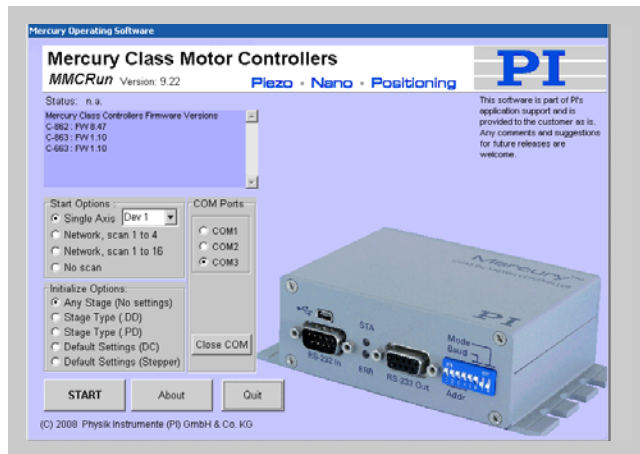


# MS 139E Software Manual

## MMCRun

### Mercury™ Operating Software

Release: 2.0.0      Date: 2007-12-19



This document describes software for use with the following product(s):

- C-862 Mercury™ Networkable Single-Axis DC-Motor Controller
- C-663 Mercury™ Step Networkable Single-Axis Stepper Motor Controller
- C-863 Mercury™ Networkable Single-Axis DC-Motor Controller



© Physik Instrumente (PI) GmbH & Co. KG  
 Auf der Römerstr. 1 · 76228 Karlsruhe, Germany  
 Tel. +49 721 4846-0 · Fax: +49 721 4846-299  
 info@pi.ws · www.pi.ws

The following designations are protected company names or registered trademarks of third parties:  
Windows, LabView

Physik Instrumente (PI) GmbH & Co. KG is the owner of the following company names and trademarks:  
Mercury™ Step, PI®

Copyright 1999–2007 by Physik Instrumente (PI) GmbH & Co. KG, Karlsruhe, Germany.  
The text, photographs and drawings in this manual enjoy copyright protection. With regard thereto, Physik Instrumente (PI) GmbH & Co. KG reserves all rights. Use of said text, photographs and drawings is permitted only in part and only upon citation of the source.

First printing 2007-12-19  
Document Number MS 139E, Release 2.0.0  
MMCRun\_MS139E.doc

This manual has been provided for information only.  
Product specifications are subject to change without notice.

# About This Document

## Users of This Manual

This manual is designed to help the reader to work with the MMCRun operating software .

This document is available as PDF file on the product CD which comes with the controller hardware and the PI Motion CD. Updated releases are available for download on [www.pi.ws](http://www.pi.ws) or by email: contact your Physik Instrumente Sales Engineer or write [info@pi.ws](mailto:info@pi.ws).

## Related Documents

For the Mercury™ Controller product family, the following documents are available from [www.pi.ws](http://www.pi.ws) or from your PI Representative:

Hardware user manuals	User Manual for each hardware component
MMCRun, MS139E	Mercury Operating Software, native-command based (this document)
Mercury Native DLL & LabVIEW, MS177E	Windows DLL Library and LabView VIs (native-command based)
Mercury Native Commands, MS176E	Native Mercury™ Commands
Mercury GCSLabVIEW, MS149E	LabView VIs based on PI GCS command set
Mercury GCS DLL,MS154E	Windows DLL Library (GCS commands)
PIMikroMove User Manual, SM148E	PIMikroMove® Operating Software (GCS-based)
Mercury Commands, MS163E	Mercury™ GCS Commands
PIStageEditor, SM144E	Software for managing GCS stage-data database

# Contents

1	Introduction to MMCRun	3
1.1	System Requirements .....	3
1.2	Firmware Requirements .....	3
1.3	Installation .....	4
1.3.1	MMCRun .....	4
1.3.2	Setup .....	4
1.3.3	USB Drivers .....	4
1.4	Basic Features of <i>MMCRun</i> .....	5
1.4.1	Start Screen .....	5
1.4.2	Main Screen with Subwindows .....	6
1.4.3	Command Editor Window .....	9
1.4.4	Position Window .....	14
1.4.5	Mercury™ Macro Manager .....	14
1.4.6	Host Macro Editor .....	16
1.4.7	Macro File Manager .....	18
1.5	Windows Library (DLL) .....	19
2	Command List	20

# 1 Introduction to MMCRun

*MMCRun* is Windows operating software for C-862 and C-863 Mercury™ DC motor controllers, and C-663 Mercury™ Step controllers.

The software works on all Windows platforms and allows operating these controller types alone or in any multiple-axis combination.

*MMCRun* offers many features to handle Mercury™ controllers in a direct, convenient way, for command entry and for macro programming.

---

## 1.1 System Requirements

*MMCRun* software runs on all PCs under Windows 95/98/2000/NT and XP. Up to 16 controllers can be networked off a single RS-232 or USB\* port on the host PC. In a multi-axis network, the controllers are daisy chained together in an RS-232 bus architecture.

