

T E C H N I C A L N O T E

Doc No.: TN30-19
Date: July 2004
Version: 3.6
Subject: MC302-K Setup information with ServoStar S300 / S600 / S700 Amplifier

The Trio MC302-K *Motion Coordinator*[™] is designed to plug into the expansion slot on the Kollmorgen ServoStar S300 / S600 / S700 drives, and provide full programmable positioning functions using standard Trio BASIC commands. The MC302-K communicates with the drive through a Dual Port RAM (DPR) system, allowing full read/write functionality of any valid drive parameter.

Please follow this Note carefully noting that some sections/setup are REQUIRED and some are OPTIONAL.

Quick Check List

The following is a quick list of required items and procedures to set up the S300, S600, and S700 for use with the MC302-K *Motion Coordinator*[™].

1. MC302-K *Motion Coordinator* plug-in board with firmware **v1.9300** or higher.
2. Trio software **Motion Perfect 2**, version **2.2.1.12** or higher (available from our web site www.triomotion.com).
3. Kollmorgen Servostar 300 Drive with firmware **v2.00** or higher.
4. Kollmorgen Servostar 600 Drive with firmware **v5.99** or higher (use v6.48 if using registration features).
5. Kollmorgen Servostar 700 Drive with firmware **v3.05** or higher.
6. Kollmorgen Servostar 300 / 600 / 700 Setup Software [Drive.Exe], and necessary null-modem serial cable.
7. Trio programming cable (PC to MC302-K) part number **P350**.
8. Select **High-Speed serial port** parameters in *Motion Perfect 2* for use with MC302-K (See Section 2.0).



Note

This Tech Note assumes a standard Kollmorgen Servostar S300/S600/S700 drive using resolver feedback on X2. Other special feedback devices and drive operating modes may require additional drive parameter settings not discussed here (e.g. Wake-and-Shake). Such feedback devices include, EnDat, SSI, and Sine-Cosine. Consult Trio for additional information.

1.0 Setting Up the ServoStar Amplifier (REQUIRED)

The standard Kollmorgen GUI interface software should be used to set the S300/S600/S700 drive for use with your specific motor prior to installing the MC302-K.

It is recommended to reset all S300/S600/S700 drive parameters to default values. This can be done from the drive's GUI software (Fig 1). To clear all drive parameters in the drive's GUI, press the *Clear EEPROM* button then *Save to EEPROM* button found on the toolbar. Cycle power on the drive.

