle.

No	Command	Data1	Data2	Data3	Explanation
00	MOVE	110.00	100.00		Turn ON the LED's.
01	ARC	0.00	-5.00	90.00	The ARC will start at position X= 110.0, Y=100,00. The centre point is always specified relative to the current position with Data1 and Data2 parameters. Therefore the centre point is located in X: 110.00 + (0.00) = 110.00 = X_c and for Y: 100.00 + (-5.00) = 95.00 = Y_c The arc will have an angle of 90 degrees (Data3)
02	:				
03	:				
04	:				
05	:				
06	END				End of program.

DO_MOVE (.......WHILE)

This command is a conditional Move command and is similar to the Do … While command except that it includes a position for each axis to move to. It will move the distance set by the user unless the condition is not true. Note that this command requires the user to "close" the command by entering a WHILE command.

No	Command	Data1	Data2	Data3	Explanation
00	DO_MOVE	100.00	0.00		The axis 0 (Data1) will start to move to 100.00 and axis 1 (Data2) will remain in 0.
01	LOAD	R05 IP.b	2		Loads the Register 5 (R05) with the logic value of the input 2 (Data2).
02	WHILE	R05 ==	0		This is the end of the DO_MOVE-WHILE command.
03	MOVE	0.00	0.00		Moves axis 0 and axis1 (Data1 and Data2) to zero
04	END				End of program.

The Loop will repeat as long as the condition stated here is met or the DO_MOVE distance has not reached its target set (100.00 in this example). In this example, if the register R05 is loaded with '0' the execution of the program will jump to the DO_MOVE command and execute all the lines in between, in this case only the LOAD command. The == parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears with multiple options.

HELIX (PMC Series Only)

The Helix command is a 3 axis command moving X and Y in a circular movement whilst changing Z in the required direction at a rate to give the required step. This is also used by the Gcode G2 clock wise arc command. There are four parameters to the command to determine the required result:

Data[0] = x constant or register (float) relative X centre point

Data[1] = y constant or register (float) relative Y centre point

Data[2] = Angle constant or register (float) angle of rotation

Data[3] = Height constant or register (float) the relative height change

The controller will automatically calculate the speeds based on the angle and Z final position. A 180° angle will only give half a rotation, 360° a full rotation and 720° two full rotations. Increase the angle to the desired number of revolutions based on the finish required.

HOME

The HOME Command allows the user to re-calibrate to the home sensor at any time. If there is no Home sensor allocated the current position will be set to zero. If a sensor has been set then the display will show 'setting neutral' and the axes will all move back to the home switches.

Usually the MAP system will home on first run and then retain this setting and remain accurate however if a system is liable to change due to temperature, wear or some other means, a Home command can be used to reset the zero point as often as necessary.

INDEX

This command is similar to the MOVE command however it is a relative position move.

Therefore if the command is INDEX 5 , the axis will advance 5 units from the current position. The data entered is the same for both Absolute and Relative axis as it is an increment from the current position not from 'Home 0,0'.

No	Command	Data1	Data2	Data3	Explanation
00	MOVE	100.00	200.00	0.00	Move axis '0' to 100 mm and axis '1' to 200
01	INDEX	75.00	10.00	0.00	Index advance 75 mm from the current position for axis '0' and 10 mm for axis '1'
02	END				End of program.

MOVE

The move command as it might suggest causes the controller to move one or more axis. All moves are automatically interpolated moves. Since an axis will only move when the system encounters a MOVE (or INDEX) command there must be a difference in the old/new positions.

Therefore it is simple to move only one axis if required by changing the position data for only one axis between two or more MOVE commands. The Move command works by taking the current position and moving all the axis in an interpolated move, therefore if an axis has its current and new positions programmed the same, then the axis will not move, only the axis which have a new and different destination will move.

The movement will take place in the following sequence:

1. The controller will calculate the distance of each axis relative to its current position.
2. The controller will then work out which is the master axis (the master axis is the one which is moving the greatest distance), where more than 1 axis are Masters because all are moving the same amount, the controller will use the lowest number as the master i.e. AX0, AX1, AX2.....
3. The controller will then ensure that none of the axis will be running at greater than its maximum speed, the speed therefore, is the maximum speed of the slowest axis. The move Command may include positive or negative moves if the axis has been programmed to accept negative moves in the calibration menu.

No	Command	Axis 0	Axis 1	Axis 2	Explanation
00	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200. Please remember that MOVE command is a Linear interpolation command.
01	MOVE	150.00	300.00	0.00	Move axis '0' to 150 and axis '2' to 300. Please remember that MOVE command is a Linear interpolation command.
02	END				End of program.

Note: in the example above we assume that the axis are set as 'Absolute' in the technical menu, the data could be mm, inch or degrees etc.

Absolute Move Example:

No	Command	Axis 0	Axis 1	Axis 2	Explanation
00	MOVE	100.00	200.00	0.00	From start at 0,0 both axis move
01	MOVE	100.00	300.00	0.00	Axis 0 holds position while Axis 1 moves another 100 to 300.
02	MOVE	200.00	300.00	0.00	Axis 1 holds position while Axis 0 moves 100 to 200.
03	END				End of program.

If the value for an axis between two move commands does not change then there will be no movement.

Relative Move Example:

No	Command	Axis 0	Axis 1	Axis 2	Explanation
00	MOVE	100.00	200.00	0.00	From start at 0,0 both axis move
01	MOVE	0.00	300.00	0.00	Axis 0 holds position while Axis 1 moves another 300 to 500.
02	MOVE	200.00	0.00	0.00	Axis 1 holds position while Axis 0 moves 200 to 300.
03	END				End of program.

If the value for an axis is not zero on the second move command then the axis will move by the amount programmed relative to where the machine is currently positioned.

Origin

This command is used to offset the position of an axis. The user has to select the axis and the value to offset that axis. The offset value can also be stored in a register and use that register in the offset value. In order to select a register the user has to position the cursor on the offset column and press the '+' key, then write the number of the register from 0 to 255 (there are 256 registers available).

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10. The VorR (Value or Register) parameter is accessed by placing the cursor on top of the data1 or data2 of this command and pressing ENTER. A pop up window appears. In this example we choose "Value or Register" (VorR) which indicates that the data2 has to be a value or a register. Please note that the data2 is attached to the option selected.
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
02	MATH	R05 *	2		The value stored in the Register 5 (R05) will be multiplied for 2 and the result will be stored again in R05. The * parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears.
03	ZERO Pos.	1 Val	5.00		The axis 1 (data1) will be offset 5 and the axis will remain offset 5 until a new value is used. The Val parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears with 2 options. If the Man(Manual) option is selected the program will run normally but at this command the user will be required to offset manually the axis specified in data1 (e.g.: axis 1)
04	MOVE	50.00	70.00	0.00	Move axis '0' to 100 mm and axis '1' to 200
05	END				End of program.

Rapid Move

Moves the programmed axis at it's maximum set speed which is setup by the machine installer in the defaults menu. A position to move to can be entered in each of the available axis columns. Test the process without tooling initially to ensure no collision or damage will occur. The alternative to this command is to set a 100% Speed command, a Move command followed by another Speed command to return to the required system speed.

REVOLVE

This command allows an axis to continuously revolve at a desired speed. In order to do that the axis has to be set as a relative axis without limits in the installation menu (see installation manual). The command has 3 settings, the first one is the number of axis, the second one is to select if the axis will be affected in a move command in another part of the program or not and finally the speed set to revolve.

In order to have access to the 2nd 'type' parameter the user has to position the cursor over the Command name and press 'ENTER', this will give the user 2 options: "Relinked on Move" and "Solo Axis". By selecting "Solo Axis" the axis will remain revolving when a MOVE command appears in a different part of the program. Selecting “Relinked on Move” will mean that the axis revolves and executes the rest of the program, when a move command is processed the axis is then moved by the distance programmed.

This latter option may be useful for conveyor type situations where, the conveyor runs until eventually a product is detected via an Input command, the process then has to move a set distance for an operation to be performed and once complete starts revolving again until the next product comes along.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	INDEX	75.00	10.00	0.00	Index advance 75 mm from the current position for axis '0' and 10 mm for axis '1'
02	MATH	R05 *	2		The value stored in the Register 5 (R05) will be multiplied for 2 and the result will be stored again in R05
03	REVOLVE	1 Solo	85.00		The axis number 1 (data1 column) will revolve continuously at 85.00%. Please note that the user can select between SOLO or RELINK ON MOVE [....] by pressing the ENTER key when the cursor is on one of the data columns of the REVOLVE command.
04	REVOLVE	0 ...	R05		The axis number 0 will revolve continuously at the speed set in R05.
05	END				End of program.

SPEED

This allows the user to adjust the speed as a percentage of the maximum speed of each axis set in the Speed Settings Menu (see installation manual). The speed is set in percentage of maximum speed where 100 is maximum and 0.01 is the minimum.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	INDEX	75.00	10.00	0.00	Advance 75 mm from the current position for axis '0' and 10 mm for axis '1'
03	END				End of program.

STOP AXIS

This command allows the user to stop an axis revolving following the use of a Revolve command. This just has 1 parameter which is the axis number to be stopped.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	INDEX	75.00	10.00	0.00	Index advance 75 mm from the current position for axis '0' and 10 for axis '1'
02	MATH	R05 *	2		The value stored in the Register 5 (R05) will be multiplied by 2 and the result will be stored again in R05
03	REVOLVE	1 Solo	85.00		The axis number 1 (data1 column) will revolve continuously at 85.00%. Please note that the user can select between SOLO or RELINK ON MOVE [....] by pressing the ENTER key when the cursor is on one of the data columns of the REVOLVE command.
04	STOP AXIS	0			This command will stop the axis 0 (Data1) revolving.
05	END				End of program.

TORQUE LIM

This command allows the user to limit the output torque for the selected axis (i.e.. current). The command has 2 parameters which are the Axis and "power %" which is the maximum limit in percentage. "Power %" sets the limit for torque.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
02	MATH	R05 *	2		The value stored in the Register 5 (R05) will be multiplied by 2 and the result will be stored again in R05
03	REVOLVE	1 Solo	85.00		Axis 1 (data1 column) will revolve continuously at 85.00%.
04	TORQUE LIM	1	50.00		This command will limit the torque of axis 1 (data1) to 50% of the maximum
05	END				End of program.