---

## 1.2 Firmware Requirements

This manual describes the *MMCRun* software version 9.22. For proper operation, the following combinations of firmware and *MMCRun* versions should be installed:

Controller	Firmware Version	<i>MMCRun</i> Versions	Program Name
C-862 Mercury™ (DC motors)	8.40	8.22	MMCRun822.EXE
	8.47	9.12 or higher	MMCRun912.EXE
C-663 Mercury™ Step	1.06	9.12** or higher	MMCRun912.EXE
C-863 Mercury™ (DC motors)	1.10 or higher	9.22	MMCRun922.EXE

---

\* USB control requires at least one C-863 or C-663 controller. If the host's RS-232 port is used to drive the bus, the current-sourcing capability of its output stages could limit the number of networkable devices to as few as 6. Because under Windows NT third party drivers are required to activate any host-PC USB ports, PI does not support use of the USB interface under that operating system.

---

## 1.3 Installation

### 1.3.1 MMCRun

A formal installation procedure for the *MMCRun* software is not required. Just copy the executable file into your working directory on your drive.

### 1.3.2 Setup

If you use the Setup procedure on the controller product CD (if any) *MMCRun* will be installed by it.

### 1.3.3 USB Drivers

If one of your controllers has a USB interface and you wish to use it, then you must install the USB drivers provided on the CD that came with the controller. Note that because Windows NT requires 3rd party support for the USB hardware on the host PC, PI does not actively support the USB interface under that OS.

#### CAUTION

Never connect the RS-232-IN and USB connectors of the same controller to a PC at the same time, as damage may result.



Install the drivers by connecting the controller to the host with the included USB cable (make sure no RS-232 cable is connected to the host), powering it up, and following the instructions that appear when the new hardware is discovered. Choose to select the drivers from a list, and enter the name of their directory on the CD (e.g. D:\Drivers\ ) when asked for a location to search.

#### NOTE

The USB drivers will make the USB interface appear to *MMCRun* as a new COM port. That port will be present only when the controller is connected via USB and powered-up.

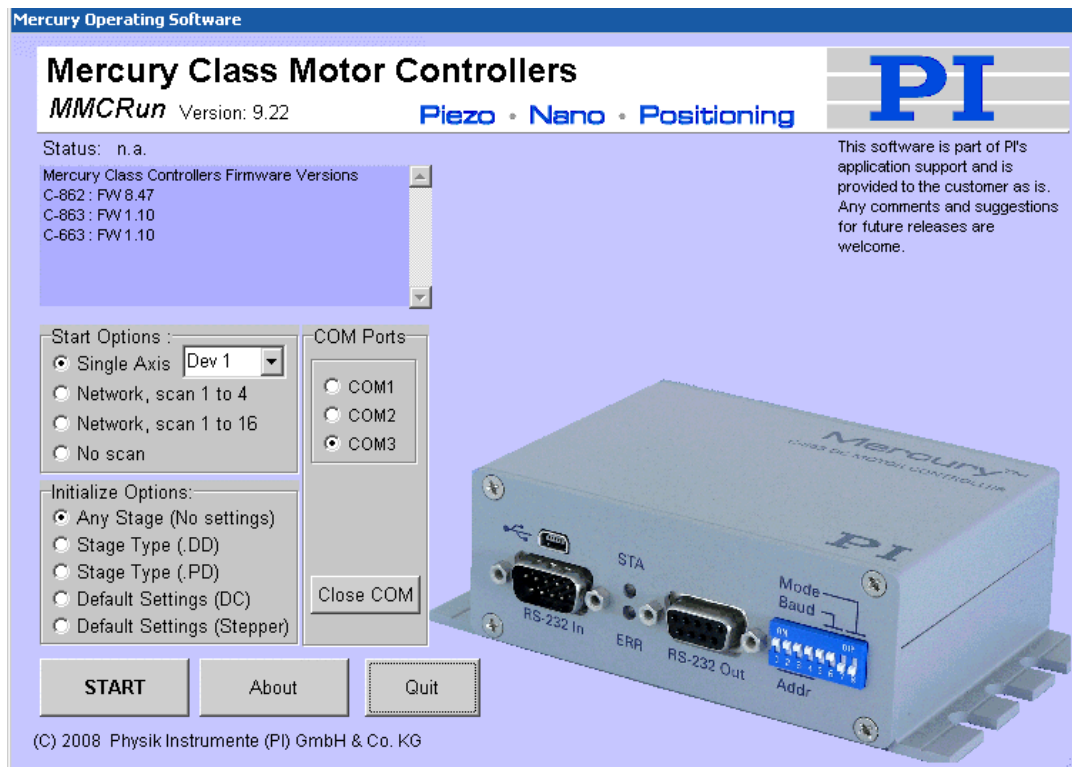
With current firmware, it may be necessary to power-cycle the controller while the host PC is on to establish communication with it.

To use the USB interface, simply select the “new” COM port on the opening screen when you start *MMCRun*. The baud rate setting is still relevant and must agree with that set at all controllers, even if there is only one. Typically DIP

switches are used for the controller-side setting,,with factory default of 9600; see the controller User Manuals for details.

## 1.4 Basic Features of *MMCRUn*

### 1.4.1 Start Screen



The *MMCRUn* software supports up to 16 Mercury™ controllers in a network. Any combination of Mercury™ and Mercury™ Step controllers can be networked. If only one unit is connected, select *Single Axis* and specify the Device Number.