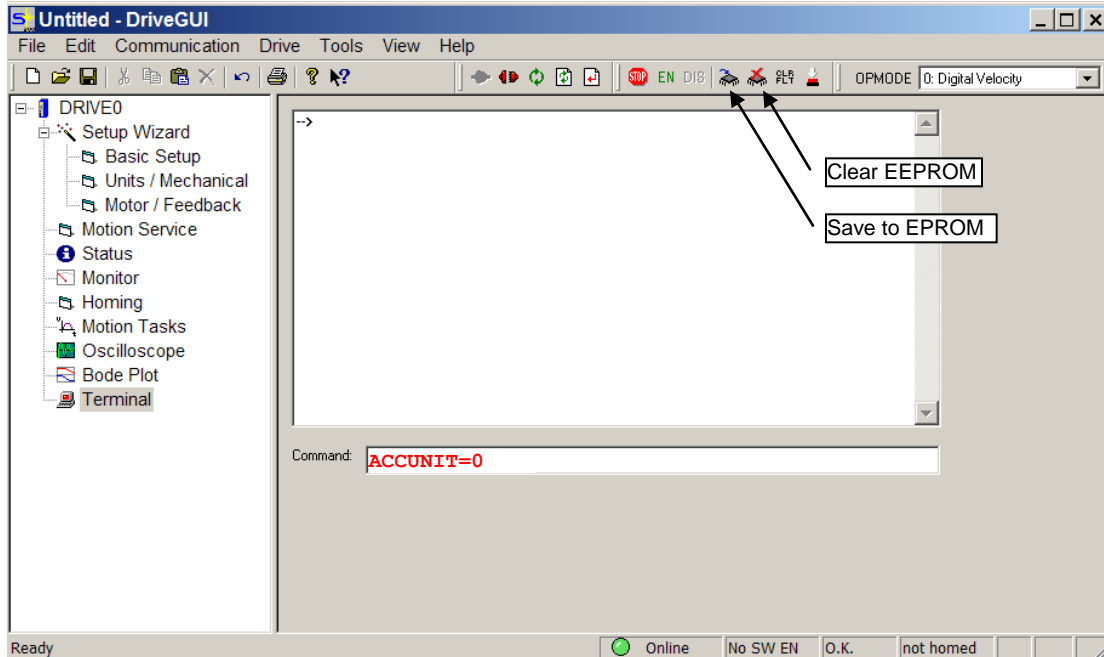


Fig 1 - Drive GUI Terminal Command Line

The following drive parameters in TABLE 1 are setup using the S300/S600/S700 GUI Software [Drive.Exe] without the Trio MC302-K installed. This ensures proper saving to the drive's flash memory since most parameters require a drive power cycle and/or reset to take effect.

S300 / S600 / S700 Drive Setup Procedure (Before installing the MC302-K)

1. Reset all S300/S600/S700 parameters to default
2. Configure the S300/S600/S700 for your specific motor
3. Set the parameters as shown in TABLE 1 and save to EPROM (in S300/S600/S700 Setup Software). You can use the **Parameter Name** directly in the drive's Terminal screen (see above Screen 1).
4. Tune the S300/S600/S700 in **Digital Velocity** mode (OPMODE=0) by setting the appropriate gain parameters
5. Power down the S300/S600/S700 and install MC302-K Motion Coordinator

TABLE 1 – One-time S300 / S600 / S700 Initialization Parameters (REQUIRED)

Parameter Name	Parameter No.	Setting	Function	Notes
ACCUNIT	345	0	Sets Accel/Decel units in the drive to “ms”	Allows proper ramp control of the drive when using the MC302-K
ACC	1	1	Sets Accel ramp for the drive’s velocity controller	This setting must be small to allow fast command changes from the MC302-K.
AENA	6	0	Software auto-enable	Allows drive enable to be controlled by Trio BASIC command WDOG=ON
BUSP2	402	3 (S300/700 only)	Position loop synchronization	This parameter effects positioning repeatability and <u>must</u> be set.
DEC	34	1	Sets Decel ramp for the drive’s velocity controller	This setting must be small to allow fast command changes from the MC302-K.
ENCMODE	52	0	Drive enc emulation	Encoder Equivalent Output; 0 = OFF
EXTMUL	56	1 (S600 only)	External encoder multiplier	Multiplies the encoder counts on the Drive’s X5 input.
EXTPOS	57	3 (S600) 0 (S300/700) -3 (S300/700) for X5 gearing	Drive position feedback + control type	Required for external encoder (X5) operation. Note: Normal operation EXTPOS=0. If gearing on X5, be sure to connect the encoder BEFORE setting to “-3”. Otherwise an F10 drive error will result.
FILTMODE	60	1	Luenberger Observer Filter Mode	1 = Disable, 4kHz update 3 = Enable, 4kHz update
FPGA	315	3 (S600 only)	Drive FPGA functionality	Required for proper bus synch over DPR.
GEARMODE	63	0 3 6	Electronic gearing mode	Required for external encoder (X5) operation with MC302-K (master-slave). 0 = No gearing on X5 3 = Gearing on X5 6 = EnDat feedback, S600 Note: If not using an encoder on the S300/700 drive, GEARMODE <u>must</u> be set to 0.
O1MODE	175	23	Function of dig output 1	Allows drive’s output control via MC302-K
O2MODE	178	23	Function of dig output 2	Allows drive’s output control via MC302-K
OPMODE	180	0	Drive operating mode	Set to Digital Velocity mode in drive
PGEARI	202	65536	Sets number of counts sent to the MC302-K (resolution of the system) for position feedback.	Yields 65536 counts in the MC302-K. Default PGEARO=1048576 in S600; 1 in S300/700. These should not be changed. Use the UNITS parameter in the MC302-K for setting users units (i.e counts/unit).

