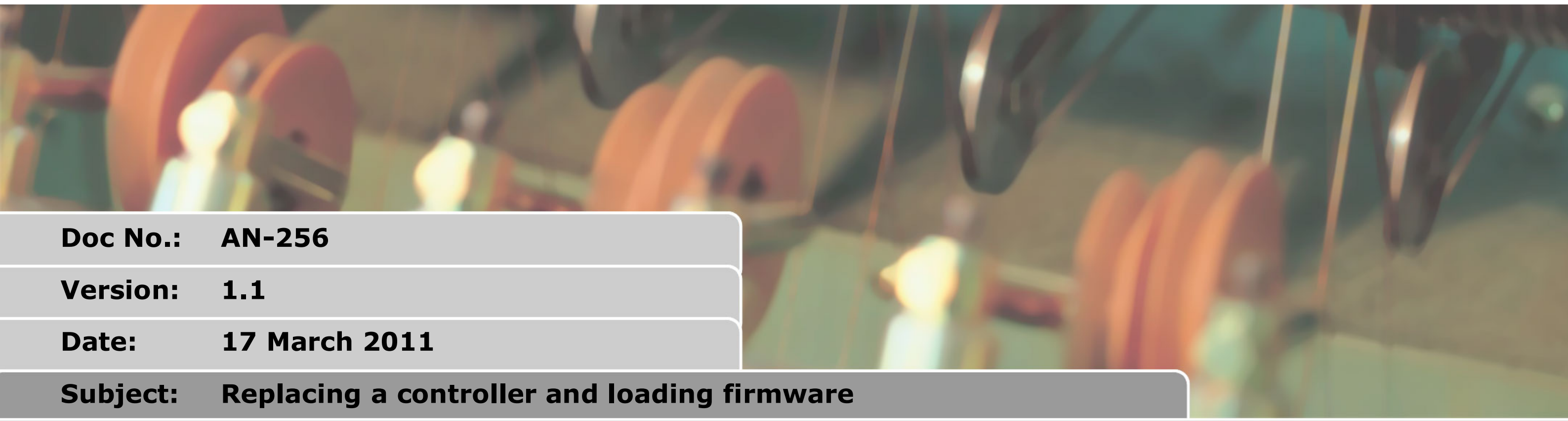
Trio Motion Technology Ltd.
Shannon Way, Tewkesbury,
Gloucestershire. GL20 8ND
United Kingdom
Tel: +44 (0)1684 292 333
Fax: +44 (0)1684 297 929

1000 Gamma Drive
Suite 206
Pittsburgh, PA 15238
United States of America
Tel: +1 412 968 9744
Fax: +1 412 968 9746

B1602 Tomson Centre
188 Zhang Yang Rd.,
Pudong New Area, Shanghai,
Postal code: 200122
CHINA
Tel/Fax: +86 21 587 97659



Doc No.:	**AN-256**
Version:	**1.1**
Date:	**17 March 2011**
Subject:	**Replacing a controller and loading firmware**

APPLICATION NOTE

1. Introduction

Replacing an existing controller can be a worrying exercise. This guide should make the operation easier by detailing all the required steps and highlighting any potential pitfalls. This document concentrates on the MC464 but the same logic applies to all Motion Coordinators.

1.1. Document History

V1.1 Added new Flash parameters to list in section 2.1.4.

Added section *5.3* about MC_CONFIG 17 March 2011.

2. Preparation

Before you need to replace a controller it is advised to make a backup of all programs and settings. This should be done after commissioning and the information should be safely stored, maybe on a memory stick or CD in the Machine control cabinet. You could even keep a SD card with all the information so that you just need to plug it in and the controller is configured and ready to go.

2.1. Information to store

2.1.1. TrioBASIC Project

This is the obvious one to save, but it is amazing how many people forget. It is best to store the entire project folder from your PC. If you do not have the original project you can save it using Motion Perfect or onto an SD card

Motion Perfect – see the help file on Save Projects

SD card – see the manual for the FILE “SAVE_PROJECT” command.