### NOTE

The device numbers run from 1 to 16. Mercury™ class controllers typically use DIP switches for the address settings, and these settings begin with a (negative-logic) binary 0 for Device 1. See the controller User Manual for complete description.

If multiple Mercury™ controllers are networked, select either *Network, scan 1 to 4* (if all device numbers are smaller or equal to 4) or *Network, scan 1 to 16* (if any Mercury™ is set to a device number larger than 4).

If you know that each Mercury™ controller is properly configured for the motor drive connected, then choose the *Any stage* initialize option. In this case, no macros will be overridden and no address selection code or parameter changes will be sent to any controller. If you are using the Mercury™ controller the first time and you have mechanics of a special type, choose the corresponding *Initialize* option (in the current program version, the specified initialization settings are applied to the controller with the highest address only).

Move to the main screen with the *START* button.

## CAUTION

With stepper-motor stages, commanding a velocity above the maximum possible for the stage will cause the motor to stall.

Because stepper motors do not have position encoders, the position counter will continue to increment. The controller's motor position may not correspond with the actual motor position and this might endanger your application.

The maximum velocity depends on various influences, operating voltage, current phase current setting and mechanical load. Datasheet values are for orientation only and may not work in all applications.

Check out the maximum possible velocity for your individual application!

### 1.4.2 Main Screen with Subwindows

The main screen has a large pane which is a sort of desktop on which other program windows can be placed, as well as many buttons to perform specific functions.

To return to the start screen click *CLOSE* on the main screen.



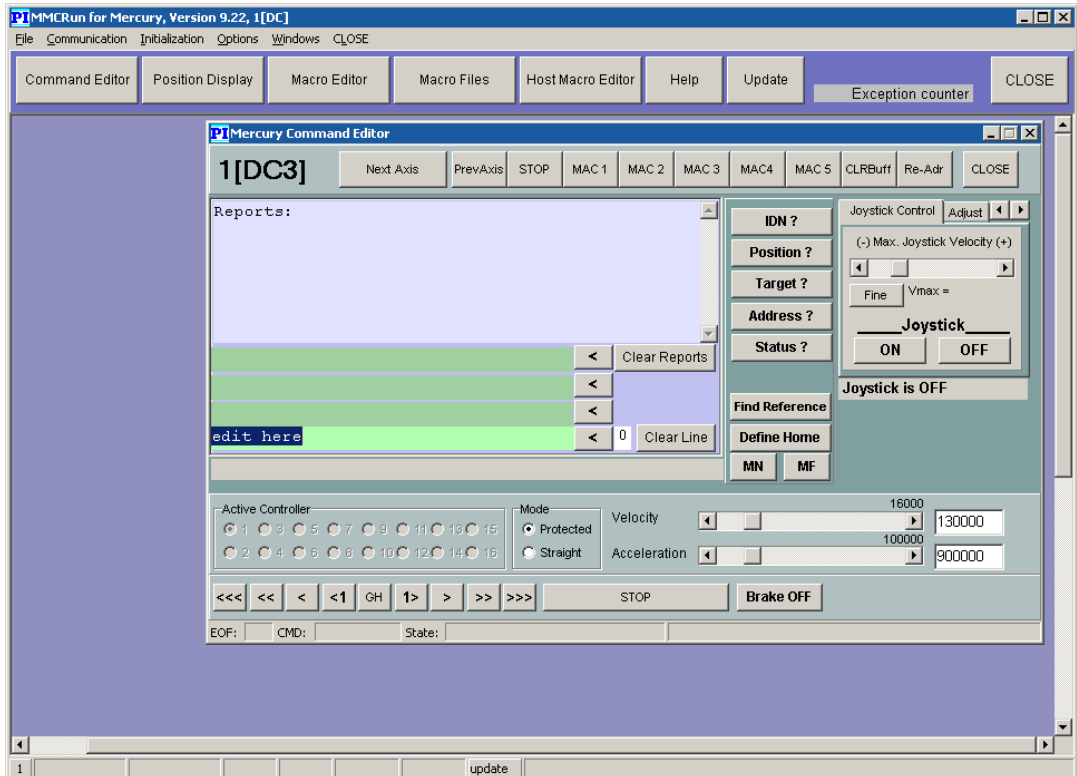


Fig. 1: Main screen with Command Editor as it appears upon entry; scrollbars indicate present of unseen surface

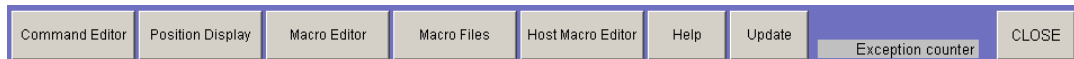
The main screen opens with the *Command Editor* window open. To display other windows, click on the corresponding toolbar button at the top of the main screen (*Position, Display, Macro Editor, Macro Files,...*).

## NOTE

If when you attempt to display a new window, a vertical scroll bar appears at the right of the main window instead; using it is one way to bring the desired window into view. Clicking the toolbar button is usually more convenient.

### Buttons at the Top of the Main Window

The toolbar buttons just below the menu bar have the functions described below. Most of them open a corresponding subwindow.



