

Gerety/Chase E-Roll Player System

MIDI Controller User's Guide

Document Revision: 1.2

Revision Date: July 6, 2004

1 Introduction

The Gerety/Chase E-Roll Player system includes one of the most advanced, user-configurable MIDI controllers available on the market today. This powerful MIDI controller includes built-in operating software that lets you "converse" with the controller via its built-in serial port to "browse" the controller's current configuration and to change that configuration in just about any way you like. Any changes you make are remembered by the controller, even when the power is off.

Some of the many advanced features of the Gerety/Chase MIDI controller are:

- the ability to "browse" the current configuration
- "MIDI Activity" control over AC (or DC) power, with user-specifiable turn-off delay
- specify the MIDI channel number to which the controller will respond (OMNI mode can also be specified)
- unrestricted control over the mapping of MIDI notes (and controllers) to digital outputs and/or valves. This lets you "unscramble" non-sequential wiring/tubing and/or accommodate non-chromatic scales.
- Load, list, edit and save MIDI maps
- Automatically identify the order of valve blocks (in a pneumatic system)

These and other features of the Gerety/Chase MIDI controller are described in greater detail in the remainder of this document.

2 Overview

The Gerety/Chase MIDI controller processes MIDI messages received over its MIDI IN plug. All MIDI IN messages are repeated over the MIDI THRU plug, allowing the MIDI controller to be included in a chain of controllers.

As MIDI messages are received, note and controller events are compared against a "**MIDI map**" stored within the controller that identifies what actions should be taken for each event. This map associates specific MIDI note with valves, digital outputs, or the SSR (AC/DC power) control. Some MIDI note and/or controller events may be left "unassigned", in which case the controller takes no action when those events are received. When a note or controller event is received for which there is an assignment defined, the controller will take the appropriate action.

For example, in a typical Duo-Art MIDI map, MIDI notes 18 through 113 are assigned to valves sequentially starting with the first valve on the first block (B1V1) and ending with the last valve (#16) on the last block (#16, *i.e.*, B6V16). In addition, MIDI controller #64 (sustain pedal) is mapped to B1V1 (which controls the sustain pedal on a Duo-Art piano) and MIDI controller #67 (soft pedal) is mapped to B6V16 (which controls the soft pedal). If a MIDI "ON" event is received for MIDI note number 23, then B1V6 is turned on (opened). When a MIDI "OFF" event is received for the same note, the valve is turned off (closed) again.

MIDI maps can be defined by the user, or can be copied from pre-defined maps stored within the controller. The user creates or edits a map in a "**working map**", then saves the working map into the controller's non-volatile memory as a "**user map**". By loading pre-stored maps (including the user map) into the working map, they can be edited/modified/stored at any time. Further, by means of ordinary serial communication software, maps can be recorded into text files, edited, and copied back to the controller. The controller uses a simple, user-friendly language to define maps.

When the controller is first powered up, it examines its non-volatile memory to see if there is a pre-stored user map. If so, it uses that map to define how the controller should respond to MIDI events. If not, it falls back to a default map stored within the controller. The controller's non-volatile memory can be erased by the user at any time, effectively restoring the controller to its original factory settings.

After checking for a user map at power-up, the controller next checks for a saved "**blocklist**" defining the ID numbers and physical position of the valve blocks connected to the controller. Since the controller can access ID numbers for all connected blocks, but has no way to determine their physical location, it is necessary to identify the "**block order**" for all blocks connected to the controller. It is this information that is stored in the blocklist. If there is no saved blocklist, then they are

assumed to be organized such that the physical positions of the blocks are organized according to ID number order (as reported by the controller at power-up).

3 Controller Communications

3.1 Communication Connections

To communicate with the controller from a PC, it is necessary to connect a 9-pin serial communications cable between your PC's serial port and the serial port connector on the MIDI controller. The cable should be a D-Sub 9-pin Female to Female, straight-through connector. If you do not have such a cable, it can be ordered from a variety of suppliers, such as Digikey and Cables-to-go. The Digikey part number is AE1017-ND. The Cables-to-go part number is #02695.

3.2 Communication Software

Any serial communication software with terminal emulation capability is suitable for communicating between the PC and your MIDI controller. HyperTerminal[®] by Hilgraeve Software, which is shipped with Microsoft[®] Windows[®] versions is already available on many computers. Many other suitable freeware and shareware communications programs are available via the Internet.

