

MC433Convert.exe Utility Manual

Version 1.0

Motor Control Software Version 0.99

MC433Convert.exe

SOC Robotics, Inc.

www.soc-robotics.com

Warranty and Limitation of Liability

This software is provided free of charge and as such comes with no warranty of any kind or obligation to fix, modify, upgrade or change any features. We do, however, welcome any comments, recommendations or suggestions so we can continue to upgrade and improve the program. Notwithstanding anything to the contrary contained herein, SOC Robotics shall not, under any circumstances, be liable to User or any third parties for consequential, incidental, indirect, exemplary, special, or other damages. Use this software at your own risk.

© Copyright 2009. SOC Robotics, Inc. All rights reserved.

The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein.

Table of Contents

1.0 Overview

2.0 Configuration Command Description

- 2.1 Introduction
- 2.2 Default Settings
- 2.3 Stepper Motor Parameters
- 2.4 PWM Parameters
- 2.5 Step Mode Selection
- 2.6 Automatic Motor Current Shut Off
- 2.7 Motor Torque Reduction
- 2.8 Step and Direction Polarity
- 2.9 Limit/eStop Switch Detection
- 2.10 I2C Address

3.0 Programming the Motor Controllers

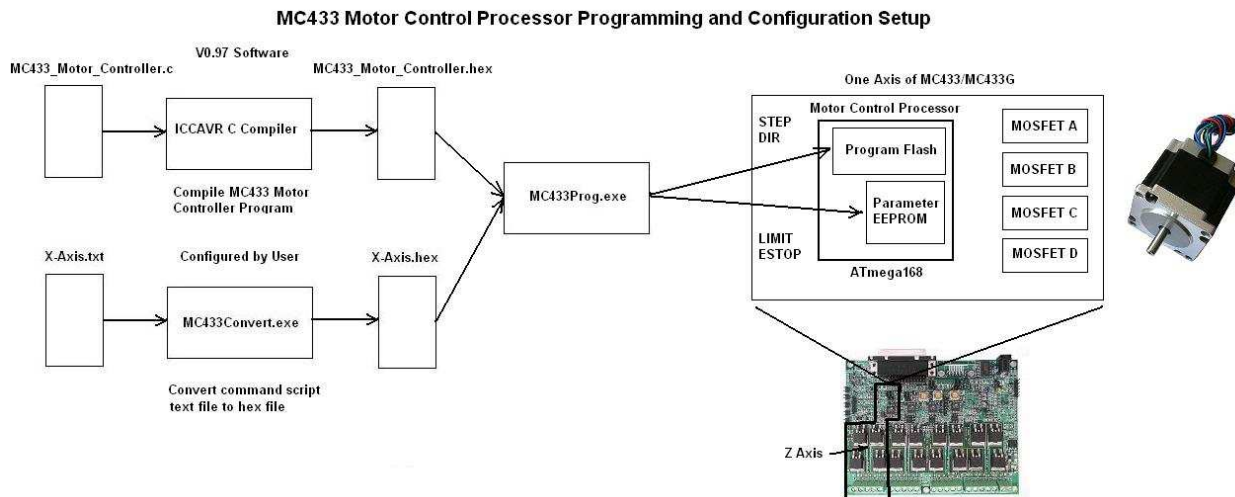
4.0 Example Configuration Files

1.0 Overview

1.1 Introduction

With the introduction of MC433/MC433G Motor Control Software Version 0.99 (and higher versions) it is now possible to configure the characteristics of each motor axis individually by loading parameters from a compiled script file created on the desktop into each axis processor's EEPROM memory. MC433Convert.exe is a PC desktop command line utility that translates a file of MC433 Motor Controller Configuration Commands (script) into a format that can be loaded into an MC433 Motor Controller's EEPROM. The latest motor control software version is 0.99.