Command Editor	Opens the <i>Command Editor</i> window. It is used to set parameters, to move the motor and to retrieve reports from the device.
Position Display	Opens the real time position display window for the active axis.
Macro Editor	Opens the <i>Macro Manager</i> window. Use this tool to edit and display macro sequences stored inside the Mercury™.
Macro Files	Opens the window where you can load and save macro files
Host Macro Editor	Opens a command editing list or table for macro sequences running on the host PC
Help	Shows a list of commands
Update	Update the status line entries
CLOSE	Close the main window and return to the start screen

### 1.4.3 Command Editor Window

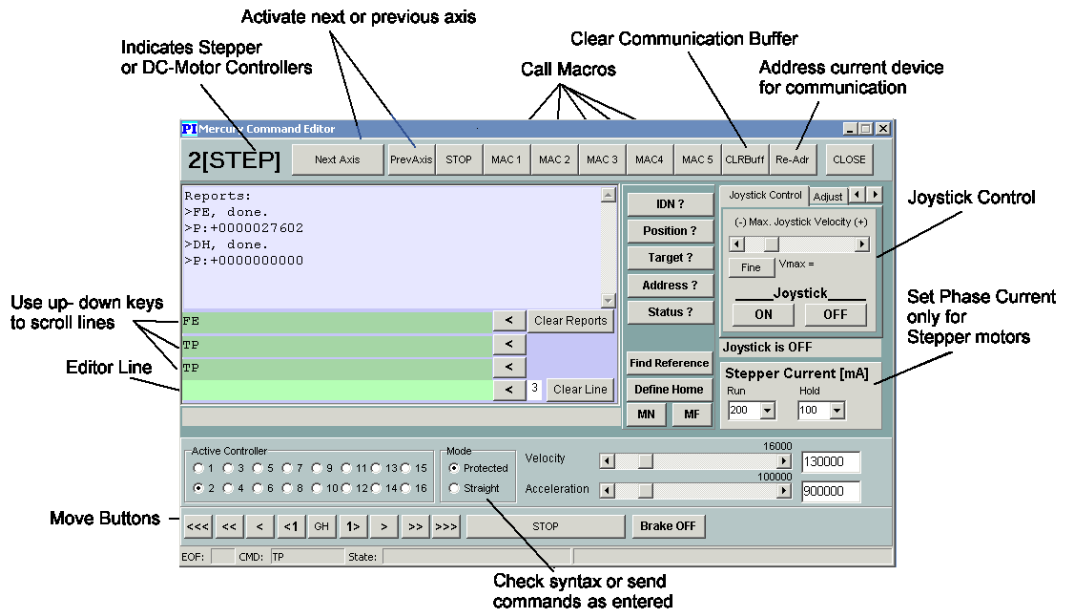
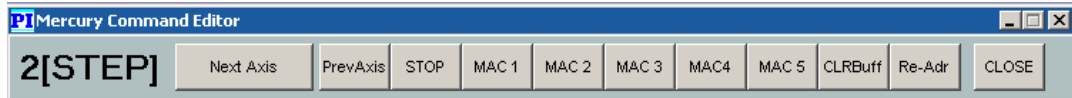


Fig. 2: Main screen controls: with stepper motor controllers, be sure velocity is set below stage maximum; if “Command unknown” appears, set mode to “straight” instead of “protected”

Buttons at Top:



- NextAxis Switches to device with next higher address
- PrevAxis Switches to previously-active device
- STOP Immediate motor stop
- MAC *n* Execute macro *n*
- CLR Buff Clears the communication buffer
- Re-Adr Send the *address-selection code* of the currently selected controller to re-address it. If power to the device was temporarily disconnected or you have connected a different stage, the device may have gone into the *deselected* state. Use this button to activate it again.
- Close Close *Command Editor* Window

## Query Buttons

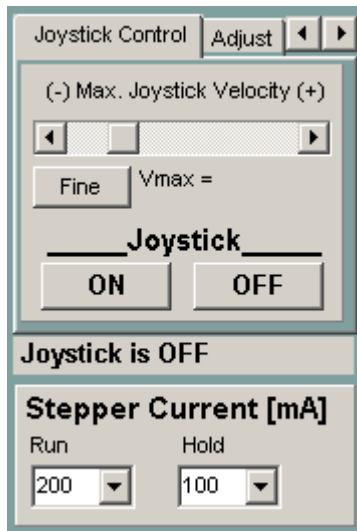


Fig. 3: Query buttons and Report pane with some query responses

The response to the queries will appear in the report pane at the left as the report from the corresponding command.

IDN?	Queries firmware version information
Position?	Queries current position of the active axis
Target?	Queries current target position
Address?	Gives <i>Address</i> and <i>Device number</i> (=address+1) of selected controller
Status	Queries controller status

## Joystick & Stepper



The joystick section refers to the one connected joystick axis. When activated (ON), the joystick position in that axis will determine the motor axis velocity. The joystick maximum velocity is set here. On the *Adjust* tab card it is possible to calibrate the joystick (calibrate before activation!)

With stepper motor controllers (C-663) the stepper current settings are shown and adjustable (note that the hold current is flowing whenever the controller is powered up and the motor connected).