3.2.1 Using HyperTerminal

The following instructions should help you get started configuring and using HyperTerminal to communicate with your MIDI controller:

Setting Up/Starting HyperTerminal

If you've never set up a HyperTerminal connection before, if you want to create a new HyperTerminal connection with new settings, or if you are just not sure, follow steps 1 - 5 below:

INITIAL SETUP:

Connect a 12V power supply to the MIDI controller, if you are doing a "bench setup". The green power light should be on. (When installed in an instrument, the power supply is already hooked up. Just verify that the power is on by checking for the green power light.)

Connect a serial cable between your computer's serial port and the "serial config" connector on the front panel of the MIDI controller.

To create a HyperTerminal connection for the MIDI controller:

1) Start HyperTerminal. On most computers this is located under:

Start->Programs->Accessories or

Start->Programs->Accessories->Communications

- 2) The program responds with a "New Connection" Dialog. Enter a name for the connection (e.g., "MIDI Controller Programming"), and optionally pick an icon, then click OK
- 3) The program responds with a "Connect To" dialog box. Select "Direct to COM1" (or whatever your serial port happens to be) from the "Connect Using" drop-down list. All of the other settings will "gray out". Now click OK.
- 4) The program responds with a "COM1 Properties" dialog box. Choose the following settings, then click OK:
 - Bits per second: 57600
 - Data bits: 8
 - Parity: None
 - Stop bits: 1
 - Flow control: None
- 5) You should now be connected. From the Menu bar choose File->Save to save your connection configuration to disk. You will probably not need to enter a new connection name. The name you entered previously should automatically be selected.

You are now ready to begin using HyperTerminal!

Press the "RESET" button on the MIDI controller. You should see a message similar to the following:

```
*** GERETY/CHASE MIDI CONTROLLER (C) 2004 ***
      Firmware revision 3.12
Cont. Serial # = 017C1BCC080000DA
Config:
  Ctlr Name: MY CONTROLLER
  Motor Off Timeout = 30 secs.
Detected blocks:
  Blk #  ID
  ----  -
    1  0529080F0000004D
    2  05290C0F00000052
    3  05290C0F00000052
    4  05350B0F00000025
    5  050D0F0F00000076
    6  0573060F0000000E
Saved Blocklist:
  Blk #  Order  ID
  ----  -
    1     5  0529080F0000004D
    2     2  05290C0F00000052
    3     4  05290C0F00000052
    4     3  05350B0F00000025
    5     6  050D0F0F00000076
    6     1  0573060F0000000E
Map Name = Ampico A
Chan = 0
Map specifies 6 blocks, 0 blocks detected

CMD>
```

You are now ready to communicate with the controller as described later in sections 4 and 5 of this document.

TO STORE A USER MAP FROM A TEXT FILE INTO THE CONTROLLER:

- 1) Select **Transfer->Send Text File** from the Hyperterminal menu bar
- 2) Navigate to your map file and select it to start loading it into the controller. The controller will respond with:

Programming...

followed by:

OK.
CMD>

The whole process takes a second or two.

For the future:

Whenever you want to call up your settings again without going through steps 1-5 above, start up HyperTerminal, but cancel out of the "New Connection" dialog by clicking "Cancel". Then either click on the open folder icon or choose File->Open. Your settings will show as a file with the name you selected and an extension of ".ht". Choose that file, and you should be connected again.

3.3 Using HyperTerminal to Update the MIDI controller Firmware

In the very unlikely event that you need to update the MIDI controller's firmware, this can also be done via the controller's serial port as follows:

HyperTerminal should be up and running at 57600 baud, 8 data bits, no parity, 1 stop bit, no flow control, as described in section 3.2 above, with a serial cable connected between your computer's serial port and the MIDI controller's "Serial Config" connector on the front panel. 12V power should be applied to the MIDI controller and the green power light should be lit.

- 1) Press and Hold Reset button until step 8 below. The button MUST be held throughout the entire process below. If the button is accidentally released, start the process over.

-- with reset button held...

- 2) press the **"Enter"** key. MIDI Controller should respond with prompt similar to the following:

DS89C420 LOADER VERSION X.X COPYRIGHT (C) XXXX DALLAS SEMICONDUCTOR
>

3) Type the letter "**K**" followed by the **Enter** key. After a few seconds the controller will respond with the ">" prompt again. This step erases the MIDI controller.