Machine Specific:

Air Op Drill

The drill command is used where an axis needs to be sensed in the up position before movement can occur and once in position a 'go' signal is given which then waits for a 'down' position sensor to be triggered before turning off the output and if required turning on another output to reverse the drill action. The second output is sometimes not required as many drill heads have a spring or pneumatic automatic return.

The parameters for the command are quite straight forward, there are 2 columns for the X/Y positions of the hole to be drilled and the third column is for the head number to be used. X/Y Offsets for a maximum of 3 heads can be programmed in the installation menu along with 6 inputs and 6 outputs in the IO installation menu (see installation manual for further details on setup).

No	Command	Data1	Data2	Data3	Explanation
00	Drill	25	25	0	Set X, Y and Head to be used.
01	Drill	50	25	1	Set X, Y and Head to be used.
02	Drill	100	25	2	Set X, Y and Head to be used.
05	END				End of program.

Further MAP commands can be added around and in-between the drill commands to achieve the correct procedure for the machine being used.

The X/Y positions can also be set from registers and these in turn can be set from a downloaded file to speed up the programming cycle. This enables the operator to use one Drill command in a loop to read multiple drilling positions; see Load Parameter File and Parse Parameter File commands for further details.

No	Command	Data1	Data2	Data3	Explanation
00	REG_TYPE	R000 Flpt	R002		Set register type to float to use decimal points.
01	LD_PAR_FILE	RS232			Load the data via serial port.
02	MAIN_SECT	0->1			Start of the main section.
03	PARSE_FILE	R000	R002	R003	
04	DRILL	R000	R001	R002	
05	END				End of program.

COOLANT

Pressing enter while the cursor is over the Coolant command will pop-up a menu with 3 options. These Also relate to the three G code commands M7, 8 and 9 which are as follows:

M7 Coolant Spray

M8 Coolant Flood
M9 Coolant OFF

Select the option required. The outputs used by these commands are setup in the "Assign Outputs" menu in the Technical Setup Manu.

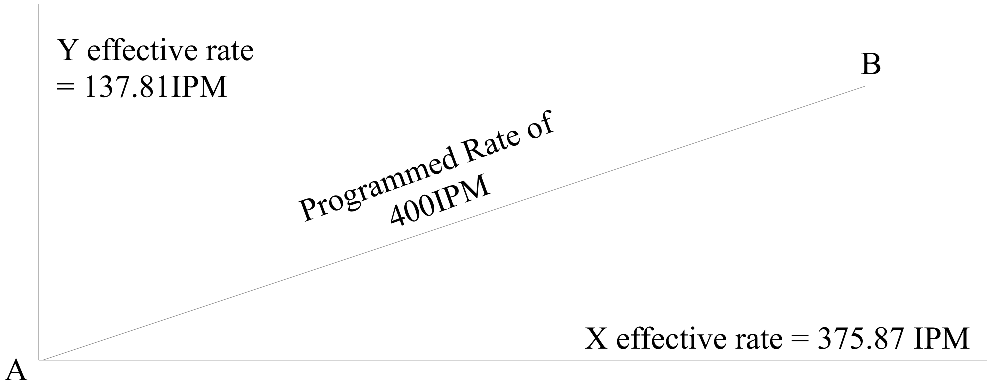
When the program is run initially all the outputs will be in the OFF state. Once a Coolant command is run with M7 or M8 the associated output will be turned on and will stay on until either a M9 is used or the program is terminated. If the program ends or is stopped all outputs will be turned off automatically.

Throughout a program M7 can be followed with M8 and back plus M9 where necessary to suit the cut being performed.

FEEDRATE

This is a single parameter command and affects all axis moved on the following motion commands until reset to another value. Feedrate sets the feed rate in feet per minute or metres per minute according to the data in data[0]. It Automatically adjusts the speed depending on the measurement standard in use.

Data[0] = the feed rate in constant float or register representing travel in mm or inches per minute.



Gcode

This command allows the execution of a G-code program created externally. This is an excellent command for multi-axis CNC machines. The user generates the G-code via a CAD/CAM package or manually with text editor and then by using the USB port, the user loads the file into the controller and executes it.

No	Command	Data1	Data2	Data3	Explanation
00	OUTPUT	0000			Turns off all outputs
01	MOVE	0	0	0	Ensures that all axis are at zero position.
02	INPUT	7	0		Waits for input 7 to read 0 input
03	MAIN_SECT				Start of the main section. Commands above this line will only be run once.
04	G_CODE				When this line is run a file selector box appears and the Gcode file can be selected that is to be run.
05	WAIT	10			Machine waits for 10 seconds
06	JUMP	20	4		Always jumps to line 4 which will cause the file selector box to re-open for the next Gcode program to be selected.
07	END				End of program.

In the above example program we have setup an initialisation section that will ensure the machine is ready for use. In reality this will probably be a longer section as sensors may need to be tested, safety features checked for and other sensors/outputs of the machine setup.

The MAIN_SECT command defines where the operating part of the program starts and in this case we are asking for a Gcode file. Once the program is selected from the pop-up box the Gcode will be checked for consistency and any errors advised.

The user then has the option to setup their origins and start the processing of the Gcode. If selected a live drawing will be shown that informs the operator where the machine is currently up to and what the part should look like.

Once the Gcode has completed the machine will pause at the wait command before jumping back to the Gcode line and asking for a new file.

This provides a simple way of running files continuously with out having to re-run MAP.

Park (and Power Down)

If there is a need for an operator to access the machine to remove or reload parts then as a safety precaution the machine needs to be powered down. This can be done however with the head left in an inconvenient position it would be difficult to reload the machine.

This parameter accepts XYZ positions which is where the controller can move the head to so the machine is safe and the tool or part cannot be damaged. Also as most machines will have custom IO not only for electrical but also pneumatic operations it may not always be desirable to turn off cylinders or extraction facilities.

By placing the cursor over the Park command and pressing enter a pop-up box will appear giving two choices. Park and turn off all Outputs or Park and Preserve outputs.

Please note that the Z axis will retract before any other axis moves to ensure that the tool is clear of the job to avoid damage to either the tool or the part.

PLANES

MAP Planes command is used to the set the arc planes of travel. It uses no data fields but there are 3 condition code options:

Condition code = 0, XY are the set planes Condition code = 1, XZ are the set planes Condition code = 2, YZ are the set planes

The planes command is to be used before the Arc command and by highlighting the command and pressing enter a pop-up box will appear allowing you to select the plane required. This is a persistent command and any following arc will use the planes set by this command. Use another Planes command to reset to normal axis.

SPINDLE

Spindle Has two parameters and also a pop-up menu. Highlight the Spindle command and press enter, the options given are:

M5 Spindle Off
M4 Spindle CCW
M3 Spindle CW
G96 + M3 CSS CW
G96 + M4 CSS CCW

When Spindle is set to M5 Spindle OFF neither parameter has any affect as the output is off.

The M4 and M3 direction commands are setup in the “Assign Outputs” in the Technical Setup Menu. These outputs would be linked to a drive to automatically set rotation direction - useful for tapping applications. A lot of machines may never use reverse rotation and therefore if the drive is hard wired the output type can be ignored and only the speed set.

Data[0] = RPM setting from a constant or a register number

Data[1] = Maximum speed in the CSS mode only from a constant or a register number

Also be aware that using the “Output” command to write directly to the drivers will affect the outputs set by this command.

TOOL

** Please note that this command is to be removed in future releases.

The Tool command has only the one parameter and this is to select the number of the too to be used. The user Creates a tool.map file for this command to use.

Input Command Definitions:

ADC

This command is used to store the digital value of the analogue input in a register(s).

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	INDEX	75.00	10.00	0.00	Index advance 75 mm from the current position for axis '0' and 10 mm for axis '1'
02	MATH	R05 *	2		The value stored in the Register 5 (R05) will be multiplied for 2 and the result will be stored again in R05
03	REVOLVE	1 Solo	85.00		The axis number 1 (data1 column) will revolve continuously at 85.00%.
04	ADC	2	R00	R02	The digital value of the analogue input 2 (Data1) will be stored in R00, R01 and R02
05	END				End of program.

The ADC command uses just one or a range of registers. If a range is to be used, the first time the command is processed the value read will be written to the first register, the next time the ADC is read the register number is indexed and the value written. This way a loop can be used to write to a number of registers and then by using an average command on those registers any noise or drift can be nulled out.

Import

This command is used to receive data into the controller from a computer (a program from TRM has to be installed into the PC first.) or from another controller running compatible software. The data received is stored into the registers for future use. The controller has to be connected to the computer via a serial lead using the RS232 connector.

Before using this command, the user has to set up the speed of communications (Baud Rate) in Technical Menu ---> Set Up Menu ---> Serial Comms Menu

No	Command	Data1	Data2	Data3	Explanation
0	LOAD REG	R31 VorR	5		Register 31 (R31) is loaded with the value 5 (data2). In this program Register 31 is only in use as a flag and when sending the data from the controller be sure to load '0'. The VorR parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears.
1	IMPORT	R00 LDR	R31		Please remember to use the register 31 as a flag only for this example. Load it with a value '0'. Here the controller will import the information from R00 (data1) to R31 (data2). The LDR parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears. An explanation of each options below this table.
2	IF	R31 ==	0	1	The If test is comparing the flag register 31 is equal to '0' and will jump to line No 1 if it is NOT '0'. In other words it is waiting for data from computer. The == parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears.
3	MOVE	R00	R03	R06	Lines 3 to 5 and on are the program you want to run if the register R31 is 0.
4	MOVE	R01	R04	R07	Please remember that the register 31 is used as a flag so do not use it for importing data. Always load R31 with 0 for this example
5	MOVE	R02	R05	R08	
6	END				End of program. The program will start again waiting to receive data from the computer unless it is already received the data while moving.

Options for importing data:

- **Load Data & Clear Stat (LDR):** when the data is sent out to the controller, it is stored in the buffer even if the controller is executing another command. When the data is stored in the buffer the status flag (Stat) is set to '1' which means there is data ready to be loaded. By selecting this option the controller will load the data already in the buffer and after will clear the status flag (Stat). If there is no data ready (stat flag is cleared) the program will continue to the next instruction command without loading any data.

- **Load if Data Present (LDP):** The same as LDR but will not clear the Status flag (Stat).