Joystick Max.Velocity

If the joystick is connected and set to ON, this slide-bar can be used to set the maximum joystick velocity. Commanding too high a velocity with a stepper motor (Mercury™ Step controller) will cause the motor to stall, and invalidate the position counter.

Adjust

Clicking this button is the way to bring the Adjust tab card to the fore: it displays information facilitating adjustment of the joystick neutral position.

Joystick ON/OFF

Enables/disables joystick control. If the joystick is not in the neutral position, or not properly calibrated, it may not be recognized; if that occurs, press the *Adjust* button and calibrate the joystick.

As long as the joystick control is ON, move commands from the *Editor* line or *Move* buttons are not accepted.

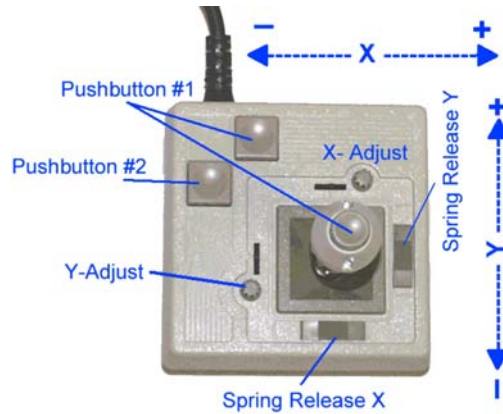
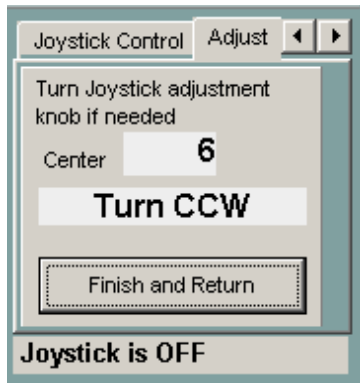
Stepper Current  
*Work*

Set the value to use for the stepper motor current while moving.

Stepper Current  
*Hold*

Set the value to use for the stepper motor current while not moving. This level is applied beginning a fixed time (default 0.7 s, see HT command) after the end of a move.

### Adjust Tab Card



Turn CCW—OK—Turn CW

Turn the corresponding Adjust knob on the joystick in the indicated direction until the *Center* number is close to zero and the status field indicates *OK*.

Finish and Return

Leaves the *Adjust* tab card

### NOTE

The joystick is connected directly to the controller(s), not to the host PC. If a 2-axis joystick is plugged into one controller, only the X-axis and both #1 buttons are active.

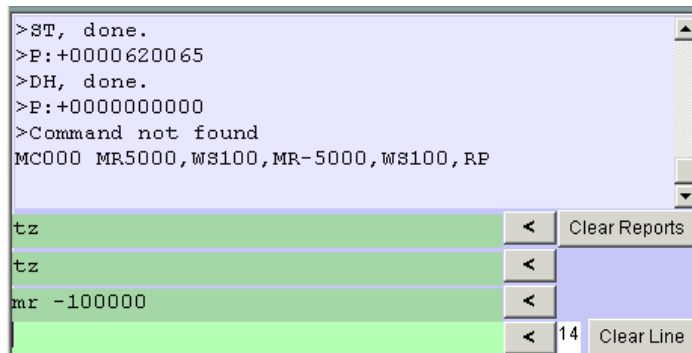
If a Y-cable is used, joystick power is taken from the X-axis branch, so it must be connected to a powered-up controller.

Bottom Row



- <<< Move relative -200,000 counts (negative direction)
- << Move relative -50,000 counts (negative direction)
- < Move relative -10,000 counts (negative direction)
- >>> Move relative +200,000 counts (positive direction)
- >> Move relative +50,000 counts (positive direction)
- > Move relative +10,000 counts (positive direction)
- GH go to zero position
- 1 Move one step in the negative direction
- >1 Move one step in the positive direction
- STOP Stops the current move and quits a running macro or compound command.
- Brake OFF If the stage has a mechanical brake, or uses the “brake on” signal to disable the motor, click this button to take the brake off.

Command and Report Panes



- Clear Reports Clear the report pane
- Clear Line Clear the command entry line
- < Send command (same as pressing the **ENTER** key)

Note: some commands are only recognized in *Straight* mode.

#### 1.4.4 Position Window

The *Position* window shows current position in counts or steps, depending on the controller type, obtained by polling.

If a Mercury™ DC motor controller is connected, the servo position error is also displayed.

Position polling can be enabled/disabled by checking/unchecking the check box next to the display.

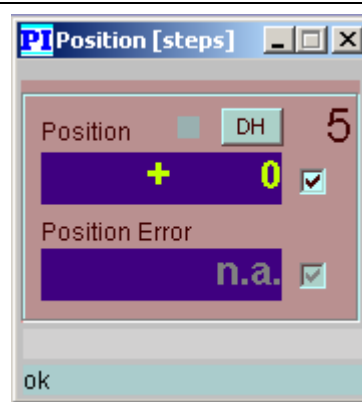


Fig. 4: Position window

#### 1.4.5 Mercury™ Macro Editor

The Macro Editor allows display and editing of macro command sequences stored in the controller.