4) Type the letter "**L**" followed by the **Enter** key. This starts the controller's built in loader program. The controller will behave as if it is not responding. It is actually waiting for you to send a "hex" file to program it.

5) Select **Transfer->Send Text File** from the Hyperterminal menu bar

6) In the "Send Text File" Dialog Box, change "**Files of Type:**" to "**All Files (*.*)**"

7) Find and double-click on your hex file to start loading it into the controller. The controller will begin programming itself, echoing a few rows of G's as it goes. The "G's" are indications that a row of data from the Hex file was successfully loaded. (This step may take a couple of minutes to complete)

8) When the controller finishes programming (there should be quite a few rows of GGGGG...) it will respond with the ">" prompt again. Release the reset button - You're done!

4 Command Set

The following commands can be entered when the MIDI controller produces a "CMD>" prompt. Hitting <ENTER> without entering a command will cause the controller to produce another "CMD>" prompt without taking any action. All commands are entered into the controller via the <ENTER> key.

4.1 HELP or '?' command

The "HELP" has several forms. You can enter either HELP or just enter a question mark (?) to get a list of controller commands. You can also type "HELP MAP" or "? MAP" to get a list of *map mode* subcommands. In response to "HELP" or "?", the controller responds as follows:

```
CMD>HELP
Commands:
  HELP or '?'
  HELP MAP
  ECHO ON/OFF
  MIDIMON ON/OFF
  IOMON ON/OFF
  RESET
  MAP
CMD>
```

Note that in all command examples, text entered by the user is shown in **RED**, while text generated by the controller is shown in **BLUE**.

In response to "HELP MAP" or "? MAP", the controller responds as follows:

```
CMD>HELP MAP
MAP commands
<esc> to quit
CONFIG time,name
DIR ALL, MAP, BLK or CFG
NEW 'name'
LOAD map#
LIST
DUMP
  Nx (0-127) = BnVm (n,m = [1 to 16])
  Nx-y = BnVm
  Nx = DOn (n = 1-4)
  Nx = SSR ON or OFF
  Cx = (same options as Nx)
  Nx or Cx = <blank> to unmap
  MC = chan (0-16, 0 = OMNI)
IDENT
NAME name
ERASE ALL, MAP, BLK or CFG
SAVE
RESET
```



```
CMD>
```

4.2 ECHO command

The ECHO command controls whether or not your typed commands are echoed back to you by the controller. The ECHO OFF command turns off command echoing. ECHO ON turns it on. "ECHO" all by itself causes the controller to report whether echoing is off or on.

4.3 MIDIMON command

The MIDIMON command controls whether the controller reports MIDI events via its serial port. MIDIMON ON turns this feature on. MIDIMON OFF turns it off. MIDIMON alone causes the computer to report the current MIDIMON state. The <ESC> key cancels the MIDIMON command.

Example: In the following exchange, the user turns on MIDI monitoring. MIDI notes 48-52 are played in sequence on MIDI channel #1 and the controller reports the associated note-on and note-off events. Then the user turns monitoring off.

```
CMD>MIDIMON ON
CMD>
  ON c01n048
OFF c01n048
  ON c01n049=100
OFF c01n049
  ON c01n050=100
OFF c01n050
  ON c01n051=100
OFF c01n051
  ON c01n052=100
OFF c01n052
CMD>MIDIMON OFF
CMD>
```

4.4 IOMON command

The IOMON command is very similar to the MIDIMON, command, but instead of reporting MIDI events, it reports which valves, outputs, etc. it operates in response to those event. The <ESC> key cancels the IOMON command. The following is the IOMON result for the same sequence shown above for the MIDIMON command:

```
CMD>IOMON ON
CMD>
SSR ON
B03V01 ON
SSR ON
B03V01 OFF
SSR ON
B03V02 ON
B03V02 OFF
SSR ON
```

```
B03V03 ON
SSR ON
B03V03 OFF
SSR ON
B03V04 ON
SSR ON
B03V04 OFF
SSR ON
B03V05 ON
B03V05 OFF
CMD>IOMON OFF
CMD>
```

4.5 MAP command

The MAP command puts the controller into MAP mode, which is described in greater detail in section 5 below. In MAP mode, the controller generates a **MAP>** prompt, instead of the **CMD>** prompt. Either the escape key <ESC> or an EXIT command exits map mode.