- **Wait, Load & Clear stat (WLC):** When the import command is executed it will check for the status flag, if it is set, it will load the data straight away. If it is cleared it will wait until receiving the data and then load the data after which it will clear the status flag and continue to the next command.
- **Wait for data & Load (WLD):** It is as the option above but after loading the data it will not clear the status flag (Stat)

Please note that the controller will always be open to receive data into the buffer regardless of the value of the status flag (stat).

Input

This command is intended to pause the program until an input corresponds to the programmed level (high or low). The system can be set-up In the options menu so that while waiting, the command will display a message box to advise the operator. Operation of the program continues after the designated input goes low or high depending on the parameter STATE selected. The minimum low time is 1 ms. The command can be terminated at any time by pressing MENU while the command message box is being displayed.

No	**Command**	**Data1**	**Data2**	**Data3**	**Explanation**
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	INPUT	15	1		The program waits until the controller reads a '1' on the digital input 15 before executing line 3.
03	INDEX	75.00	10.00	0.00	Index advance 75 mm from the current position for axis '0' and 10 for axis '1'
04	END				End of program.

Keypress

Keypress is a simple command and can be inserted in a program to allow the operator time for inspection or as a confirmation all is OK to continue. The program halts on this line and the message **Press Run to Continue** is displayed.

Once any key is pressed the sequence continues.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	KEYPRESS				The program waits until the operator presses a key on the controller.
03	INDEX	75.00	10.00	0.00	Index advance 75 mm from the current position for axis '0' and 10 for axis '1'
04	END				End of program.

QUESTION

This command is used for printing a message on screen (the question) and wait for the user to enter the answer by pressing the number keys and store the value into a register for future use in the program.

This command has 3 parameters:
Reg No.: The user has to select one of the registers available to store the numerical answer.
Msg No.: The user can select which message (the question) to print on screen. The 32 messages are set-up in the settings menu.
Digits: This is the number of digits of the answer, until the operator has keyed in a value the program will not continue.

No	Command	Data1	Data2	Data3	Explanation
00	QUESTION	R00	22	3.00	The message number 22 (data2) will be displayed on screen, this message will ask the user a numeric answer, in this example 3 digits (data3). The 3 keys keyed by the user will be stored in the register 0 (R00, data1). The user can write the question he wants by editing the USER MESSAGES (in this case the message 22) which are found in the settings menu.
01	PRINT_REG	R00	0	0	Will print on screen the data stored in the register 0 (R00) without any decimal points.
02	KEYPRESS				The program will wait in this line until the operator presses any key, (with the exemption of MENU key or STOP key)
04	END				End of program.

READ

This command reads a file created by the WRITE FILE command. The file has the name "outputXX.mdf" where XX is the number of the file. The user has to specify the number of the file and the start and end of the array of registers to read.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	WRITE FILE	output00. mdf	0	15	The 16 registers from register 0 (Data2) to 15 (Data3) will be saved in the file called “output00.mdf” (Data1). This file can be retrieved later on for using the registers.
03	READ FILE	output12. mdf	5	15	The 11 registers from register 5 (Data2) to 15 (Data3) will be loaded with the values of those registers stored in the file called “output12.mdf” (Data1).
04	END				End of program.

Yes/No Box

MAP YES_NO command uses a question box to load a message and requests a yes or no answer from the user. If the user prompts NO to the question the program branches to the line number specified in DATA[1], on yes the program continues on the next line after the YES_NO command

Data[0] = the message number from those setup in the “User Messages” menu

Data[1] = the number of lines to skip on a NO response data

To the right of Data[1] a preview of the message is displayed to aid programming.

Output Command Definitions

ANALOGUE OUT

This command allows the use of one of the analogue outputs when it is not in use for motion control, this command has 2 parameters: "Channel" which is the number of the axis to use and "Op (+/-)%" which is the analogue output voltage in percentage being 100% equal to +/-10 V and 0% equal to 0 volts.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 mm for axis '1'
02	MATH	R05 *	2		The value stored in the Register 5 (R05) will be multiplied for 2 and the result will be stored again in R05
03	REVOLVE	1 Solo	85.00		The axis number 1 (data1 column) will revolve continuously at 85.00%. Please note that the user can select between SOLO or RELINK ON MOVE [....] by pressing the ENTER key when the cursor is on one of the data columns of the REVOLVE command.
04	ANALOG OP	3	-35.00		The analogue output 3 will output -35.00% of 10 volts (e.g. -3.5V).
05	END				End of program.

CHANGE OP

Set a single output. The user has to select the output number and the desired value '0' is in-active '1' active.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the max speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	CHANGE OP	7	1		The program turns ON output '7' (The second column shows '1' which means turn ON the output, In order to turn it OFF the user has to write '0')
03	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
04	END				End of program.

EXPORT

This command is used to send data from the controller to a computer or to another controller. A program from TRM has to be installed into the PC first to receive data. The controller has to be connected to the computer via a serial lead using the RS232 or RS485 interface.

Before using this command, the user has to set up the speed of communications (Baud Rate) in the Technical Menu.

No	Command	Data1	Data2	Data3	Explanation
0	LOAD REG	R31 Apos	1		Register 31 is being loaded with the Actual position (Apos) of the axis 1 (data2). In order to get access to the Apos data, the user has to place the cursor on data1 or data2 of this command and press ENTER and then select the option i.e. Actual Position.
1	EXPORT	R31	R31	0	Exports to the computer the data stored in registers from R31 (data1) to R31 (data2). For this example the only register exported is R31. The third item is the slave number to export to in RS485 communications.
2	INDEX	100.00	200.00	0.00	Index is a relative move and will index 100.00 to the current position of axis 0 and 200.00 to the current position of axis 1
6	END				End of program.

LED

This command gives access to the LED placed in the front of the controller. A binary value can be written in the parameter (being 255 the maximum) or a register that contains the desired value. Please note that the LED's labelled Power, RDY and Key are in use for the controller so you can not fully control those.

To turn the LED's off, enter 0 and to turn a LED or LEDS on enter a number greater than zero. For example to turn on LED 2 enter decimal number 4.

Bit	7	6	5	4	3	2	1	0
Dec.	128	64	32	16	8	4	2	U
Binary	0	0	0	0	0	1	0	0

If all were on the total is 255. Writing out a value of 7 turns on the 3 LED's 0, 1 and 2.

No	Command	Data1	Data2	Data3	Explanation
00	LED	255			Turn ON the LED's.
01	WAIT	0.20			Waits 0.2 seconds
02	LED	0			Turn OFF the LED's
03	WAIT	0.20			Waits 0.2 seconds
04	END				End of program.

MESSAGE

Message is used to display messages in the message centre on the operator screen. The messages can be up to 32 characters long and there are 32 messages to chose from. A message once displayed persists until the next message is displayed or a clear message command is used. When executing the program, the text will be displayed on screen after the Message command is executed.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the max speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	MESSAGE	User Message No 2			The message is displayed on screen. The user can edit the messages in the Technical Menu in the option "User Messages".
03	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
	END				End of program.

Placing the cursor over “Message” in the command column and pressing enter will give the user two options. “Normal Message” and “Erase the message box”. This can be used to easily erase a previously set menu and can be placed anywhere in the code.

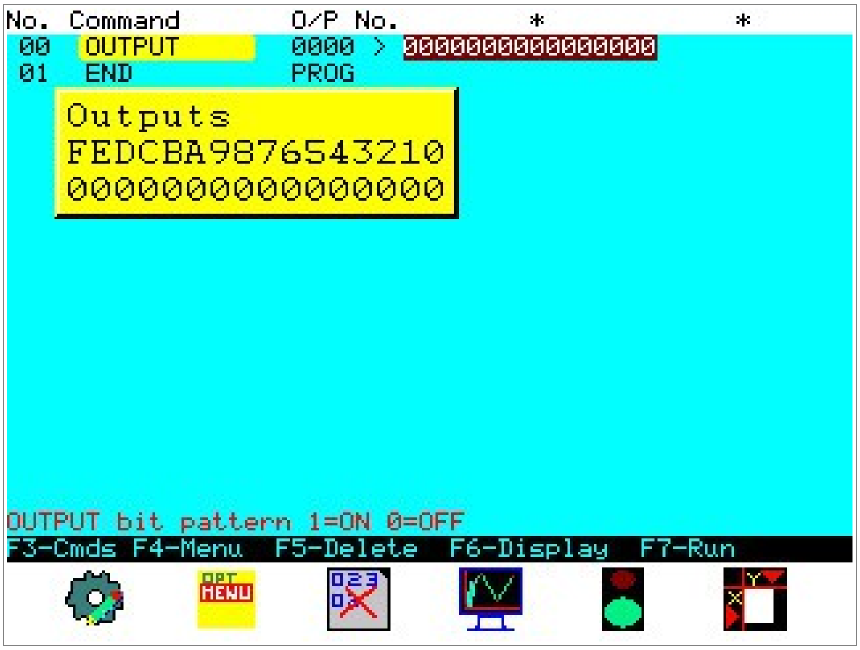
OUTPUT

The OUTPUT command gives the user direct access to the PLC facilities of the controller. The controllers outputs can be individually operated via the output command. Data may be entered in numeric form where a decimal number represents the driver pattern.

To turn the drivers off, enter 0 and to a driver or drivers on enter a number greater than zero. For example to turn on driver 2 enter decimal number 4.

Bit	7	6	5	4	3	2	1	0
Dec.	128	64	32	16	8	4	2	U
Binary	0	0	0	0	0	1	0	0

If the cursor is moved to the command name and the enter key is pressed, a wizard pop up will appear. This allows the user to move along to the outputs required and set to on or off. When enter is pressed again the correct decimal value is entered into the Value column and a representation of the outputs set appears next to the value.



```
No. Command         O/P No.          *            *
 00  OUTPUT         0000 > 0000000000000000
 01  END            PROG
      Outputs
      FEDCBA9876543210
      0000000000000000

OUTPUT bit pattern 1=ON 0=OFF
F3-Cmds F4-Menu  F5-Delete  F6-Display  F7-Run
```

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	OUTPUT	4	1		The program turns ON output '2' (4 is the representation of a binary number in the decimal system, in this case the output 2 is the only output with a '1')
03	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
04	END				End of program.

PRINT REGISTER

Print Register prints the value of a selected register on screen as the process is running. The screen position the data appears at is controlled via the % X and Y columns.