You may find that your machine builder has encrypted the project or locked the controller. In this case contact them to ensure they are keeping a backup.

2.1.2. Table Data

The table can contain recipes, profiles and much more. It is recommended to hard code any fixed values so that they are loaded from a TrioBASIC program and so stored in the project. However it is often useful to have user configurable data. This can be saved using Motion Perfect or to the SD card. The SD card is very useful in this situation as it can be left in the controller and the configuration can be save every time the data changes or on power up.

Motion Perfect – see help file for Save Table File.

SD Card – see the manual for the STICK_WRITE command.

2.1.3. VR Data

VR data is another area of memory which the user can use. Like Table it is recommended to hard code static values but you should save any user settings. Using Motion Perfect you can only save these by 'Modifying the Startup Program' which changes the project. You can also save VR data to the SD card when it changes or on power up.

Motion Perfect – see the help file for Modify Startup Program

SD Card – see the manual for the STICK_WRITEVR command

2.1.4. Flash Parameters

Most parameters are stored in RAM and need to be set in your TrioBASIC program, however there are a few that are stored in FLASH and can cause problems when replacing a controller. The following list is complete as of firmware version 2.0137. Please be aware that new parameters may have been added. If you are upgrading from a system using firmware older than 2.0137 check that the defaults of these parameters will not cause a change in your system.

These values can be checked within a TrioBASIC program and set if required. It is strongly recommended that these parameters are set in the special MC_CONFIG script as this is included in the Motion Perfect project and is therefore a more complete record of the configuration of the Motion Coordinator. (MC_CONFIG is only available in MC4xx Motion Coordinators system software V2.0110 and later.)

```
AXIS_OFFSET
CANIO_ADDRESS
CANIO_MODE
MODULE_IO_MODE
SCHEDULE_TYPE
SERVO_PERIOD
REMOTE_PROC
IP_ADDRESS
IP_NETMASK
IP_GATEWAY
```

Some Flash parameters cannot be included in the MC_CONFIG file as they are only accessible as function calls. For example:

DEVICENET function 2 – set baud rate

Ethernet Flash parameters can also be accessed using the ETHERNET command. The following values can only be set using the Motion Perfect terminal or an SD card.

ETHERNET function 0 – IP address

ETHERNET function 2 – Subnet mask

ETHERNET function 8 – Default gateway

2.1.5. Feature Enable Codes

Quite often people keep a spare controller with no enabled codes to reduce the value of their spares, you must remember though to purchase the codes when you install the replacement. You can check which codes are enables using Motion Perfect. It is a good idea to make a note of these in your project or somewhere where it will not get lost!

2.1.6. Firmware version

Firmware is regularly updated with new features and bug fixes. This can occasionally cause problems where a programs local variable has become a keyword or where execution speed varies slightly. It is advised to keep a record of the version of firmware you are using so you can always return back to it if required. This can be checked using Motion Perfect.

Motion Perfect - see the help file Controller Configuration

3. Replacing a Controller

This is simply reinstalling the data from section 2, you can either do this manually or if you have it all on a SD card you can plug it in and let it do all the hard work. The following steps will ensure your controller is configured correctly.

3.1. Loading Sequence

3.1.1. Firmware version

Check if the controller has a suitable firmware version and load a different version if required. We generally recommend using the latest version but in some instances you may need to change.

3.1.2. Feature Enable Codes

Use Motion Perfect to check the correct codes are installed.

Motion Perfect – see help file for Enable Features

3.1.3. Flash Parameters

If any parameters are not set in the program you should either set them in Motion Perfect or with the SD card.

3.1.4. VR Data

Any VR data that is not stored in a program will have to be manually configured or loaded from an SD card.

SD Card – see manual for the STICK_WRITEVR command

3.1.5. Table Data

Any Table data that is not stored in a program will have to be manually configured or loaded from an SD card.

Motion Perfect – see help file for Load Table

SD Card – see manual for the STICK_WRITE command

3.1.6. TrioBASIC project

The project will have to be loaded back onto the controller either with Motion Perfect or using the SD card.