TABLE 1 (cont) – One-time S600 / 300 Initialization Parameters (REQUIRED)

PMODE	205	1	Line phase error mode	1 = 3-phase input power 0 = 1-phase input power
PRBASE	209	16	Sets internal resolver resolution	Provides 65536 counts per motor rev to the MC302-K. Use the UNITS parameter in the MC302-K for setting users units (i.e counts/unit).
SYNCSRC	387	6 (S300/700 only)	Position loop synchronization	This parameter defines the source for the synchronization of the control loops to the external fieldbus.
SERCSET	401	0 or 2	Drive DPR (Dual Port RAM) mapping to MC302-K	For MC302-K firmware < v1.91, SERCSET = 0 For MC302-K firmware ≥ v1.91, SERCSET = 2 (set bit 1)
PTBASE	213	1 (S600) 4 (S300/700)	Interrupt interval of MC302-K to drive	1 = 0.25ms 4 = 1.0ms (You <u>must</u> set the SERVO_PERIOD setting in MC302-K, see Section 3.0)
The following Parameters are required ONLY for Registration				
DRVCNFG2	484	6		Required for registration capture mode.
EXTLATCH	385	1	Defines the source for the position information using the Latch functions.	1= Capture External Encoder (DIGITAL IN1), and Resolver Feedback on DIGITAL IN2.
IN1MODE	98	26	Configures the function of DIGITAL IN1 on the drive.	An edge on this input latches the <u>X5 encoder position</u> to Trio BASIC parameter REG_POS AXIS(1). The min. cycle time for a low/high to high/low transaction is 500µs. The min. time between two latch pulses is 8msec. Note: Drive power must be cycled.
IN2MODE	101	26	Configures the function of DIGITAL IN2 on the drive.	An edge on this input latches the <u>Motor position</u> to Trio BASIC parameter REG_POS AXIS(0). The min. cycle time for a low/high to high/low transaction is 500µs. The min. time between two latch pulses is 8msec. Note: Drive power must be cycled.

Note: In the S300/S600/S700 if changing **FBTYPE** from resolver (FBTYPE=0) to any other feedback mode, recheck ACC and DEC. ACC and DEC set the acceleration and deceleration ramp times, and must remain the default values. They are scaled to the drive's ACCUNIT parameter.

FBTYPE is by default set for a resolver based motor and should not be changed. FBTYPE is changed for use in other feedback modes. This would include using a Hall sensor motor or the drive's "wake and shake" sensorless feedback mode. If FBTYPE needs to be changed for such modes, be sure ACC and DEC are set to their minimum values.

TABLE 2 – S300 / S600 / S700 User Parameters (OPTIONAL)

Parameter Name	Parameter No.	Setting	Function	Notes
ANOFF1	11	10000 to -10000	Analog offset for SW1 input (Range = -10000 to +10000)	Allows offset voltage setting for Analog input 1 (SW1).
ANOFF2	12	10000 to -10000	Analog offset for SW2 input (Range = -10000 to +10000)	Allows offset voltage setting for Analog input 2 (SW2).
ANZERO1	15	1	Zeros out voltage on Analog Input 1.	Starts automatic offset correction for Analog input 1 (see drive Variable def manual).
ANZERO2	16	1	Zeros out voltage on Analog Input 2.	Starts automatic offset correction for Analog input 1 (see drive Variable def manual).
ANOUT1	13	0 to 8	Source of MONITOR1 analog output.	Variable source for MONITOR1 analog output (see drive Variable def manual).
ANOUT2	14	0 to 8	Source of MONITOR2 analog output.	Variable source for MONITOR2 analog output (see drive Variable def manual).
AN1TRIG	417	mV	Sets MONITOR1 analog output to a voltage.	To set an analog output voltage in mV using AN1TRIG, ANOUT1 <u>must</u> be set to 8.
AN2TRIG	418	mV	Sets MONITOR2 analog output to a voltage.	To set an analog output voltage in mV using AN2TRIG, ANOUT2 <u>must</u> be set to 8.
DIR	42	20 or 21	Changes motor direction	Can be used to switch the motor's direction polarity.
IPEAK	110	0.0 to DIPEAK	Sets the drive's output peak current.	Can be used to limit output current for torque appl. IMAX (param 95) is default value.
MONITOR1	154	none	Voltage output (Range = -10000 to +10000)	Drive's analog output at MONITOR1. Read only (S600 only).
MONITOR2	155	none	Voltage output (Range = -10000 to +10000)	Drive's analog output at MONITOR2. Read only (S600 only).
VBUS	282	Read only	Returns the bus voltage in the drive	Can be used to check for bus voltage before applying a WDOG=ON enable command.