If desired, further register prints can be made in the program at a different location to avoid overwriting the previous data. Use the same location to update the data already present.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R25 VorR	547		This loads register 25 with the value 547. The VorR parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears.
01	PRINT REG	R25	50	25	Print on screen the value of register 25 (i.e.. 547) Start printing at a position 50% of the screen width and 25% down.
04	END				End of program.

WRITE

This command stores the registers and its contents in a file for future use. The user has to specify the start and end of the array of registers to be stored in the file. The file has the name "outputXX.mdf" where XX is the number of the file.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the max. speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	WRITE FILE	output00. mdf	0	15	The 16 registers from register 0 (Data2) to 15 (Data3) will be saved in the file called “output00.mdf” (Data1). This file can be retrieved later on for setting the registers.
03	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
04	END				End of program.

Program Flow Command Definitions

BEGIN – END FUNCTION

This command is for creating a sub-program that can be called at any time. This can be used when the same routine is executed in different parts of the program and saves the user writing the same code multiple times. These sub-programs will normally be located just before the END of the program. At the end of the sub-program it is required to use a END FUNCTION command which will indicate to what line the program will return after executing the sub-program. It is possible to have 99 sub-programs in every MAP program.

No	Command	Data1	Data2	Data3	Explanation
00	LED	255			Turn ON the LED's.
01	CALL	0			Calls the sub-program number '0'. After the execution of the sub-program MAP will execute the next line. Sub-programs are located at the end of the MAP program.
02	LED	0			Turn OFF the LED's
03	CALL	0			Calls the sub-program number '0'. this is the second time the sub-program '0' is called in this program. After the execution of the sub-program MAP will execute the next line. Sub-programs are located at the end of the MAP program.
04	WAIT	0.20			Waits 0.2 seconds
05	BEGIN	0			Begins the sub-program '0'. All lines below a BEGIN belong to sub-programs, therefore will only be executed when a CALL command calls the sub-program.
06	MOVE	100.00	100.00	100.00	All 3 axis are moving to position 100.00
07	WAIT	0.20			Waits 0.2 seconds
08	MOVE	100.00	100.00	100.00	All 3 axis are moving to position 0.00
09	END	Function			End of sub-program '0'. After this instruction the program will return to the next line after the sub-program was called. In order to use this command select the END command and then move the cursor to the Data1 parameter press ENTER and select the "END return" option.
10	END				End of program.

BRANCH_NE

Branch not true, this command differs from the IF command because it does not jump to a fixed line. If the condition is false it only skips the number of lines that are programmed. This allows lines to be added or deleted from the program and the command will always skip the same amount of lines.

No		Command	Data1	Data2	Data3	Explanation
00		ADC	0	R000	R000	Get a reading from Analogue channel 0
01		BRANCH_NE	R000 <=	50	2	If R000 is less than or equal to 50 carry on to line 2, otherwise the condition is false and 2 lines will be skipped, Line 4 will then be the next command.
02		CHANGE OP	1	1		Output 1 will be turned on
03		MOVE	50	100	0	Move axis 0 and 1 to 50 and 100
04		MOVE	0	0	0	Move all axis to 0.
05		CHANGE OP	1	0		Output 1 will be turned off.
10		END				End of program.

In the above example the register R000 will be loaded with a reading from the ADC channel 0, the compare command will then test to see if it less than or equal to 50. If it is, Output 1 will be turned on and the axes 0 and 1 will move to 50 and 100 respectively. This will then be followed by line 4 and 5 and then the cycle will start again.

If the compare condition is false the lines 2 and 3 will be skipped and the axis kept in position at 0,0, the output will also be kept turned off. The cycle then repeats.

The Compare command has 4 parameters:

- The register number being used
- The type of test (press enter for options)
- The value or register being compared to
- The number of lines to skip

The tests available are:

1	Equal	==
2	Not Equal	!=
3	Greater than	>
4	Greater or equal	>=
5	Less than	<
6	Less or equal	<=

A red side bar will be drawn to indicate where the command will skip to, this helps avoid errors when programming.

CALL MAP Program

This command calls another saved MAP program that you have previously created, you can have as many sub-programs as needed depending on the controllers memory capacity. After the sub-program is executed, the program will return to the following line after the CALL command and resume the original program.

Press enter when the command is highlighted and a file selection box will appear. Use the + and – keys to change directory and the up and down arrow keys to choose the file to be used. Please remember that if you pick a file on C: the pen drive will always need to be plugged in before the program is run otherwise unexpected results or possible damage to the machine will occur. Once a file has been tested we suggest that it be moved to the A: (battery backed) or B: (FLASH) drives for permanent storage.

Each Sub-program can have it's own

Having a sub-program is very beneficial when writing complex process's as they can be broken down into small routines and called as needed. This also saves re-writing the routing if it is to be used many times. It can also be possible to test each routine on its own without running the full program which simplifies troubleshooting and prevents material wastage.

CALL SUBROUTINE

This command has to be used together with the BEGIN command and the END FUNCTION command. The user can create up to 99 sub-programs, It is not permitted to jump in or out of the sub program.

Subroutines would normally be written at the end of the program, to aid clarity and with sequential numbers to avoid confusion. Numbers assigned can be 1,2,3,4 etc or 10, 20, 30 as the programmer desires.

Enter a Begin command followed by any other commands as needed to create the subroutine. Once complete place the cursor over the end command and press enter. Select the “Function End” option and press enter. This will insert a new End command and move the Program End command down.

No	Command	Data1	Data2	Data3	Explanation
00	LED	255			Turn ON the LED's.
01	CALL	0			Calls the sub-program number '0'. After the execution of the sub-program the MAP will execute the next line.
02	LED	0			Turn OFF the LED's
03	CALL	0			Calls the sub-program number '0'. this is the second time the sub-program '0' is called in this program. After the execution of the sub-program the MAP will execute the next line.
04	WAIT	0.20			Waits 0.2 seconds
05	BEGIN	0			Begins the sub-program '0'. All lines below a BEGIN belong to sub-programs, therefore will only be executed when a CALL command calls the sub-program.
06	MOVE	100.00	100.00	100.00	All 3 axis are moving to position 100.00
07	WAIT	0.20			Waits 0.2 seconds
08	MOVE	100.00	100.00	100.00	All 3 axis are moving to position 0.00
09	END	Function			End of sub-program '0'. After this instruction the program will return to the next line after the sub-program was called. In order to use this command select the END command and then move the cursor to the Data1 parameter press ENTER and select the "END return" option.
10	END				End of program.

If a subroutine is created in between other code, the end command can be selected from the commands menu and inserted this way. A Program End will be inserted and all code below will be hidden. Highlight the end command, press Enter and change to Function End, all code below will now re-appear.

DEC JUMP

Decrement Jump is similar to the 'Jump' instruction but this time the program will decrement a variable loaded with any number, when the variable is not zero the program will jump to the instruction line indicated, when it is zero it will follow with the next line in the program.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
02	DEC JUMP	R05	1		The program will decrement in 1 the value loaded in the Register 5 (R05) and will compare the result with 0, if it is different to 0 the execution of the program will jump to line number 1. When the comparison is equal to 0 the program will continue with line number 3.
03	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
04	MOVE	50.00	70.00	0.00	Move axis '0' to 100 and axis '1' to 200
05	END				End of program.

DO

This command has to be used together with the WHILE command. The program will execute the lines located between DO and WHILE as long as the condition set in the while command is met. As the test is after the commands written in between the Do..While, the commands will always be executed once on each loop of the program.

Please see the example with the WHILE command and also see Do_Move command if a position needs to be included.

END

The End command is used to tell the system that the end of the programme has been reached and is the only command that cannot be deleted from the end of a program. When a new program is started an End command will be automatically inserted.

You may use the End Command to break up a programme which you are trying to debug. This can be done by inserting the End Command at the appropriate points in the programme and all code below will be hidden and therefore not executed. Delete the End command to restore the hidden lines.

If the Command is highlighted with the cursor and the Enter key is pressed, an option box appears to allow a Function End option to be selected. Function end would normally be used to follow a “Begin” command. With other commands in-between, this creates a function or “subroutine” that can be called from the main program.

EXIT

MAP EXIT causes map to exit from the current process. If the process is a child then map returns to the calling process whereas if the process is the main process it stops execution of MAP. There are no data fields associated with this command.

IF

This command compares a selected register to another register or value and jump to a different line in the program if the comparison is false. This command has 4 parameters:

1. **Reg No.:** The user has to select the register that is going to be compared against a value or a value held in a different register.
2. **Comparison Type:** In order to access them move the cursor to one of the other parameters and press ENTER, a window appears with the tests available.
3. **Compare Value**: The user can compare the register selected in No.1 against a value or a different register. In order to compare it against a value, enter the value using the keypad or select a register by pressing the '+' key and enter the number of the register required.
4. **Line No.:** This is the line the program will jump if the condition is met.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
02	TIMER	R31 R-1	8	1000	The register 31 (data1) will be used as a down counter timer (R-1). the timer will use the channel number 8 (data2) and the starting value will be 1000 (data3) multiplied by 10mS (e.g.. 10S)
03	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
04	IF	R31 ==	0	4	Here we are verifying if the timer used in line 2 (R31) has reached 0 (data2) or not. If not, the program will jump to line 4 which is the same line so the program will remain in this line until the timer has reached 0. Once the timer has reach 0, the program will continue to line 5. Please note that the timer by itself does not stop the program! if the user only wants to stop the program for a little while then the WAIT command is available.
05	ANALOGUE OP	3	-35.00		The analogue output 3 will output -35.00% of 10 volts (e.g. -3.5V).
06	END				End of program.

The IF command has 6 comparison options, these options are visible by positioning the cursor on the Data1 or Data2 column on this command and pressing the ENTER key. A pop up window will appear on screen with the following options:

1	Equal	==
2	Not Equal	!=
3	Greater than	>
4	Greater or equal	>=
5	Less than	<
6	Less or equal	<=

The user has to select one of the options, by default the == (Equal) will be displayed. when selecting the IF command.

JUMP

Jump is similar to an 'IF' statement in basic, Pascal or 'C' programming languages in that if an input bit is active and the bit number is within range, the processor will jump to the line listed in the data parameter 2. For example in an application input 3 is connected to a hopper sensor such that when the hopper is below a minimum level the signal becomes inactive (high). When the signal is inactive a message is displayed and the process paused. The following code illustrates this:

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	JUMP	3	4		If port bit 3 is active (low) jump to line 6
03	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
04	MOVE	50.00	70.00	0.00	Move axis '0' to 100 and axis '1' to 200
	END				End of program.