The list in the window is initially blank. Clicking *Read List* reads any macros stored in the controller into the list.

Place the focus on a macro to edit by clicking the corresponding row, then press the **F2** key to activate the edit mode. After changes are made, press **ENTER** to send the command line to the controller.



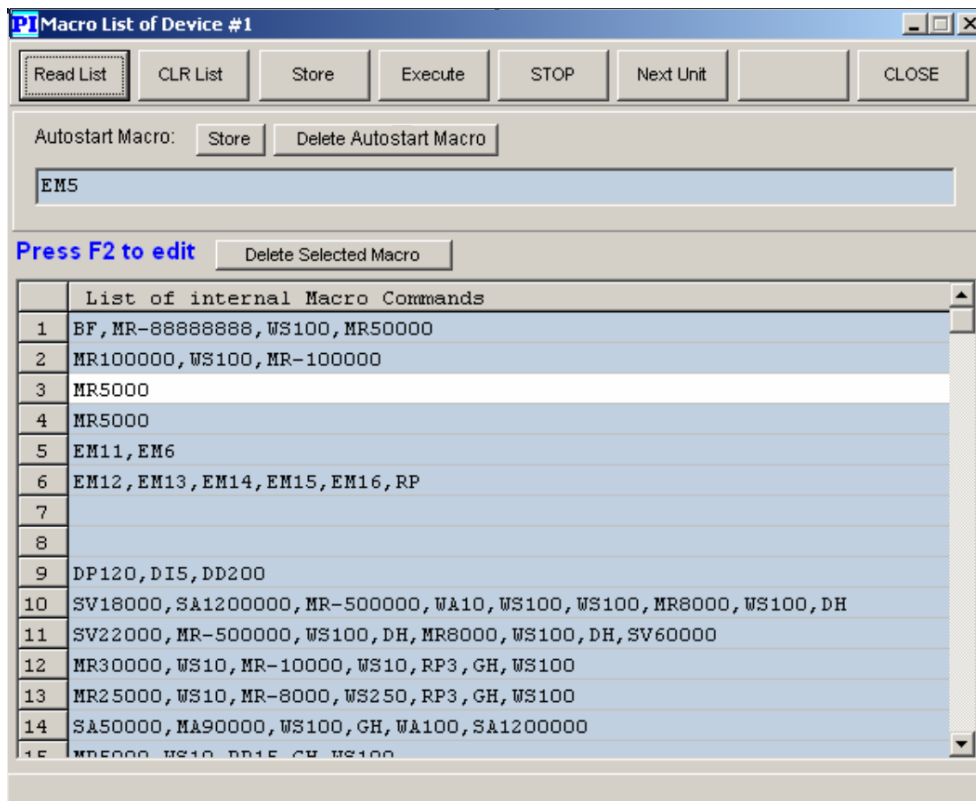


Fig. 5: Macro Manager window with focus on Macro #3

### Buttons on the Macro Window

Read List	Clears the list and fills it with the contents of all macros stored in the currently active device.
CLR List	Clears the list
Store (top line)	Writes the focused macro into non-volatile memory on the controller.  The same function is performed if the ENTER key is hit while you are focused in a command line.
Store (Autostart macro line)	Writes the autostart macro to non-volatile memory on the controller.  The same function is performed if the ENTER key is hit while you have the cursor in the autostart macro command line.
Execute	Runs the macro on the controller whose the number is the same as the line focused. (If you edit the line, first press ENTER to copy the command to the controller, then click "Execute" to run it).
STOP	Stops the motor and interrupts the running macro.
Next Unit	If a Mercury™ network is connected, this button will deselect the currently selected device and select the device with the next-highest address
CLOSE	Close the <i>Macro</i> window

### 1.4.6 Host Macro Editor

The *Host Macro Editor* allows creation and management sequences of commands that are stored in the host computer as ASCII data files (rather than in the controller itself).

Each macro file can contain up to 4 individual macro sequences that can call each other or can be used individually.