4.6 RESET command

The RESET subcommand RESETS the controller, performing essentially the same function as the reset button on the controller's front panel.

5 Map Mode Subcommands

5.1 General Information about Map Mode and Mapping

MAP mode serves primarily two functions:

- 1) It allows the user to load, save, define and edit MIDI maps, and
- 2) It allows the user to specify the physical order of valve blocks.

5.2 Listing/Viewing/Configuration Subcommands

5.2.1 CONFIG subcommand

The CONFIG command lets the user set the *Activity monitor timeout value* and give the controller configuration a name that will be reported via the serial port each time the controller is powered up.

The MIDI activity monitor is a system by which AC power can be controlled by the presence or absence of relevant MIDI activity on the MIDI in jack. The MIDI activity monitor **ONLY** monitors events on the MIDI channel to which the controller is programmed to respond. If the controller is programmed for OMNI mode, then it responds to events on all 16 MIDI channels. (See MIDI map "CH" subcommand).

As soon as the activity monitor detects any relevant channel message, it turns the SSR output ON. The output goes off a user-programmable period of time after relevant MIDI activity ceases. The default value for the timeout is 20 seconds.

The format of the CONFIG command is:

CONFIG nn, name

where '**nn**' is the number of 2 second intervals to wait before turning off the SSR output after relevant MIDI activity ceases. For example, to program a 20 second timeout, enter "10" in place of "nn". *A value of "0" for "nn" disables the activity monitor.*

Everything following the comma (" name" in this case) is used as the controller name. To program the controller for a 60 second timeout, and to name the configuration "MY CONFIG", enter the following command:

```
MAP>CONFIG 30,MY CONFIG
      OK.
MAP>
```

ALWAYS RESET THE CONTROLLER BEFORE ATTEMPTING TO PLAY MIDI FILES AFTER COMPLETING THE CONFIG COMMAND. THE CHANGES ARE ONLY INSTALLED AFTER A RESET.

5.2.2 DIR subcommand

The DIR (directory) subcommand lists the currently installed maps, configuration (see config command) and blocklist. These can all be listed by simply typing DIR. Typing DIR CFG, DIR MAP or DIR BLK causes the controller to list the current configuration (CFG), installed maps (MAP) or blocklist (BLK). This is illustrated by the example session below:

```
CMD>map
Entering Map Mode
MAP Firmware v1.0 (c) 2004
MAP>DIR CFG
Directory of CFG...
Config:
  Ctlr Name: MY CONTROLLER
  Motor off delay: approx. 10 seconds
  OK.
MAP>DIR MAP
Directory of MAP...
Built-in maps:
  1) Ampico A, DO#1=soft
  2) Duo-Art (default)
User Map:
  3) Ampico A, DO#1=soft

  OK.
MAP>DIR BLK
Directory of BLK...
  Detected blocks:
    Blk #  ID
    ----  -
    1  0529080F0000004D
    2  05290C0F00000052
    3  05290C0F00000052
    4  05350B0F00000025
    5  050D0F0F00000076
    6  0573060F0000000E
  Saved Blocklist:
    Blk #  Order  ID
    ----  -
    1      5  0529080F0000004D
    2      2  05290C0F00000052
    3      4  05290C0F00000052
    4      3  05350B0F00000025
    5      6  050D0F0F00000076
    6      1  0573060F0000000E

  OK.
MAP>
```

5.2.3 LIST subcommand

The LIST subcommand lists the current *working map* in a summary form. Note that to have a working map available to list, it is necessary to create a new one (see the "NEW" subcommand) or load one (see the "LOAD" subcommand). For example, the following is a "LIST" of a controller map for a typical Ampico A piano (the LOAD 3 command creates a working map and loads it with the current user map).

```
MAP>LOAD 3
  Map [Ampico A] loaded.
MAP>LIST
N8 = DO2
N9 = DO3
N10 = DO4
N16-105 = B1V1
N107-110 = B6V11
N111 = DO1
N112-113 = B6V15
C64 = B1V3
C67 = DO1
CH 0
MAP>
```

Examining this listing line-by-line, the map defines the following assignments:

- MIDI note 8 is assigned to digital output #2
- MIDI note 9 is assigned to digital output #3
- MIDI note 10 is assigned to digital output #4
- MIDI notes 16 through 105 are assigned to valve sequentially, starting with the first valve on the first block (B1V1) and ending with the tenth valve on the sixth block (B6V10). (Only the first valve is reported for a sequential assignment.)
- MIDI note 111 (the Ampico soft pedal tracker hole) is assigned to digital output #1 (to control a discrete solenoid valve)
- MIDI notes 112 and 113 are assigned to B6V15 and B6V16
- MIDI controller #64 (sustain) is assigned to B1V3
- MIDI controller #67 (soft pedal) is assigned to DO1 (same as note 111); and
- the map is assigned in OMNI mode (ch 0) to respond to MIDI events on all MIDI channels.

Maps are listed using the exact same commands that would be required to create them manually. That is, if all of the listed items were typed manually into a new working map and then saved, the same user map would be created.

5.2.4 DUMP subcommand

The DUMP subcommand is very similar to the LIST subcommand, but in addition it includes controller MAP subcommands to set up, name, and store the map. If the controller's response to the DUMP command is copied into a text file, it can be

transferred to the controller and stored (e.g., using HyperTerminal as described in section 3)

If the DUMP command were used instead of the LIST command in section 5.2.2, the controller would respond as follows:

```
MAP>LOAD 3
  Map [Ampico A] loaded.
MAP>DUMP
ECHO OFF
NEW Ampico A
N8 = DO2
N9 = DO3
N10 = DO4
N16-105 = B1V1
N107-110 = B6V11
N111 = DO1
N112-113 = B6V15
C64 = B1V3
C67 = DO1
CH 0
SAVE
ECHO ON

MAP>
```

If the text starting with ECHO OFF and ending with ECHO ON were copied to a file, then transferred to the controller at the MAP prompt, the controller would respond as follows:

```
MAP>ECHO OFF
  New map [AMPICO A] created.
  Programming...
  OK.
MAP>
```

Note that the leading "ECHO OFF" command suppresses echoing of the commands during the file transfer. Also note that the user doesn't actually type anything. The commands are stored in the MAP file.

5.2.5 HELP subcommand

The HELP subcommand in MAP mode is exactly the same as in CMD mode.

5.3 Map Editing Subcommands

5.3.1 LOAD subcommand

The LOAD command creates an empty working map and loads it with a user specified map. LOAD 1 and LOAD 2 cause the preconfigured maps (Ampico A- 1 or Duo Art- 2) to be loaded into the working map. To retrieve the user map, enter LOAD 3.

Example:

```
MAP>LOAD 3
  Map [Ampico A] loaded.
MAP>
```

5.3.2 NEW subcommand

The NEW subcommand creates a new working map. Any text following "NEW" on the command line is used as the title of the map.

Example:

```
MAP>NEW my map
  New map [MY MAP] created.
MAP>
```

5.3.3 NAME subcommand

The NAME subcommand lets you rename a working map. The text following "NAME" on the command line becomes the name of the map.

Example:

```
MAP>NEW my map
  New map [MY MAP] created.
MAP>NAME my other map
  OK.
MAP>DUMP
ECHO OFF
NEW MY OTHER MAP
CH 0
SAVE
ECHO ON
MAP>
```

5.3.4 ERASE subcommand

The ERASE subcommand has several forms:

- ERASE or ERASE ALL cause the stored user map, blocklist and configuration to be erased. This effectively returns the controller to its factory default condition.
- ERASE MAP erases only the user map
- ERASE CFG erases only the configuration (timeout and configuration name)
- ERASE BLK erases only the blocklist

ALWAYS RESET THE CONTROLLER BEFORE ATTEMPTING TO PLAY MIDI FILES AFTER COMPLETING ANY ERASE COMMAND

5.3.5 SAVE subcommand

The SAVE subcommand saves the current working map into the user map. Note that the user map must be erased before the SAVE command will work.