The MC433 and MC433G have four Motor Control Processors (ATmega168) – one for each axis. Each Motor Control Processor has internal Flash memory and EEPROM memory. The Flash memory holds the program that implements the motor control functions such as following step/direction inputs, setting stepper motor coil phase current, recognition of limit switch state and other functions. The EEPROM memory holds the default settings of all parameters that configure the motor control program at power up. If a motor control function needs to be changed, such as enabling limit switch detection, the script file for that axis is edited with the new settings. The edited file is then converted by MC433Convert.exe to a Hex file that is loaded into EEPROM memory using MC433Prog.exe. The Flash is only programmed when a new version of the Motor Controller Software is installed such as the migration from V0.99 to V1.00. The next time the Motor Control Processor is powered up the new parameters are loaded from EEPROM.



If motors with different inductance and resistance are connected to the MC433 then PWM settings must be set differently for each motor. The MC433 is shipped with potentiometer recognition turn on – whatever the pots are set to determines the PWM characteristics of all four axis. The pot must be disabled if PWM characteristics need to be set for each axis differently. Scripts provide a means to tailor the operation of each Motor Controller to meet the needs of each motor.

1.2 Configuration Control

Each Motor Controller can be loaded with different configuration parameters to suit the specific needs of each axis stepper motor. For example, in a small mill application the X and Y axis may

be controlled by a 125oz-in NEMA23 stepper motor while the Z axis is controlled by a 186oz-in motor. The Z axis PWM settings must be changed to suit the different characteristics of the higher torque motor. Once the correct PWM settings are determined the text file is then converted into an Intel Hex file that is loaded into the ATmega168's EEPROM using programming utility MC433Prog.exe.

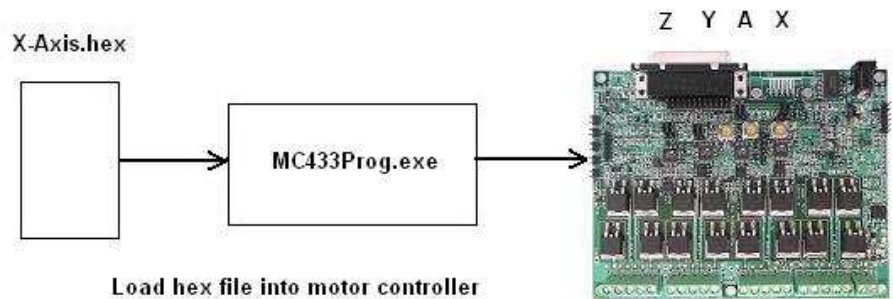
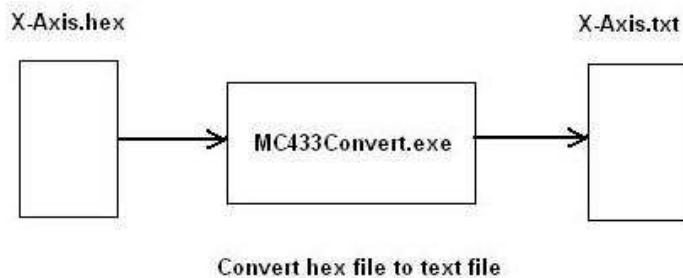
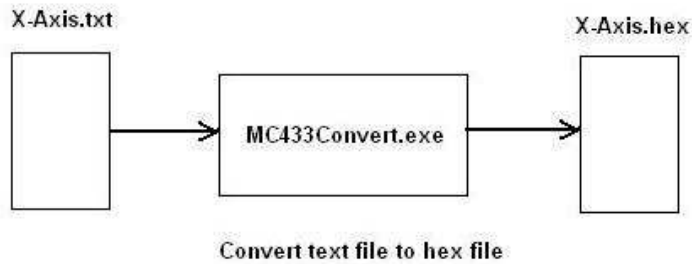
1.3 Configuring Different PWM Settings for Different Stepper Motors

Stepper motor coils have inductance and resistance that can vary with different motors. Resistance sets the current at which the motor reaches maximum torque for a specific drive voltage. Inductance determines how fast the maximum current and hence torque is reached. The higher the inductance the longer it takes to reach maximum torque and hence top speed. Lower inductance motors typically have a higher top speed for a given drive voltage. The MC433 PWM settings on time, duty cycle and slew rate must be adjusted to suit the specific motor. If different motors are attached to the MC433 then different PWM settings must be used on each axis. By default all PWM settings are the same for each axis. To set different PWM settings pot following must be turned off and the three PWM settings set for each axis individually.