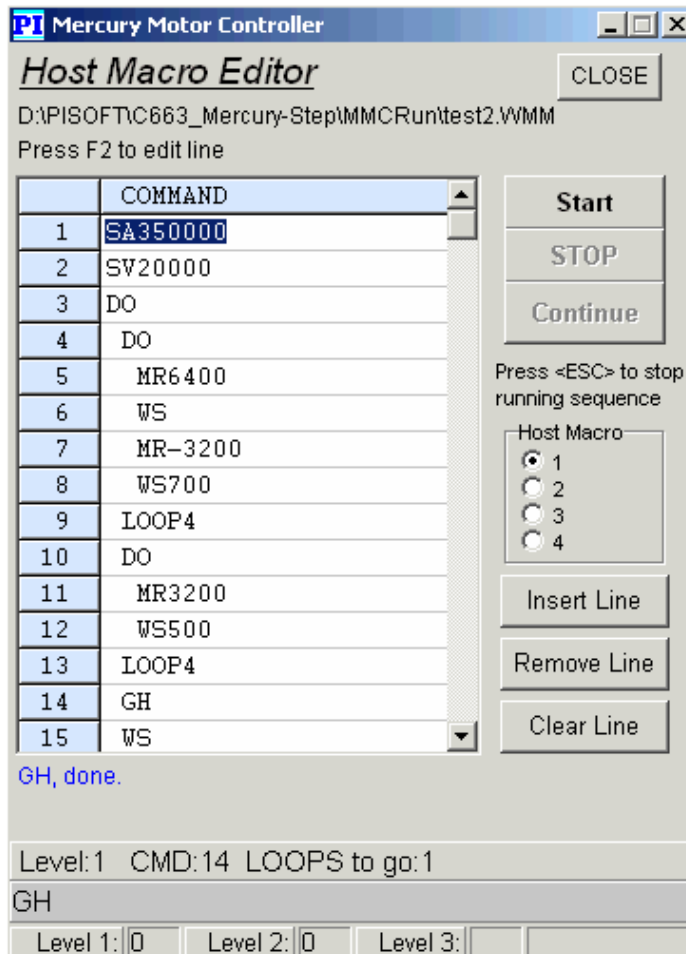


Fig. 6: Host Macro Editor window

The *Host Macro Editor* also makes it possible to sequence commands to different controllers on a network. In this way, the motion of one axis can be coordinated with that of another. The following commands are of special interest for building programmable loops and coordinating motion.

SCn:	Select controller number $n$ (range of $n$ : 1 to 16): when executed in a host macro, this command causes the host to send an <i>address-selection code</i> with address $n$ over the network
WSn	Wait for motor stop. This halts command execution until the move of the selected controller has terminated. Then an addition wait time of $n$ milliseconds is added.  Example: WS500 : waits 0.5 s after the motor has stopped
DO:	Start a DO-LOOP structure; for use in host macros only
LOOPn	End a DO-LOOP structure and branch to the corresponding DO until the loop has been executed $n$ times. For use in host macros only. Do-Loop structures can be nested 4 levels deep.

### 1.4.7 Macro File Manager

The Mercury™ File Manager can be used to manage the macros stored on the controller itself. With it, the controller macros can be downloaded, uploaded and copied to and from the host PC. On the host, the macro sequences are stored in ASCII files, so any text editor can be used to edit them.

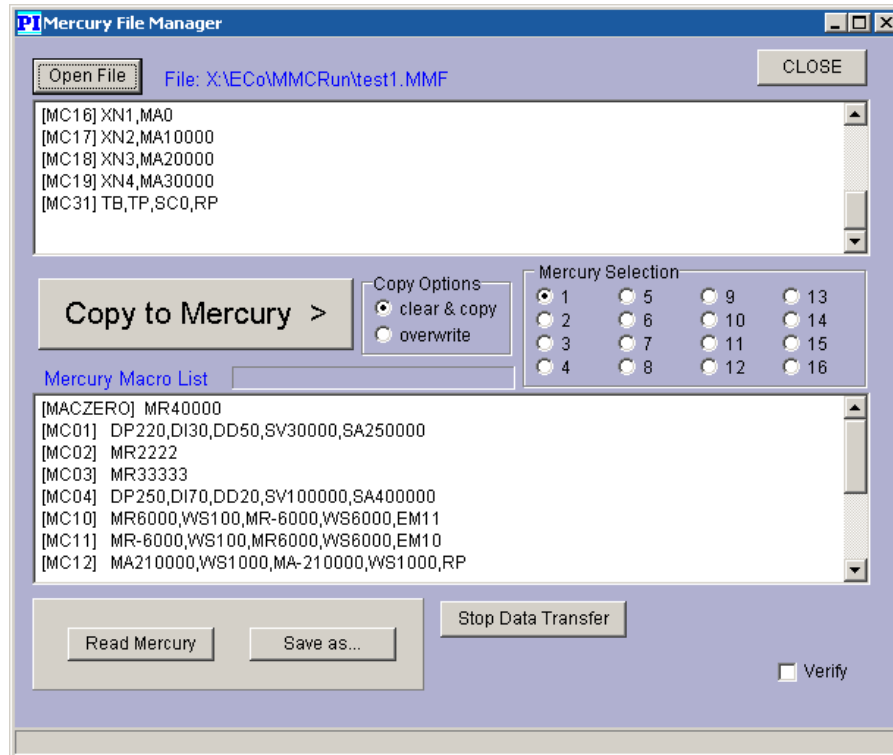


Fig. 7: Macro file manager window

#### Buttons in the File Manager Window

Open File	A file-open dialog appears and the macro file to load can be selected.
Copy to Mercury™	Depending on the copy options, the macro file is copied to the Mercury™ or Mercury™ Step controller.  If <i>clear &amp; copy</i> is selected, all macros currently stored in the controller are erased before the new macros are downloaded (stored in the Mercury™).  When <i>overwrite</i> is selected, the Mercury™ macro storage is not first cleared: the only macros erased are those that are overwritten by new macros.
Read Mercury™	Reads all currently stored macros in the device.
Save as...	The macro list previously read from the Mercury™ is now copied to an ASCII file.  In the file save dialog, the file name can be edited and warnings will be issued if the filename already exists.

---

## 1.5 Windows Library (DLL) and LabVIEW VIs

Mercury™ and Mercury™ Step controllers come with a common native-command-based Windows™ DLL library (MMC.DLL) that can be used with all 32-bit Windows development systems.