* Note: Additional S300/S600/S700 parameters may be needed for your specific setup. Please consult the Kollmorgen "VarCom Reference Guide" document for complete parameter descriptions.

2.0 Trio's *Motion Perfect2* Setup to Connect to MC302-K (REQUIRED)

You will need a Trio serial cable (part number P350) to communicate from the PC to the MC302-K controller. Before connecting to the MC302-K, install *Motion Perfect2* on your PC. The latest can be downloaded from www.triomotion.com.

To set the MC302-K serial comms:

1. Connect the Trio P350 Serial cable from the PC to the MC302-K and start *Motion Perfect2*, Fig 2.
2. Press the **Cancel** button.
3. Go to the **OPTIONS \ COMMUNICATIONS** menu as shown in Fig 3.
4. Uncheck "Packet Comms" if checked, Fig3.
5. Press the **Edit** button, then press the **Fast (38400,8,e,1)** button to select the higher baud rate setting for COM1 as shown in Fig 4.
6. Press the **Connect** button on the Tool Bar to establish comms link.

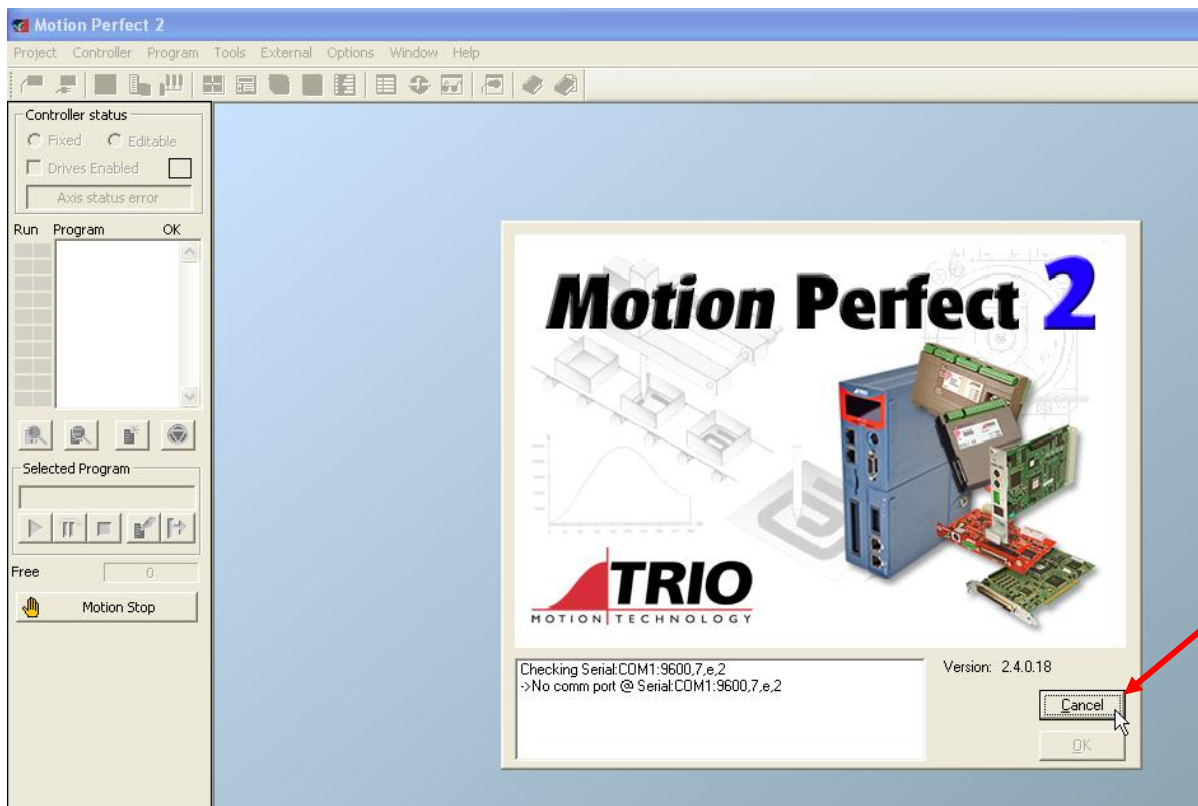
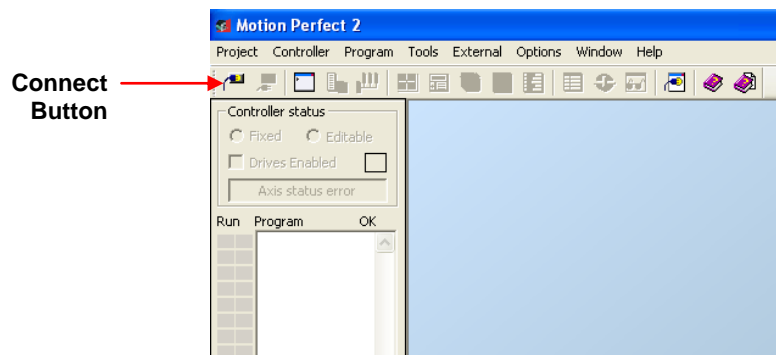