Main_Section

The Main_Section command defines the start of the 'Running' program. All code prior to this line can be regarded as 'set-up' code. Main_section has two options, press enter to access them. They are 'Run from start' and 'Run if not homed'.

Run From Start: Every time the program is run the initialisation code (all code before the Main Section) will be run as well.

Run if Not Homed: The initialisation code will be run only if the system has not homed. Once homed, pressing start will only run the code following the Main Section command.

REPEAT_IF_TRUE

REPEAT IF command which is similar to the MAP WHILE command except that it can not be nested and it uses branch which are relative jumps. The command performs a logic compare and if TRUE will run the lines below the command when the last line is run eg the last one before the branch the command will branch back to the compare and the cycle will continue until another instruction or the equation is

FALSE. A test False causes a branch outside

Data[0] = the base resister and is a constant register number

Data[1] = the compare value which is a constant or register number

Data[2] = the number of lines to skip

Condition code gives the test options 0 = equal EQ: 1 = Not Equal NEQ: 2 = Greater than GT: 3 = Greater than equal GTE: 4 = Less Than LT: 5 = Less than equal LTE:

WHILE

The WHILE command is used to make comparisons between values stored in registers or a value of a register and a value. WHILE can be used in conjunction with the DO command or on its own. The command has 3 parameters:

1. **Comparison:** In order to select a comparison, move the cursor to highlight the command and press enter to view the pop-up box. There are 6 options to choose from.

1	Equal	==
2	Not Equal	!=
3	Greater than	>
4	Greater or equal	>=
5	Less than	<
6	Less or equal	<=

   The user has to select one of the options, by default the == (Equal) will be displayed.

2. **Reg No.:** The user has to select the register to be compared.
3. **Compare:** The user can compare the register selected against a value or a different register. In order to compare it with a value just enter the value using the keypad or select a register by pressing the '+' key and enter the number of the desired register.

See following table:

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	DO				The DO while loop starts here. all commands between this loop will execute at least the first time. If the condition in the WHILE command is true, the loop will repeat again.
02	INDEX	75.00	10.00	0.00	Index advance 75 mm from the current position for axis '0' and 10 mm for axis '1'
03	LOAD REG	R05 IP.b	3		Loading the value of the input 3 (data2) into the register R05. The IP.b parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up yellow window appears with 10 options.
04	WHILE	R05 ! =	0		This is the end of the DO-WHILE command.
05	OUTPUT	5			Turn ON the Outputs 0 and 2 for its representation in binary. The user can use a wizard by pressing the ENTER key when the cursor is in the parameter (5 in this example)
06	WAIT	10.00			The program will wait 10 seconds in this line.
07	OUTPUT	0			Turn OFF all the OUTPUTS
08	END				End of program.

In the above example the loop will repeat as long as the condition stated here is met. If the value stored in register R05 is not 0 the execution of the program will jump to the DO command and execute all the lines in between.

Maths Command Definitions

AVG

This command is for averaging the value of the specified registers and the result will be stored in another specified register.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10.00		The Register 5 (R05) is being loaded with the value 10
01	LOAD REG	R06 VorR	20.00		The Register 6 (R06) is being loaded with the value 20
02	LOAD REG	R07 VorR	30		The Register 7 (R07) is being loaded with the value 30
03	AVG	R05	R07	R00	This command is averaging the values of the registers R05 to R07 (specified in start register (Data1) and end register (Data2)) and storing the result in R00 (Data3)
04	END				End of program.

Cosine

This takes the cosine of an angle entered in Radians or in Degrees and returns it in the Register specified. Place the cursor over the Cosine command and press enter to show a selection box for Radians or Angle.

Find Maximum

To find the maximum value contained in a series of registers enter the start register and the finish register in a range. Also enter the result register to be used. Once the command has run the **number of the register** containing the maximum value will be placed in the destination register.

Find Minimum

To find the minimum value contained in a series of registers enter the start register and the finish register in a range. Also enter the result register to be used. Once the command has run the **number of the register** containing the minimum value will be placed in the destination register.

Logic Operators

These are Binary (to the power of) Shift Left, Shift Right, OR, AND, XOR and NOT

Shift Left and Shift Right
Shifts the value already in the register by the number explicitly specified or in another register and stares the result in the initial register.
OR, AND, XOR, NOT
Does a binary conversion on the two values given and returned into the register specified.

For example the AND operation on 8 and 3 would result in 0

	128	64	32	16	8	4	2	U
8=	0	0	0	0	1	0	0	0
3=	0	0	0	0	0	0	1	1
Result=	0	0	0	0	0	0	0	0

Whereas 8 and 14 = 8

	128	64	32	16	8	4	2	U
8=	0	0	0	0	1	0	0	0
3=	0	0	0	0	1	1	1	0
Result=	0	0	0	0	1	0	0	0

MATH

This command is to make mathematical operation between the data stored into the specified registers.

The user can select the Mathematical operation by positioning the cursor on the MATH command line and in the column of one of the registers and press ENTER. A pop down window will be displayed with all the mathematical and logical functions available.

Mathematical operations available are:
Add +
Subtract -
Multiply *
Divide /
Not !

The “Not” operation takes a value and re-writes it as a negative number. i.e. 100 becomes -100. All the user needs to do is specify the register to use to perform this operation on.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10. The VorR (Value or Register) parameter is accessed by placing the cursor on top of the data1 or data2 of this command and pressing ENTER. A pop up window appears with options. In this example we choose "Value or Register" (VorR) which indicates that the data2 has to be a value or a register. Please note that the data2 is attached to the option selected.
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
02	MATH	R05 *	2		The value stored in the Register 5 (R05) will be multiplied by 2 and the result will be stored again in R05. The * parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears with options.
03	MATH	R05 /	R07		The value stored in the Register 5 (R05) will be divided with the value stored in Register 7 (R07) and the result will be stored again in R05. In order to select the R07 register the user has to position the cursor in the point and press the '+' key
04	MOVE	50.00	70.00	0.00	Move axis 'to positions shown.
05	END				End of program.

Sine

This takes the sine of an angle entered in Radians or in Degrees and returns it in the Register specified. Place the cursor over the Cosine command and press enter to show a selection box for Radians or Angle.

Square Root

This takes the square root of a value (value or register) and returns it in the Register specified.

Sum

To find the sum of values contained in a series of registers enter the start register and the finish register in a range. Also enter the result register to be used. Once the command has run the sum will be placed in the destination register.

Tangent

This takes the tangent of an angle entered in Radians or in Degrees and returns it in the Register specified. Place the cursor over the Tangent command and press enter to show a selection box for Radians or Angle.

Timer Command Definitions

Halt

Stop the Timer. This command has just 1 parameter and that is the register assigned to the timer that you want to stop.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
02	TIMER	R31 R-1	8	1000	The register 31 (R31) (data1) will be used as a down counter timer (R-1). The timer will use the channel number 8 (data2) and the starting value will be 1000 (data3) multiplied by 10mS (e.g.. 10S). The R-1 parameter is accessed by placing the cursor on data1 or data2 of this command and pressing ENTER. A pop up window appears with 2 options. R-1 and R+1
03	INPUT	5	0		Index advance 75 from the current position for axis '0' and 10 for axis '1'
04	HALT TIMER	R31			Stops the timer running on R31
05	IF	R31 !=	0	2	Here we are verifying if the value of the timer used in line 2 (R31) is different (!=) to 0 (data2) or not. If not, the program will jump to line 2 (data3). Once a timer has reach 0 it will stop running! Please note that the timer by itself does not stop the program! if the user only wants to stop the program for a little while then the WAIT command is available.
05	ANALOGUE OP	3	-35.00		The analogue output 3 will output -35.00% of 10 volts (e.g. -3.5V).
07	END				End of program.

Timer

This command gives access to the timers, there are 8 timers available. The command has 4 parameters:

- **Reg No:** The user has to set 1 of the registers available as a counter. The user can then manipulate the counter as required by utilising any other command that can write to the register.
- **R-1 or R+1:** The user has to choose if the counter is a down counter or up counter. In order to select this parameter the user has to move the cursor to one of the Data* columns and by pressing ENTER the 2 options will be shown in a pop-up box. Select the option required and press Enter.

- **Channel:** Select 1 of the 8 timers available (from 1 to 8)
- **Initial value:** By pressing the + key in the Data3 column, you can opt to use another register to set the timer to. Otherwise just enter the value required (whole number, no decimals).

Please see an example of the TIMER command together with HALT TIMER command.

Wait

The wait Command allows the controller to wait for a specific period of time. The minimum time is 10 ms. The time delay is entered in seconds. The command will allow time entry to 0.01 seconds by using the decimal point '.' When the Wait Command is executed, a message can be displayed (depending on setup options) to the operator informing them that the controller is timing. After the given amount of time the controller will commence execution of the next instruction. The command can be terminated at any time by pressing MENU while the command message box is being displayed.

The Wait message box can be enabled or disabled in the Options Set-up Menu. If set to disabled the controller waits for the programmed time but the screen remains blank.

No	Command	Data1	Data2	Data3	Explanation
00	SPEED	80.00	100.00	0.00	Sets the speed of axis '0' to 80% and axis '1' to 100% of the maximum speed.
01	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
02	WAIT	10.00			The program waits 10 seconds before executing line 3
03	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
04	END				End of program.

If a register is used to set the wait time there are various options. An integer register will set the time in whole seconds only therefore if decimals are needed the register type must be set to a floating point type. The minimum time period is still 10mS.

Copyback (Copy Child Registers)

This command must be used in each child process if the entire register range is not to be copied to the parent on return to the parent process.By using the Map CopyBack command you can set the values of the start and end registers to copy back to the parent process. If used and set to Reg0..Reg0 as a range just the one register will be copied back.

Data[0] = the start register number to copy back note this must be the lowest value
Data[1] = the end register number to copy back note this must be the highest value

Example if Data[0] =100 and Data[1] = 200 registers from 100 to 200 will be copied back to the parent process.

Load Register

Load register is used for loading a value into one of the 256 registers with a value that can be from a keypress, inputs, ADC reading, a pulse count and more. The user has to specify what register to load from 0 to 255 and what value (parameter) to load or 'copy' into the register. The full list of parameters that can be loaded in the registers are found by moving the cursor to the command name and pressing ENTER.