A LabVIEW driver set based on the native-DLL is also available.

For further information see the MMC DLL and LabView manual, document MS177, available as PDF on the product CD of the controller and on [www.pi.ws](http://www.pi.ws).

## 2 Native Command List

For detailed command descriptions see the Mercury™ Native Commands Manual.

Com- mand	1-char Alias	Function	Report Identifier	C-663	C-863	C-862
AB	!	Abort: Stop motion abruptly		☺	☺	☺
AB1		Abort softly			☺	☺
BF		Set brake OFF		☺	☺	☺
BN		Set brake ON		☺	☺	☺
CF		Channel OFF (digital output control)		☺	☺	☺
CN		Channel ON (digital output control)		☺	☺	☺
CP		Channel pattern (digital output control)		☺	☺	☺
CS		Report checksum	C:	☺	☺	☺
DC		Set phase current		☺		
DD		Set d-term			☺	☺
DH		Define home		☺	☺	☺
DI		Set i-term			☺	☺
DL		Set integration limit			☺	☺
DP		Set p-term			☺	☺
EF		Echo OFF			☺	☺
EM		Execute Macro		☺	☺	☺
EN		Echo ON			☺	☺
FE		Find edge (find origin position reference)		☺	☺	☺
GD		Get d-term	D:		☺	☺
GH		Go home		☺	☺	☺

<b>GI</b>	Get i-term	DI:		☺	☺
<b>GL</b>	Get integration limit	M:		☺	☺
<b>GP</b>	Get p-term	G:		☺	☺
<b>HC</b>	Set hold current		☺		
<b>HT</b>	Set hold time		☺		
<b>JF</b>	Set Joystick OFF		☺		
<b>JN</b>	Set Joystick ON		☺		
<b>JT</b>	Load predefined Joystick Table		☺		
<b>LF</b>	Limit switch operation OFF		☺	☺	☺
<b>LH</b>	Limit switches active-high		☺	☺	☺
<b>LL</b>	Limits switches active-low		☺	☺	☺
<b>LN</b>	Limit switch operation ON		☺	☺	☺
<b>MA</b>	Move absolute		☺	☺	☺
<b>MD</b>	Define Macro		☺	☺	☺
<b>MF</b>	Motor off			☺	☺
<b>MN</b>	Motor on			☺	☺
<b>MR</b>	Move relative		☺	☺	☺
<b>RM</b>	Remove (erase) macro		☺	☺	☺
<b>RMAL</b>	Remove macros and reset parameters		☺		
<b>L</b>					
<b>RP</b>	Repeat from beginning of line		☺	☺	☺
<b>RT</b>	Reset (like power-on reset)		☺	☺	☺
<b>RZ</b>	Remove macro zero (autostart macro)		☺	☺	☺
<b>SA</b>	Set Acceleration		☺	☺	☺
<b>SC</b>	Select/deselect controller		☺	☺	☺

<b>SI</b>		Set joystick table index		☺		
<b>SI?</b>		Get joystick table index	I0:	☺		
<b>SJ</b>		Set joystick table value		☺		
<b>SJ?</b>		Get joystick table value	J0:	☺		
<b>SM</b>		Set maximum following error			☺	☺
<b>ST</b>		Stop motion smoothly		☺	☺	☺
<b>SV</b>		Set Velocity		☺	☺	☺
<b>TA</b>		Tell analog input value or joystick position	TAn:	☺	☺	☺
<b>TB</b>		Tell board address	B:	☺	☺	☺
<b>TC</b>	#	Tell channel (digital input)	H0n:	☺	☺	☺
<b>TD</b>		Tell d-term	N:		☺	☺
<b>TE</b>	?	Tell position error	E:		☺	☺
<b>TF</b>	(	Tell profile error	F:		☺	☺
<b>TI</b>		Tell iteration number	X:	☺	☺	☺
<b>TL</b>		Tell programmed acceleration	L:	☺	☺	☺
<b>TM</b>		Tell macro contents	MC001.. .	☺	☺	☺
<b>TP</b>	'	Tell position	P:	☺	☺	☺
<b>TS</b>	%	Tell status	S:	☺	☺	☺
<b>TT</b>		Tell target position	T:	☺	☺	☺
<b>TV</b>		Tell current velocity	V:		☺	☺
<b>TY</b>		Tell programmed velocity	Y:	☺	☺	☺
<b>TZ</b>		Tell Macro Zero	MC000.. .	☺	☺	☺
<b>UD</b>		Update: store current parameters as new defaults			☺	☺
<b>VE</b>		Display version number	(C)2006 ...	☺	☺	☺





<b>WA</b>	Wait absolute time	☺	☺	☺
<b>WF</b>	Wait while channel OFF	☺	☺	☺
<b>WN</b>	Wait while channel ON	☺	☺	☺
<b>WS</b>	Wait until stop	☺	☺	☺
<b>XF</b>	Execute if input channel OFF (TTL-low)	☺	☺	☺
<b>XN</b>	Execute if input channel ON (TTL-high)	☺	☺	☺
<b>YN</b>	Execute if joystick button is pressed	☺		
<b>YF</b>	Execute if joystick button is released	☺		