Fig 2 – *Motion Perfect2* Boot Screen – Cancel to change comms settings for MC302-K

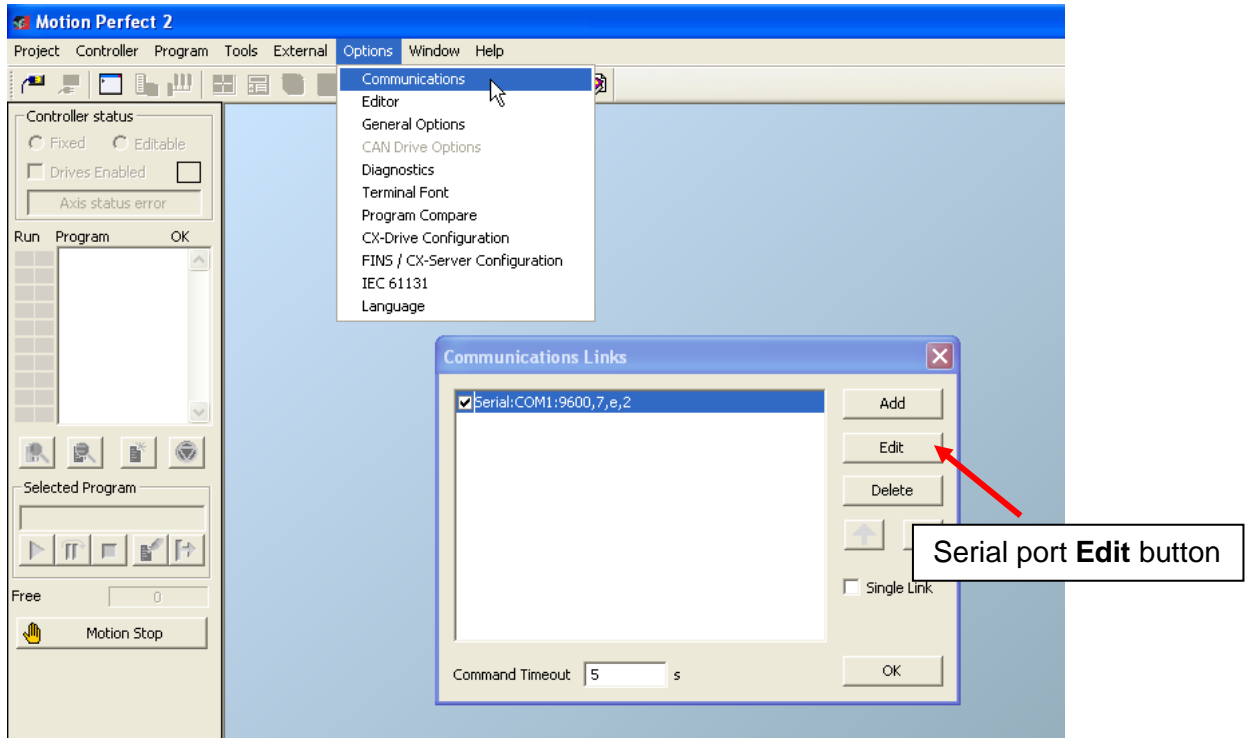


Fig 3 – Communication Options Screen – Edit to change port settings

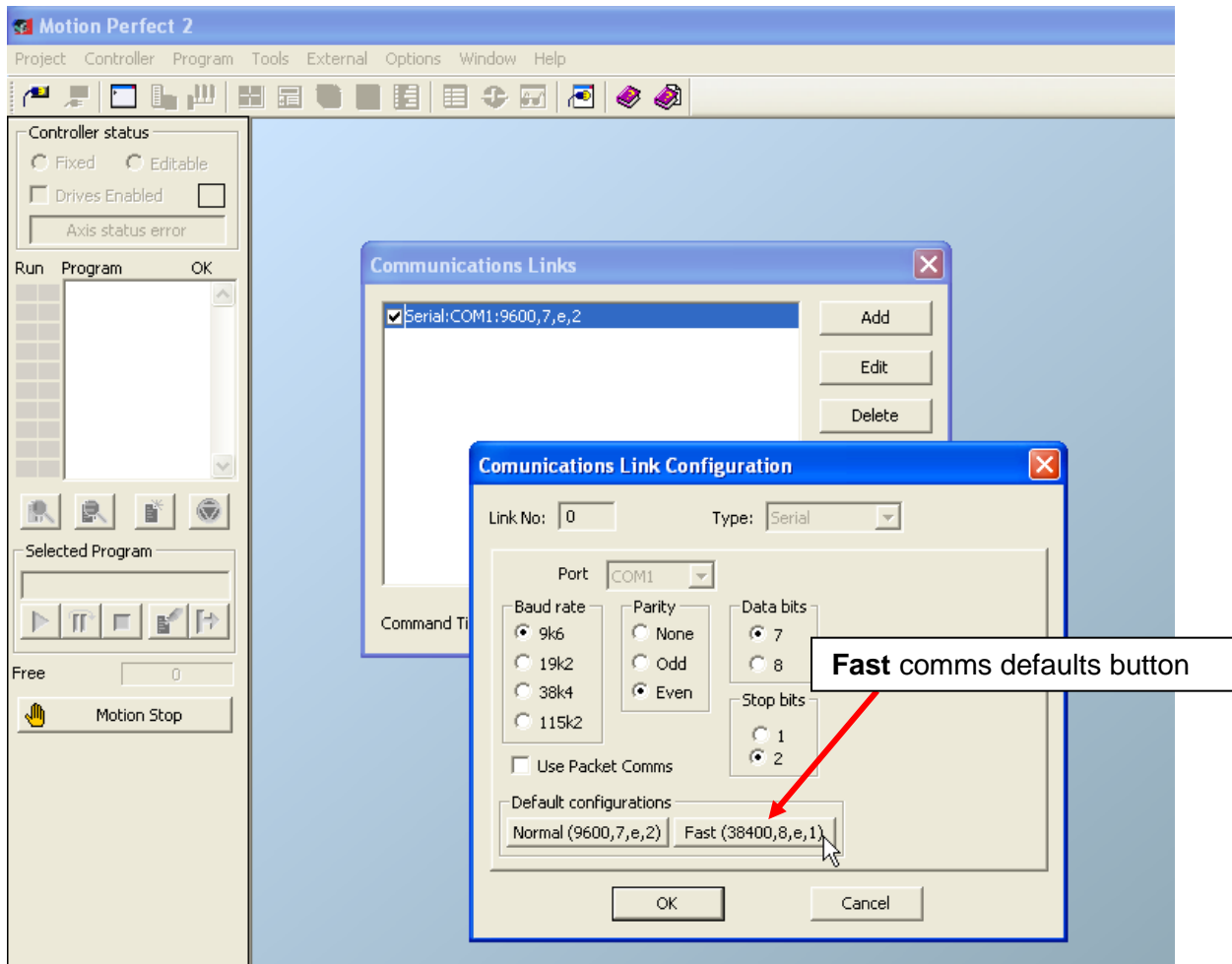


Fig 4 – Setting High Speed Comms for the MC302-K

3.0 Setting Up the Trio MC302-K MotionCoordinator (REQUIRED)

For proper operation of the MC302-K with the S300 / 600 the Trio parameter SERVO_PERIOD must be set correctly. The SERVO_PERIOD parameter is only set once at the MC302-K Command prompt ">>" (do not enter in a program). The SERVO_PERIOD takes effect and is saved automatically on the next power cycle. The default is SERVO_PERIOD=1000 for use with the S300/S700.