1	Value or Register VorR	By using this option it is possible to load into the register a value or the value stored in a different register.
2	Actual position Apos	The user selects a register and loads it with the actual position of any of the axes, the user has to specify which axis.
3	Encoder Register Encr	The user can select any of the 2(SMC) or 3(PMC) on board encoder registers and load its value into a selected register.
4	Following Error Ferr	The following error is loaded into the register selected by the user.
5	Torque Output Torq	The torque output value is loaded into the register.
6	Get Input Port Port	The decimal value of the 16 digital inputs is loaded into a selected register.
7	Input Port Bit IP.b	The user selects what input (bit) to load into the register. As it is only one bit the value loaded will always be 1 or 0.
8	Get keypad Input Key	Load the value of the key pressed into a selected register.
9	Get Jog Encoder Position Jpos	This command loads the position of the Jog encoder into the programmed register.
10	Tool Number T->R	Load the current tool number into the destination register. Data[1] is unused.
11	Load Register Indirect (R)R	The contents of a register pointed to by the value contained in another register are loaded into the destination register.

No	Command	Data1	Data2	Data3	Explanation
00	LOAD REG	R05 VorR	10		The Register 5 (R05) is being loaded with the value 10. The VorR (Value or Register) parameter is accessed by placing the cursor on top of the data1 or data2 of this command and pressing ENTER. A pop up window appears with 10 options. In this example we choose "Value or Register" (VorR) which indicates that the data2 has to be a value or a register. Please note that the meaning of data2 depends of the option selected.
01	INDEX	75.00	10.00	0.00	Index advance 75 from the current position for axis '0' and 10 for axis '1'
02	DEC JUMP	R05	1		The program will decrement in 1 the value loaded in the Register 5 (R05) and will compare the result with 0, if it is different to 0 the execution of the program will jump to line number 1. When the comparison is equal to 0 the program will continue with line number 3.
03	MOVE	100.00	200.00	0.00	Move axis '0' to 100 and axis '1' to 200
04	MOVE	50.00	70.00	0.00	Move axis '0' to 100 and axis '1' to 200
05	END				End of program.

Load Counter

Allows the user to load one of two counters, 0 and 1, with a value. This relates to the Load Register command when it is set to load from 'Get Jog Encoder Position'

Map LD_COUNTER command is used to load a value into the jog wheel registers where the value loaded is long signed data for SMC there are two channels 0 to 1 and for the PMC4 there are 3 channels 0 to 2

Data[0] = the constant long data for the channel

Data[1] = The data to load into the jog encoder register

No	Command	Data1	Data2	Data3	Explanation
00					
01	DO				
02	LOAD REG	R000 Jpos	0		Loads the Register 0 with the 'Jog Encoder' Position.
	IF				
	LD_COUNTER	0 Jpos	1000		Loads the counter 0 with 1000
03	WHILE	R000 <=	1500		
04					
05	END				End of program.

Load File Data

New command*

Load Multiple

Load Multiple registers in one command. In Data1 set the start register, in Data2 set the end register and in Data3 set the value to load into those registers. Bear in mind certain values can only be loaded in to specific register types so ensure the type has been set first.

In Data3 you can also use the + key to set the value from another register.

Load Parameter File

This command is used to load data via the serial communications port. The command does not accept any other information and simply waits for data to be received on the RS232 port. A message is displayed to the operator confirming that the data can now be sent, selecting send on the PC software will then download all of the data to the controller.

This data is stored on the controller in a system file which is not accessible to the operator. The Parse File command looks for the assigned file and reads the data into registers, see Parse File for further details.

Load System Vars

To load certain system variables this command can be used and the value to be modified is chosen by placing the cursor over the command and pressing enter. A list of values is then selectable from a pop up window. Options are:

- Calibration Length
- Tool Height
- Tool Diameter
- Tool Changer X
- Tool Changer Y
- Tool Changer Z
- Velocity
- Data Logger Counter

The Calibration Length is the length that has been set in the Calibration menu as the maximum size for an axis. If Calibration Length is selected there are 2 parameters available. The first is the Register to put the value into and the second is the number of the axis the data is to be used from. The tool parameters are setup in the Technical menu (see installation manual).

If Tool Height is selected there are 2 parameters available. The first is the Register to put the value into and the second is the number of the tool the data is to be used from. The tool parameters are setup in the Technical menu (see installation manual).

If tool Diameter is selected there are 2 parameters available. The first is the Register to put the value into and the second is the number of the tool the data is to be used from. The tool parameters are setup in the Technical menu (see installation manual).

If “Tool Changer X,Y or Z” is selected there are 2 parameters available. The first is the Register to put the value into and the second is the number of the tool the data is to be used from. The tool changer is setup in the Technical menu (see installation manual).

The Velocity parameter is the maximum velocity set in the Speed Settings menu (see installation manual). There are 2 parameters available. The first is the Register to put the value into and the second is the number of the axis the data is to be used from.

Data logger all samples are on same time period and this records the number of samples taken.

Parse File

Adding the Parse File command to a program allows the system to extract co-ordinates and other information sent to the controller. Various commands in MAP can then use the information stored in the registers to move to position, select a tool, cut or drill before moving onto the next position. A third register is used to store any leftover data that may have been received but not supported.

Normally another command such as “While” would be used to check to see if any more data was present. Testing for a zero on the third register signifies that the end of file has been reached and there is no further data to process. The program can then be stopped or continue with the rest of the program. If the parsing of the file is stopped part way through a “Rewind File Pointer” command can be used to restart the parsing of the file.

REG_TYPE

This command is to set the type of register. There are 5 options to choose the register type. They are as follows:

1	SLONG	Signed long is an integer number that can be positive or negative
2	ULONG	Unsigned long is an integer number without sign
3	FLOAT	Float is a register that can store decimal places
5	BYTE	Unsigned Byte is a type of register that can only store maximum of a byte. This means that the maximum number it can store is 255. If you try to store a bigger number the controller will 'AND' the byte with 255 and will store the result.

Place the cursor in the Data1 or Data2 column and type in the register number to start from. You can either enter the number directly or press the + key for a prompt box. Pressing the Enter key in either column will bring up the pop-up box of options available as in the table above. Scroll to the required type and press enter.

No	Command	Data1	Data2	Data3	Explanation
0	REG_TYPE	R000 Char	R005		This line will set all 6 registers from 0 to 5 to type Char
1	REG_TYPE	R006 Float	R020		This line will set all 15 registers from 6 to 20 to type Float for use with decimal places.
…	…	…			
10	END				End of program.

Rewind File Pointer

If a file has been partially completed (parsed) and for example a tool breaks, the position can be re-wound to the beginning and the whole job run again. Keys can be set up as yes/no by using registers to capture a key to and comparing the value captured with that expected.

Canvas Drawing Command Definitions

Draw Canvas

This allows the user to redraw the Canvas (run screen) following a “Clear Canvas” command.

Clear Canvas

Clears the running screen of all data. For example it may be necessary to setup a machine by printing many registers to the screen. Once complete this data is no longer required and can be removed from the screen. Normally following a “Clear Canvas” command a “Draw Canvas” command is used to restore the screen data.

Canvas Draw Pointer

Sets the position on screen in percent as to where the data or object will be drawn.

Filled Circle

Draws a circle or quadrants of and fills with a block colour. Colours can be set via the “set colours” command prior to executing the Filled Circle command. Quadrants are 1, 2, 4 or 8 and combining those values will draw just those Quadrants. If 15 is used then a full circle is drawn.

Draw Rectangle

Once a position has been set using “Canvas Draw Pointer” a rectangle size can be set, again using %.

Set Colours

The Foreground and Background colours can be set for items being drawn. Enter the value of each colour required in Hexadecimal. There are 5 bits of Blue, 6 Green and 5 Red.

For example entering FFFF will give a peak white colour whilst 0000 will produce Black. Turning on each bit of a colour increases it's brightness and by mixing the brightness of each of the three primary colours the user can program any of 65535 colours.

Peak RED = F800, Peak GREEN = 07E0, Peak Blue = 001F

Draw Line

There are four parameters to the Draw Line command. Enter the X and Y start position in % of where the line is to start in the first two data columns. Next set the X and Y finish points in the last two data columns.

Use the Set Colours Command before the Draw Line command to set the colour of the line.

No	Command	Data1	Data2	Data3	Data4	Explanation
0	Draw Line	50	0	0	100	Starts drawing at the top centre of the screen and finishes at the bottom centre (Vertical Line is produced)
1	Draw Line	0	50	100	0	Starts drawing at the middle left of the screen and finishes at the right middle (Horizontal Line is produced)
…	…	…				
10	END					End of program.

In the above example, a crosshair pair of lines is produced centrally on the screen.

Draw JPEG Image

The Jpeg Image command draws an image on the canvas starting at draw point X,Y set with the DRAW POS command. The data fields are joined together to form the stored file name on the SMC this should be no more than 12 characters on the SMC and on the PMC4 16.

JPEG Image Size

MAP command Jpeg Size sets the size of a Jpeg image in the X and Y planes in percentage of the canvas area

Data[0] = X plane size max value 100%
Data[1] = Y plane size max value 100%

This command should be used before the Draw Jpeg command so that the size is set when drawn.

Work Piece Size

Specifies the dimensions of the work piece to be drawn in the live tool path image

There are 3 dimensions for the workpiece, they are X, Y and Z which can be set directly or via a register. Press the + key to enter the register number.

Data Recording

Log Task

The data logger can record ADC inputs, DAC outputs and other features of the controller that may need to be recorded. This gives the user great flexibility when analysing data as diagnostics may show that when a certain combination of outputs and inputs are active there is an adverse affect in the analogue readings or it may be used as verification that once a setpoint is reached an operation actually occurs.

Options are:

1. ADC Channel
2. Position Reg
3. Elapsed Time
4. Torque Demand
5. Register
6. Input Port

Explanation:

1. If the ADC command is selected there are 2 available data options; the channel to be recorded and the number of samples to average during a read of the ADC.
2. The position register option has one parameter which is the encoder number, the second option is ignored.
3. Elapsed Time has no options to be set
4. Torque Demand option has one parameter which is the axis to be recorded.
5. Register option has one parameter which is the register number to be recorded.
6. Input Port has no options and records the whole port.

This command should be followed by the Start Log command which is where the number of samples and time period are set.