The quickest way to set PWM characters is to attach one motor at a time to the MC433, adjust the pots for optimum performance and then read the PWM settings using MC433Prog.exe. PWM settings are loaded into EEPROM.

1.3 Programming Flow

MC433Convert.exe converts a text file to an Intel Hex file or an Intel Hex file to a text file. **MC433Prog.exe** loads the hex file into the Motor Controller's EEPROM. **MC433Prog.exe** allows the contents of the Motor Control EEPROM to be uploaded to the desktop. **MC433Convert.exe** can then convert the Hex file into a text file. This provides a way to use the pots initially to set the PWM characteristics for a specific stepper motor that is then duplicated in subsequent MC433 configuration setups. **MC433Convert.exe** uses the file extension (.hex or .txt) to determine how to process the file.



2.0 Configuration Command Description

2.1 Introduction

Motor Controller configuration parameters are grouped into eight major categories:

- Stepper Motor Parameters
- PWM Parameters
- Step Mode Selection
- Automatic Motor Current Shut Off
- Motor Torque Reduction
- Step and Direction Polarity
- Limit/eStop Switch Detection
- I2C Address

Each category may have one or more settable parameters. Only those parameters you want to change need to be entered although it is good practice to enter all parameters rather than rely on pre-programmed defaults. A parameter is entered by giving it's name followed by a value. The range of correct values is defined in this section.

2.2 Default Settings

The default settings of all Command settings as shipped from the factory are as follows:

Parameter	Default Value	Unit
Stepper Motor Parameters		
followmotorparameters	no	NA
resistance	0.70	ohms
inductance	2.10	mhenry
current	3.00	amps
drive_voltage	24.0	volts
PWM Parameters		
followpots	yes	NA
ontime	128	NA
dutycycle	128	NA
slewrates	2	NA
recoverytime	10	usecond
Step Mode Selection		
followstepheader	yes	NA
stepmode	wave	NA
Automatic Motor Current Shut Off		
automotorshutoff	yes	NA
motorshutofftime	8000	msecond
Motor Torque Reduction		
motorpowerreductionmode	on	
motorpowerreduction	50%	
powerreductiontime	4000	mseconds
Step and Direction Polarity		
step polarity	positive	NA
direction polarity	highcw	NA
Limit/eStop Switch Detection	yes	NA
I2C Address	0x10	Hexidecimal

2.3 Stepper Motor Parameters (Rev 1.0 feature only)

Stepper motors are characterized by inductance, resistance, current and are driven at a specific voltage. The inductance and resistance of the motor is fixed and can't be changed while the drive voltage is set by the stepper motor power supply voltage. These four parameters determine the performance of the motor. The higher the drive voltage the faster the motor will turn.

By enabling this mode MC433Convert automatically calculates the best PWM settings for your motor. Note that this mode is only available in Motor Control Software version 1.0 and is ignored by Motor Control Software version 0.99. This mode is enabled by the command:

```
followmotorparameters    yes
```

followed by the motor resistance, inductance, current and drive voltage.

```
followmotorparameters [yes|no]
resistance 0.70
inductance 2.10
current 3.00
drive_voltage 44.00
```

2.4 PWM Parameters

Pulse Width Modulation parameters must be set for the specific type of stepper motor attached to each MC433 axis. These parameters can be set using the onboard potentiometers or by loading these values using the script. Note that if the followpots command is active then the entered values are ignored. By default the motor control software follows the potentiometer settings. If PWM settings are adjusted by the potentiometers and MC433Prog.exe is used to read the contents of the EEPROM then MC433Convert.exe can be used to determine the values of the ontime, dutycycle and slewrate parameters.