Motion Perfect – see help file Project Load

SD Card – see manual for the FILE “LOAD_PROJECT” command

4. Loading Firmware

After loading firmware you will need to load your project, VR and Table. It is also advisable to check the Flash parameters as new ones may cause different operation.

Note: please check MODULE_IO_MODE as this is 1 by default which will cause different addressing of Trio CAN IO. See the manual entry for more detail.

5. SD Card

5.1. *Saving Table and VR data*

If you wish to save the most recent data you should write the data to the SD card somewhere during the machine cycle.

For example if you are using a Modbus HMI to configure parameters in a machine you may set VR(10) to tell the controller to store the memory. If the controller fails to write to the SD card the value 2 is returned to the HMI so an error can be displayed to the user.

Depending on TSIZE this routine can take a couple of minutes, you can always save a smaller portion of the table or VR’s, just be sure you have not missed any data.

```
success = FALSE
IF VR(10)= 1 THEN
 success = STICK_WRITE(123,0,TSIZE,0)
 success = STICK_WRITEVR(124,0,65536,0)
 IF success = FALSE then
   VR(10) = 2
 ELSE
   VR(10) = 0
 ENDIF
ENDIF
```

5.2. *Loading controller configuration*

Note: This example applies to controllers which do not use the MC_CONFIG script. If MC_CONFIG is available then it is recommended to place all the settable flash parameters in MC_CONFIG instead.

The example below is a TRIOINIT.BAS file that will load all the correct data onto a new controller. You may not need to load all of the parameters.

```
'Check firmware version and load 2.0084
IF VERSION<>2.0084 THEN FILE “LOAD_SYSTEM” “MC464_2.0084”

'Set IP Address
ETHERNET(1,-1,0,192,168,0,250)

'Set Subnet Mask
ETHERNET(1,-1, 2,255,255,255,0)

'Set Default Gateway
ETHERNET(1,-1,8,192,168,0,225)
```

```
'Set Flash Parameters
AXIS_OFFSET = 0
CANIO_ADDRESS = 32
MODULE_IO_MODE = 0
SCHEDULE_TYPE = 0
SERVO_PERIOD = 1000
REMOTE_PROC = -1

'Load VR and Table
STICK_READ (123, 0)
STICK_READVR (124, 0)

‘Load Project
FILE “LOAD_PROJECT” “MyProject”

PRINT “Completed loading controller setup”
PRINT “Connect to Motion Perfect and check Feature Enable Codes”
PRINT “Then power cycle controller to start”
```

5.3. *Using MC_CONFIG*

The example below is a MC_CONFIG script that will set all the Flash parameters to their default values. Create MC_CONFIG with Motion Perfect by selecting “Program->New” then name the program MC_CONFIG. MC_CONFIG will run automatically on power up, there is no need to set it to autorun.

```
AXIS_OFFSET SLOT(-1)=0
AXIS_OFFSET SLOT(0)=0
' other axis_offsets as required
CANIO_ADDRESS=32
CANIO_MODE=0
MODULE_IO_MODE=1
SCHEDULE_TYPE=0
SERVO_PERIOD=1000
REMOTE_PROC=-1
```

Optionally these parameters can also be set:

```
IP_ADDRESS=192.168.0.250
IP_NETMASK=255.255.255.0
IP_GATEWAY=192.168.0.225
AUTO_ETHERCAT=ON  ' non-flash parameter
```

6. Common problems

6.1. *I don’t have enough axes*

Have you enabled the correct feature codes

6.2. *My axes are in the wrong order*

Is AXIS_OFFSET the correct value

6.3. *My CAN IO is offset by 8, 16 etc*

The MODULE_IO_MODE is 1 by default, this includes any module I/O in the IO sequence. Your project was probably written before this was implemented and so you need to set it to OFF.

6.4. My programs do not run

Check that the programs compile

Check that a program is set to autorun

6.5. My programs do not compile

Maybe a firmware change has caused an error in your project, check the error and either adjust the project or load the version of firmware that the machine was commissioned with.