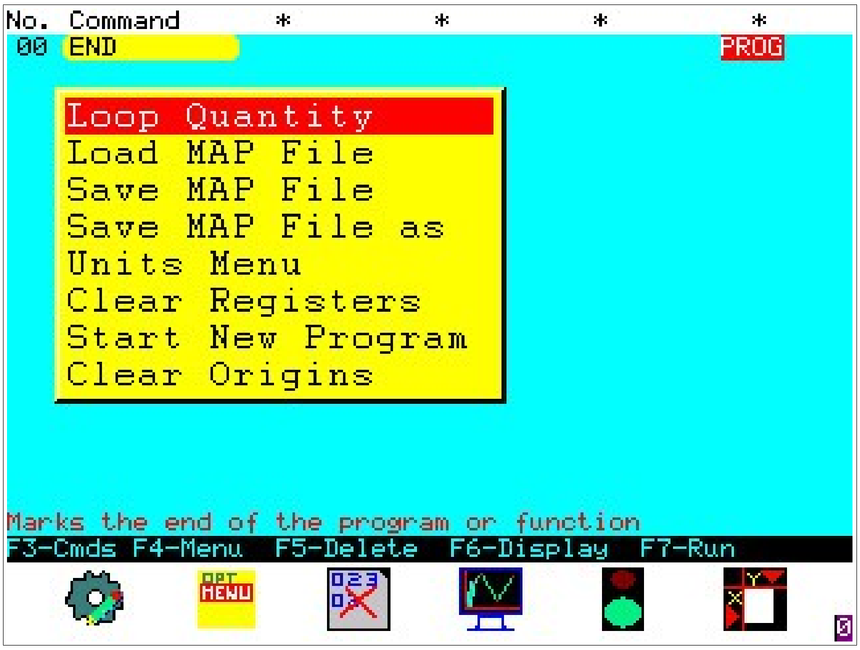
Start Data Recording

Starts Recording data based on two parameters – Maximum number of samples to take and the time period between samples. Time period is in

Stop Data Recording

Stops Recording data, there are no further parameters for this command.

Other Menus



Marks the end of the program or function

Options Menu OPT MENU

By pressing the key under the menu icon, a pull down menu appears showing 4 options as shown above. These options are:

Loop quantity

The controller has a loop counter, which stops the controller when the counter reaches zero. If the counter is initially set to zero then the machine will operate until the operator stops the controller by pressing a key on the Keypad. The operator should programme the loop counter with the number of repetitions they wish the system to complete. Once the loop counter reaches '0' the program will stop and the counter will be set to zero again.

Load, Save, Save As MAP file

The user can store and recall from Memory as many files as needed, the only limit being the size of the memory. For the available internal memory please see the datasheet for the particular controller you are using.

The user can also save programs into a USB drive if extra space is required or for back up. When the pop up window appears the user first has to select where to save the file by using the + and - keys to select a drive: A:= RAM Memory, B:= Flash Memory, C:= USB drive and then type the required file name.

Units Menu

By selecting Units menu a new pull down menu appears from which the user can select metric or imperial units. This is a stored setting so that after the power has been turned off the previously set units are used.

Clear Registers

The user has 256 registers available in a program in which he can store data, do mathematical operations, logic operations etc. By selecting this option the data stored into the registers will be deleted. The system can also be setup in the settings menu's to clear the registers on start up if required.

Start New Program

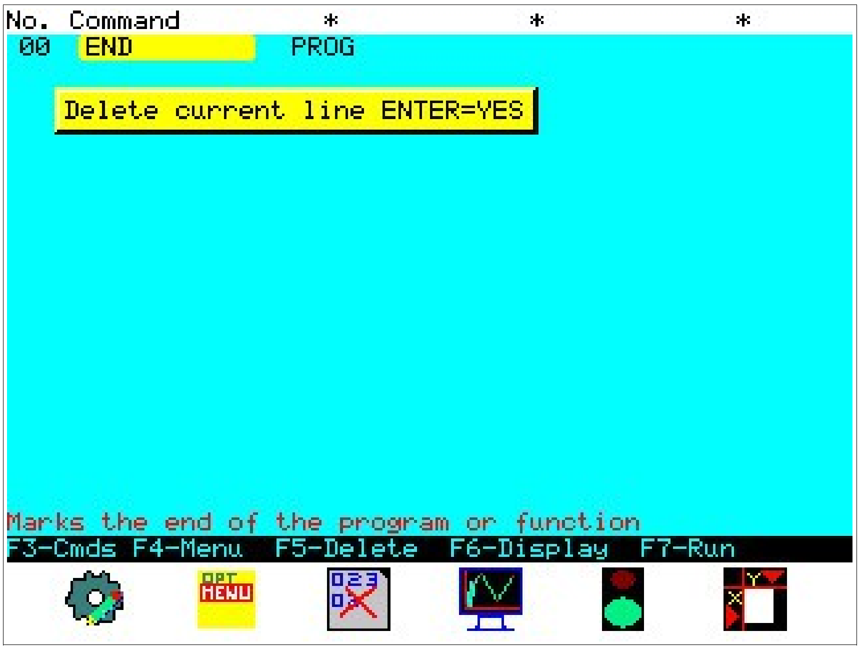
This option is used when the user wants to create a new program. By selecting this option the user will be prompted that any unsaved data in the current program will be lost.

Clear Origins

The Origins can be cleared with this option if set and for instance a new job is to be started.

Deleting a command

To delete a line move the cursor over the command that requires deleting. Pressing F5 a message box will appear with the caption "Delete current line ENTER = Yes". As stated press ENTER to delete the line and any other key to cancel the delete command. After deleting the current line, all the lines following will be moved up one position.

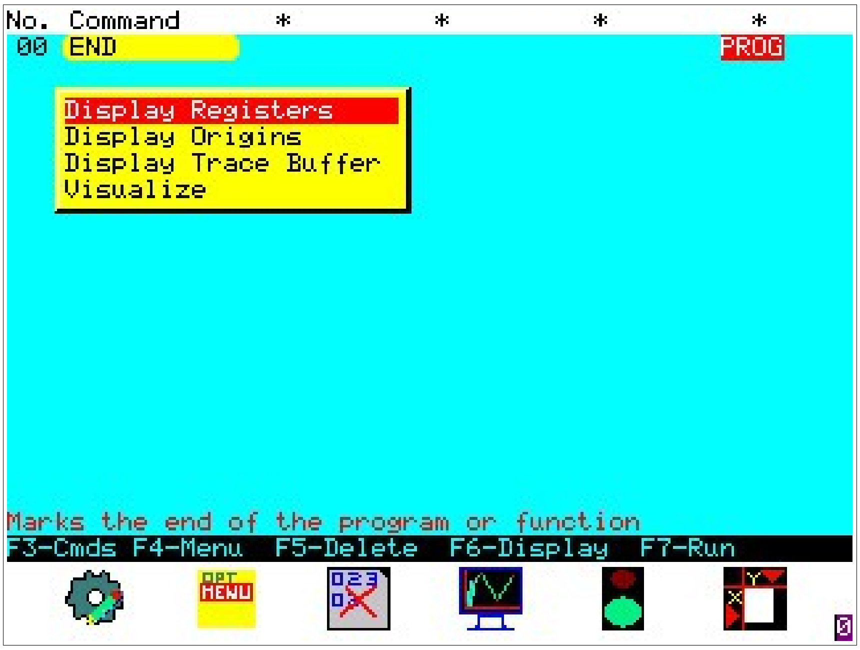


```
No. Command          *            *            *
 00  END          PROG

    Delete current line ENTER=YES

Marks the end of the program or function
F3-Cmds F4-Menu  F5-Delete  F6-Display  F7-Run
```

If any Jump commands or similar that require a specific line number are used in the program and are below the deleted line, the operator will be prompted to automatically adjust these numbers or leave as they are.

Display Registers and Graph

By pressing the corresponding key, a pop up menu will be drawn as follows:



```
No. Command       *         *         *         *
 00 END                                         PROG
```

Marks the end of the program or function

The user can scroll down between the options using the UP and Down keys and press ENTER to select an option.

Display Registers

Will show on screen the current value stored on each Register. Using the up and down arrows the user can scroll through the 256 registers.

Display Origins

This will display a small pop-up box with the x,y,z origins that have been set for the edges of the part to be machined.

Display Trace Buffer

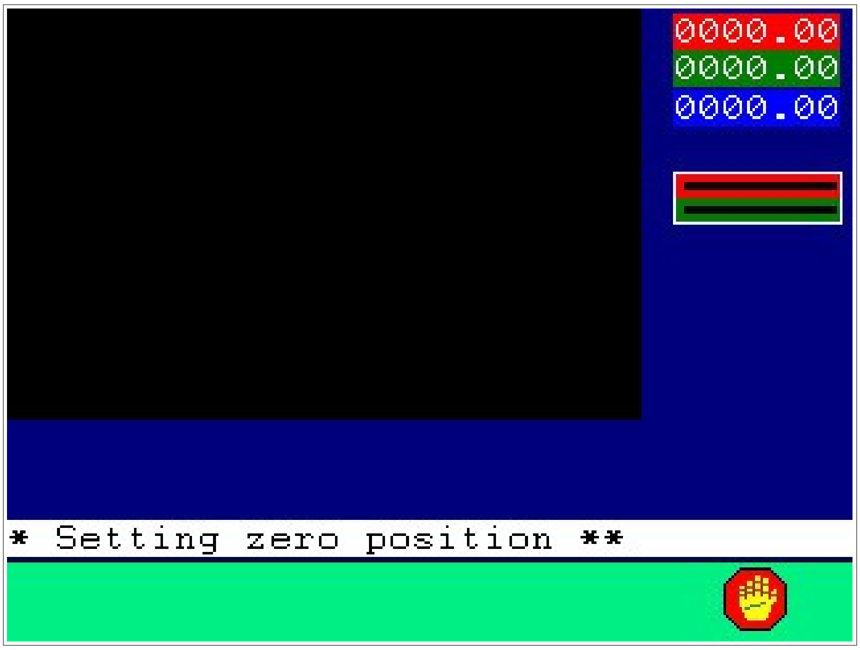
This feature shows the sequence of lines that have last been run. This is useful in debugging a program as you cannot always see where a jump has actually gone to or which condition was met. This is a circular buffer and gets overwritten and so only shows the last 256 lines run.

Visualise

This option shows the toolpath that is generated by the commands entered. This is useful for checking the part generation and the envisaged shape.