Example:

```
MAP>LOAD 3
  Map [AMPICO A] loaded.
MAP>CH 5
  OK.
MAP>NAME AMPICO A, MIDI CH 5
MAP>SAVE
  User map present.  ERASE MAP first.
MAP>ERASE MAP
Erasing MAP...
  OK.
MAP>SAVE
Programming...
  OK.
MAP>DIR
Directory of MAP...
Built-in maps:
  1) Ampico A, DO#1=soft
  2) Duo-Art (default)
User Map:
  3) AMPICO A, MIDI CH 5

  OK.
MAP>
```

ALWAYS RESET THE CONTROLLER BEFORE ATTEMPTING TO PLAY MIDI FILES AFTER COMPLETING A SAVE COMMAND

5.3.6 CH subcommand

The CH subcommand lets you specify for the working map which controller will respond to. The number following "CH" on the command line specifies the MIDI channel. CH 0 specifies "OMNI" mode, whereby the controller responds to all MIDI channels. This command has no effect unless the working map is saved to the user map via the SAVE command.

5.3.7 Assignment subcommands

The assignment subcommands actually do the work of building maps. All map commands have the format:

<Event> = <Destination>

An *<Event>* is either a MIDI note, a series of sequential MIDI notes, or a MIDI controller. A *<Destination>* specifies a valve (or a starting valve in the case of a series of notes), a digital output, or an SSR (solid state relay) command.

5.3.7.1 "Destinations"

The destination portion of an assignment command can be any of the following:

DOn	Assign <i><event></i> to digital output <i>#n</i> (<i>n = 1-4</i>)
BbVv	Assign <i><event></i> to Block <i>#b</i> Valve <i>#v</i> (<i>b = 1-8, v = 1-16</i>) or to a series of valves starting with Block <i>#b</i> Valve <i>#v</i> .
SSR ON	Turn the SSR output on in response to <i><event></i>
SSR OFF	Turn the SSR output off in response to <i><event></i>
<blank>	A blank destination causes <i><event></i> to be un-mapped. That is, any current assignment of <i><event></i> to a <i><destination></i> is cleared.

5.3.7.2 "N" (assign MIDI note) subcommand

Note assignment specifiers are as follows:

Nnnn	MIDI note <i>#'nnn'</i> , when <i>nnn</i> can be 0-127
Nnnn-mmm	A series of notes from <i>#'nnn'</i> to <i>#'mmm'</i> where <i>nnn < mmm</i> and <i>nnn</i> and <i>mmm</i> can be 0-127. This command form can only be used with a valve destination specifier or a blank destination specifier. Any other destination type will cause an error.

5.3.7.3 "C" (assign MIDI controller) subcommand

Controller assignment specifiers are as follows

Cnnn MIDI controller #'nnn', when nnn can be 0-127

5.4 IDENT subcommand

The IDENT subcommand lets you specify the physical order of blocks in your system. It sequentially "wiggles" the blocks (i.e., repeatedly activates the valves in a cyclic pattern that can be felt and heard) and asks you to type in the ordinal (the number in sequence) of the block that is currently "wiggling". If the first valve block that "wiggles" is the sixth one in your system, enter "6" in response to the prompt. Do this until all valves have been identified. When you are done, the controller automatically builds and saves a block map.

ALWAYS RESET THE CONTROLLER BEFORE ATTEMPTING TO PLAY MIDI FILES AFTER COMPLETING THE IDENT COMMAND

5.5 EXIT subcommand

The EXIT command (or simply the <escape> key) causes the controller to exit MAP mode and return to CMD mode.

5.6 RESET subcommand

The RESET command behaves exactly the same in MAP mode as in CMD mode. It causes the processor to be reset as if the front panel RESET button had been pressed

6 Examples of common operations

The following examples show how to perform some of the more common operations.

6.1 Changing and viewing the MIDI activity timeout

```
CMD>map ; enter map mode
Entering Map Mode
MAP Firmware v1.0 (c) 2004
MAP>config 23,My 46 second timeout config ; config and give a new name
OK.
MAP>dir cfg ; list the configuration
Directory of CFG...
Config:
  Ctr Name: MY 46 SECOND TIMEOUT CONFIG
  Motor off delay: approx. 46 seconds
OK.
MAP>exit ; exit map mode
Exiting MAP mode
CMD>reset ; reset the controller
```