Ontime, dutycycle and slewrate are set in the range from 2 to 254. Default values are 128, 128 and 2 respectively. The default is to follow the potentiometer settings.

```
followpots [yes|no]
ontime 128
dutycycle 128
slewrate 2
recoverytime 10
```

ontime is the time voltage is applied to the motor before PWM starts – typically this potentiometer is set at a mid point or 128- reducing this number lowers ontime – increasing this number raises ontime. High inductance motors typically need longer ontime settings. Duty cycle is the ratio by which voltage to the motor is turned on and off at a particular switch frequency. Reducing the duty cycle number increases the current delivered to the coils while increasing this number reduces the amount of current. The duty cycle number should be in the range 100-160. The duty cycle number should never be less than 50. Slewrate is a number that compensates for back EMF generated by the motor at fast stepping rates. Typically slewrate can be left at 2 or increased to a number in the 40-60 range. Note that motor power should be turned off when changing this parameters. Also followpots should be set to no for these parameters to overrule the potentiometer settings.

2.5 Step Mode Selection

The step mode is set by this parameter – wave, full or half step. The step mode is set either by reading the step mode header on the circuit board or by following the step mode parameter if this mode is enabled. The default setting is to follow the header jumper on the circuit board.

If the followstepheader is set to yes then the stepmode parameter is ignored and the header jumpers are followed.

```
followstepheader [yes|no]
stepmode [wave|full|half]
```

2.6 Automatic Motor Current Shut Off

Motor hold current can be automatically turned off after a period of time set by this parameter. The time frame is in milliseconds. Default setting is to turn motor current off after 8 seconds.

```
automotorshutoff [yes|no]
motorshutofftime 8000
```

2.7 Motor Torque Reduction

Motor holding torque can be set to a lower value after a time delay. Motor reduction amount is a percentage from 0 to 100. Motor reduction is activated after a time in milliseconds. The default is to reduce motor holding torque by 50% after 4 seconds.

```
motorpowerreductionmode [on|off]
motorpowerreduction 50
powerreductiontime 4000
```

2.8 Step and Direction Polarity (Rev 1.0 feature only)

The step pulse and direction polarity is set by this parameter. By default the step pulse is an active high going pulse of at least 5useconds and the direction polarity is counterclockwise high. These parameters are followed in Motor Controller software version 1.0 only.

```
steppolarity [positive|negative]
dirpolarity [highcw|highccw]
```

2.9 Limit/eStop Switch Detection

The limit and eStop switches are followed if limit switch recognition is enabled. Note that limit switch recognition has three possible states: no, yes or smart. Smart limit switch recognition allows the stepper mode controller to move the motor only in the direction away from the limit switch. By default limit switch recognition is not recognized.

```
checklimitswitch [no|yes|smart]
checkestop [no|yes]
```

2.10 I2C Address

The Motor Controller has an I2C command recognition mode that allows the G Code processor in the MC433G or another I2C Master to talk with and control the Motor Controllers directly via the I2C bus. The I2C bus is a high speed (400Kbps) party line 4 wire communications bus that ties all the processors on the MC433G and external peripherals together such as Smart Joystick, DRO's and LCD displays to communicate. The command enables or disables I2C communication recognition. By default I2C command recognition is enabled. Note that the default I2C address of each motor axis is as follows: X-10, Y-11, Z-12 and A-13 (address is hexadecimal).

```
i2cmode [on|off]
```

TWI_slaveAddress 0x10

Note: Each Motor Controller I2C address must be unique or communication contention will occur.

3.0 Programming the Motor Controllers

Use MC433prog.exe to program motor controllers EEPROM. Download the latest motor control software from the MC433 product page and re-flash the motor controllers following the procedure in the manual. Note that each axis must be selected in order to load the correct EEPROM content. Processors are selected by entering an xn command where n is 2, 3, 4 or 5. X5 is the x-axis, x4 is the y-axis, x2 is the z-axis and x3 is the a-axis. Each motor controller must have a unique I2C address in order for the G Code processor to recognize it.

To download or upload EEPROM contents use the commands “de” and “ue” respectively in MC433Prog.exe.