Running a Program

Once the database has been updated with the users program the controller is ready for use. Use the F7 key to start the motion control program running. To stop the machine press the STOP key or MENU key. When the controller is running the program, it will display the canvas as currently set up.



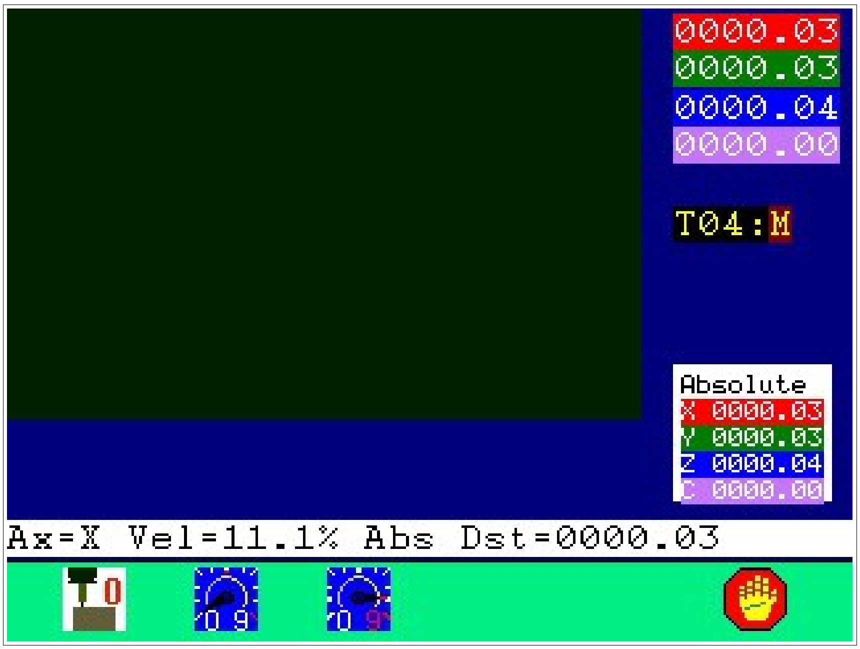
```
0000.00
0000.00
0000.00
* Setting zero position **
```

Stop

This Icon is only displayed when the program is running. By pressing F8 the program will Stop the execution of the program at the current line. Running the program again will resume from the position stopped at. Exiting the menu and returning will cause the program to start from the beginning.

Jog Screen

This screen allows the user to move to a position with the axis under control. The axis are torque limited as some machines need to be jogged whilst guards are open, speed is also set at a reduced rate.



```
0000.03
0000.03
0000.04
0000.00

T04:M

Absolute
X 0000.03
Y 0000.03
Z 0000.04
C 0000.00

Ax=X Vel=11.1% Abs Dst=0000.03
```

Axis Select

Use the Letter key assigned to an axis or the Left and Right arrow keys to select between the axis that are enabled on the controller. This will change the Ax= value to the axis selected.

Tool Height Set

Using the tool height set option will move the tool at a pre-set speed to contact the tool setter which must be positioned accurately under the tool position. Also check that the setter is clean and no swarf is trapped under the device which will give a false reading. Likewise the tool itself should be clean before performing this operation.

Speed Decrease

Pressing or holding this key decreases the speed for the selected axis within the limits set for that axis.

Speed Increase

Pressing or holding this key increases the speed for the selected axis within the limits set for that axis.

Stop Motion

Pressing this key terminates the motion, brings the motors to a controlled stop and disables the drives.

Destination

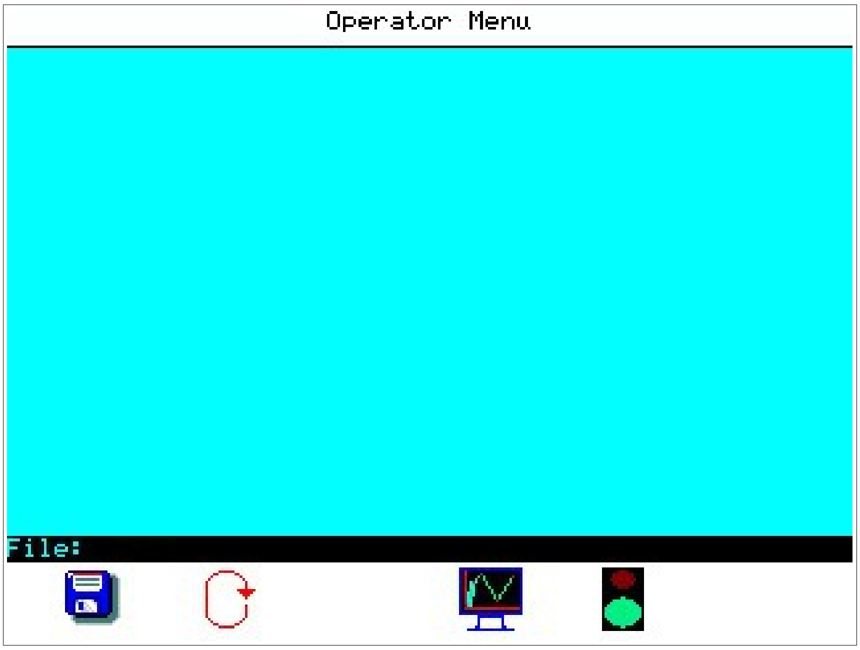
The destination for an axis can be set in different ways. The external Jog Wheel is one. These tend to have axis selection inputs as well as coarse or fine speed adjustments and are useful for setting positions as the tool and job can be viewed whilst manually moving the axis.

If fitted the internal jog wheel can be used along with the keypad to select the axis, or the position typed in directly with the numeric keypad. The up and down arrow keys can also be used to scroll the destination, if an accurate position is required we suggest that position is typed directly.

Operators Screen

Operator's Screen

Select the Operator's screen from the main menu and you will be presented with a screen as set-up using the Canvas options. All screens will have the same bottom row of icons regardless of how the canvas is set-up.



This screen is designed to execute a program already created, however it is not possible to modify a program from here. There are very few facilities in this menu, these are

- Select a file
- Enter the number cycles to run
- Start and Stop the process
- Display Graphs

Open File

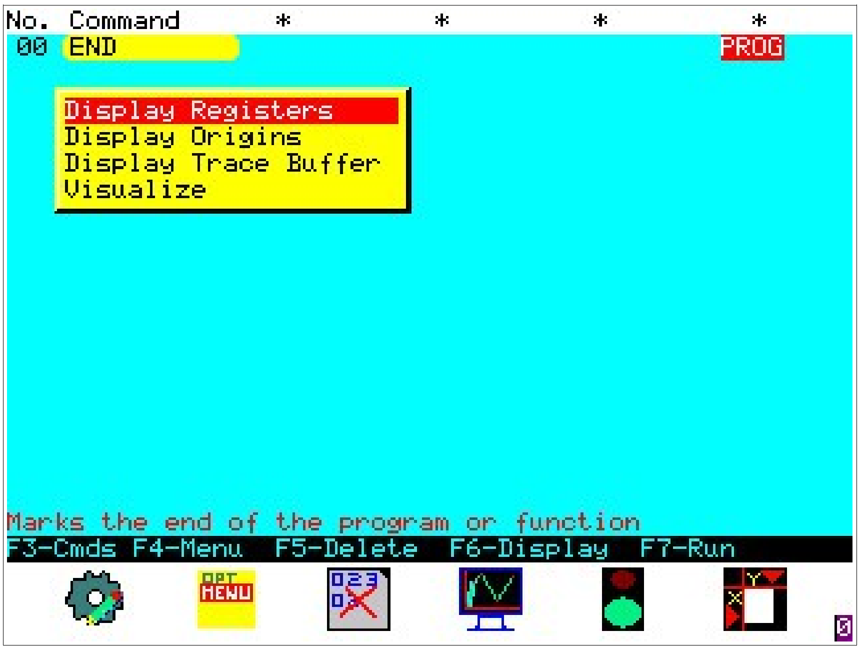
The operator menu can not be used without at least one product in existence. To open the product that is required for production press the key below the icon open that is F3 on the controller's keypad. The controller will draw the retrieve file menu as shown in Figure 25 in the next section. Use the up and down arrow keys to select the required file, this will be highlighted in white. Press ENTER to load the selected file or MENU to return to the operator menu. After a file has been selected it's name will appear in the file name box.

Loop

If the cycles quantity required is greater than 0 (zero) the system will continue to operate until the counter is counted down to zero. If the counter is zero when the process is started it is the same as "Don't count" and therefore the process will continue without limit. To enter a value in the counter press the key below the icon that is F4, the controller will display a message box containing the words "Enter loop quantity ->_______ at the prompt enter the quantity required then press enter. After the command the controller will display the new target count. This is a temporary setting and the counter will be set to '0' automatically after the loop is finish.

Display Registers and Graph

By pressing the corresponding key, a pop up menu will be drawn as follows:



```
No. Command       *         *         *         *
 00 END                                       PROG
```

Marks the end of the program or function

The user can scroll down between the options using the UP and Down keys and press ENTER to select an option.

Display Registers

Will show on screen the current value stored on each Register. Using the up and down arrows the user can scroll through the 256 registers.

Display Origins

This will display a small pop-up box with the x,y,z origins that have been set for the edges of the part to be machined.

Display Trace Buffer

This feature shows the sequence of lines that have last been run. This is useful in debugging a program as you cannot always see where a jump has actually gone to or which condition was met. This is a circular buffer and gets overwritten and so only shows the last 256 lines run.

Visualise

This option shows the toolpath that is generated by the commands entered. This is useful for checking the part generation and the envisaged shape.

Run

Once the system has a valid file loaded the process can be started. The program can be loaded either by the edit create menu or the memory but an error message will be display if the operator tries to run an empty data base. To start the process simply press the key below the icon, that is F5. The first time the system is running all axis that have valid home switch settings will start by moving towards the home position. After homing the system will commence the program

Stop

This Icon is only displayed when the program is running. Pressing the stop key terminates the motion, brings the motors to a controlled stop and disables the drives.

END

We hope that you find the information you need in this manual. In the event that you are unsure or find that something is not covered please in the first instance contact your local distributor. If they are unable to help, it will then be referred back to TRM's Technical support staff.

Notes:

Identifying an axis: If it rotates around a line parallel to

X it’s an A axis

Y it’s B

Z it’s C