6.2 Changing the MIDI channel for a user map

```
CMD>map ; enter map mode
Entering Map Mode
MAP Firmware v1.0 (c) 2004
MAP>dir map ; get a listing of the available maps
Directory of MAP...
Built-in maps:
  1) Ampico A, DO#1=soft
  2) Duo-Art (default)
User Map:
  3) AMPICO A, MIDI CH 5

OK.
MAP>load 3 ; load the user map into the working map
Map [AMPICO A, MIDI CH 5] loaded.
MAP>ch 4 ; change the MIDI channel to #4
OK.
MAP>save ; attempt to save - OOPS!
User map present. ERASE MAP first.
MAP>erase map ; let's fix that
Erasing MAP...
OK.
MAP>save ; now let's save for real...
Programming...
OK.
MAP>exit ; exit map mode
Exiting MAP mode
CMD>reset ; reset the controller
```

6.3 Changing the MIDI channel for the default map

```
CMD>map           ; enter map mode
Entering Map Mode
MAP Firmware v1.0 (c) 2004
MAP>dir map       ; get a listing of the available maps
Directory of MAP...
Built-in maps:
  1) Ampico A, DO#1=soft
  2) Duo-Art (default)
No User Map saved

OK.
MAP>load 2        ; load the default map (#2)
Map [Duo-Art (default)] loaded.
MAP>ch 4          ; change the MIDI channel to #4
OK.
MAP>list
N7 = DO1
N8 = DO2
N9 = DO3
N10 = DO4
N18-113 = B1V1
C64 = B1V1
C67 = B6V16
CH 4
MAP>save
Programming...
OK.
MAP>exit          ; exit map mode
Exiting MAP mode
CMD>reset         ; reset the controller
```

6.4 Erasing the user MAP

```
CMD>map           ; enter map mode
Entering Map Mode
MAP Firmware v1.0 (c) 2004
MAP>erase map     ; erase the user map
Erasing MAP...
OK.
MAP>exit          ; exit map mode
Exiting MAP mode
CMD>reset         ; reset the controller
```

6.5 DUMPing the user map (to copy to a file)

```
CMD>map                ; enter map mode
Entering Map Mode
MAP Firmware v1.0 (c) 2004
MAP>dir map            ; get a list of all of the available maps
Directory of MAP...
Built-in maps:
  1) Ampico A, DO#1=soft
  2) Duo-Art (default)
User Map:
  3) AMPICO A

OK.
MAP>load 3             ; load the user map (#3)
Map [AMPICO A] loaded.
MAP>dump              ; "dump" the map to your display
ECHO OFF              ; then cut and paste the text starting with
NEW AMPICO A          ; "ECHO OFF" and ending with "ECHO ON" into
N8 = DO2              ; a text file
N9 = DO3
N10 = DO4
N16-105 = B1V1
N107-110 = B6V11
N111 = DO1
N112-113 = B6V15
C64 = B1V3
C67 = DO1
CH 5
SAVE
ECHO ON

MAP>exit              ; exit map mode
Exiting MAP mode
CMD>
```

6.6 Getting a list of the general commands

```
CMD>help              ; general help command
Commands:
  HELP or '?'
  HELP MAP
  ECHO ON/OFF
  MIDIMON ON/OFF
  IOMON ON/OFF
  RESET
  MAP

CMD>
```

6.7 Getting a list of the map mode subcommands

```
CMD>help map           ; help on map subcommands
MAP commands
<esc> to quit
CONFIG time,name
DIR ALL, MAP, BLK or CFG
NEW 'name'
LOAD map#
LIST
DUMP
Nx (0-127) = BnVm (n,m = [1 to 16])
Nx-y = BnVm
Nx = DOn (n = 1-4)
Nx = SSR ON or OFF
Cx = (same options as Nx)
Nx or Cx = <blank> to unmap
CH = chan (0-16, 0 = OMNI)
IDENT
NAME name
ERASE ALL, MAP, BLK or CFG
SAVE
RESET
CMD>
```

6.8 Erasing the controller to factory default

```
CMD>map                 ; enter map mode
Entering Map Mode
MAP Firmware v1.0 (c) 2004
MAP>erase all          ; erase controller config storage
Erasing ALL...
OK.
MAP>reset              ; reset the controller

*** GERETY/CHASE MIDI CONTROLLER (C) 2004 ***
      Firmware revision 3.17

Cont. Serial # = 017C1BCC080000DA
Config:
  Ctlr Name: default
  Motor off delay: 20 secs
No block directory
  Using default block order
Using default map: [Duo-Art (default)]
Chan = 0
Map specifies 6 blocks, 0 blocks detected

CMD>
```