Upload EEPROM contents – first select the processor by typing x2, x3, x4, x5:

```
-x5
Target Atmega168 responding - default fuses set - 20MHz clock
-uex5eprom.hex
Read 512 target EEPROM bytes
Write Flash/EEPROM data to x5eprom.hex
-
```

Convert EEPROM content file to a text file by starting MC433Convert.exe and enter the file name x5eprom.hex as follows:

```
MC433 Motor Controller Configuration Compiler Utility V1.01
© Copyright 2009, SOC Robotics, Inc.

Enter input file with .hex or .txt extension -
Program will convert according to the file extension

Enter input file name: x5eprom.hex
Enter output file name: x5eprom.txt
```

The text file can be now be edited to suit the needs of that axis. When the necessary modifications have been made convert the text file back to a hex file as follows:

```
MC433 Motor Controller Configuration Compiler Utility V1.01
© Copyright 2009, SOC Robotics, Inc.

Enter input file with .hex or .txt extension -
Program will convert according to the file extension

Enter input file name: x5eprom.txt
Enter output file name: x5epromnew.hex
```

Download the new hex file using MC433Prog.exe:

```
-x5
Target Atmega168 responding - default fuses set - 20MHz clock
-dex5epromnew.hex
load file length = 72
Bytes written to target 72
Read back target to verify
Read 512 target EEPROM bytes
Target matches x5epromnew.hex
-
```

New parameters loaded into processor x5 – x-axis. Now repeat this process for the other motor controllers.

4.0 Example Configuration Files

Example configuration file for x-axis:

```
; MC433 Configuration File created by MC433Convert
; Source file: x5eeprom.hex
; File created: 15:26:54 07/24/09
;
; Product Id: 1
;
;
; Software Revision Level: 0.99
;
; Embedded Copyright Notice:  SOC MM100 V0.99 (c) 2010
;

; I2C Configuration - I2C address in hex
i2cmode on
TWI_slaveAddress 0x10

; Stepper motor parameters
followmotorparameters no
resistance 1.50
inductance 1.70
current 3.00
drive_voltage 24.00

; Calculated motor parameters:
; On time= 235.324620 usecond
; Duty Cycle= 0.235325
; Slew Rate= 0.235325

; Potentiometer set PWM parameters
followpots yes
ontime 128
dutycycle 128
slewrates 2
recoverytime 50

; Step mode header selection switch enable/disable and step mode setting
followstepheader yes
stepmode half

; Automatic Motor shut off mode and time (msec) - enable/disable
automotorshutoff yes
motorshutofftime 8000

; Reduce motor hold torque enable/disable and time (msec)
motorpowerreductionmode on
motorpowerreduction 50
powerreductiontime 4000

; Set step/dir input polarity
steppolarity positive
dirpolarity highcw

; Limit switch and eStop switch checking
checklimitswitch no
checkestop no

; Follow step/direction signal inputs or just i2c
followstepdir yes

; End of MC433 Configuration settings
```

Example Intel Hex file:

```
;10000000085c21e4f11010062534f43204d4d3130c8
:10001000302056302e3938202863292032303039ac
:1000200000000100000000c03f9a99d93f00004040f6
:1000300000000c041008d8c0032b901680000014011
:080040001f0132a00f706310d4
:00000001FF
```

Example of output generated by MC433Convert.exe

```
; MC433 Configuration File created by MC433Convert
; Source file: x-axis.hex
; File created: 14:12:09 02/19/09
;
; Motor Controller Software revision level 0.99
;
; Stepper motor parameters
followmotorparameters no
resistance 0.70
inductance 2.10
current 3.00
drive_voltage 44.00

; PWM parameters
followpots yes
ontime 128
dutycycle 128
slewrates 2
recoverytime 10

; Step mode header selection switch enable/disable and step mode setting
followstepheader yes
stepmode wave

; Automatic Motor shut off mode and time - enable/disable
automotorshutoff yes
motorshutofftime 600

; Reduce motor hold torque enable/disable
motorpowerreductionmode on
motorpowerreduction 8
powerreductiontime 4

; Set step/dir input polarity
steppolarity positive
dirpolarity highcw

; Limit switch and eStop switch checking
checklimitswitch no
checkestop no

; I2C Configuration
i2cmode on
TWI_slaveAddress 16

; End of MC433 Configuration settings
```