For S300/S700 operation, continue with the next Section.

For S600 operation, you must follow the steps below.

To set the SERVO_PERIOD:

1. Start *Motion Perfect2* and setup comms as described in Section 2.
2. Open the Channel 0 Terminal window as shown in Fig. 5.
3. At the command prompt ">>" type in the correct servo period.
MC302-K with S300/S700 Operation set SERVO_PERIOD=1000 (default)
MC302-K with S600 Operation set SERVO_PERIOD=250
4. Cycle power on the ServoStar, and watch the boot-up message in the terminal window.
5. A properly setup system will look like Fig. 7. If the SERVO_PERIOD and/or PTBASE parameters are not set correctly, the boot-up message will look like Fig. 8. Recheck both and cycle power again.
6. After power-up, verify SERVO_PERIOD at the Command prompt ">>" in *Motion Perfect2*, Fig. 6.

As with all Trio controllers, the Command Line interface ">>" can be a useful tool for debugging a system as shown in Fig 5. Drive parameters can be read and written to using the DRIVE_ commands listed in Section 5.0. Other Trio BASIC command can be issued here such as FORWARD, CANCEL, MOVE(x). Visit www.triomotion.com for the complete manual of commands.

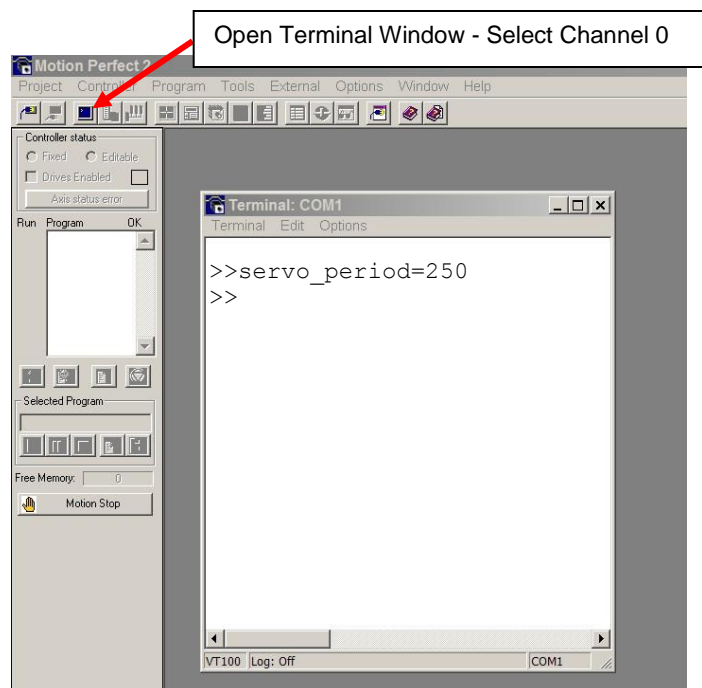


Fig 5 – Setting MC302-K SERVO_PERIOD

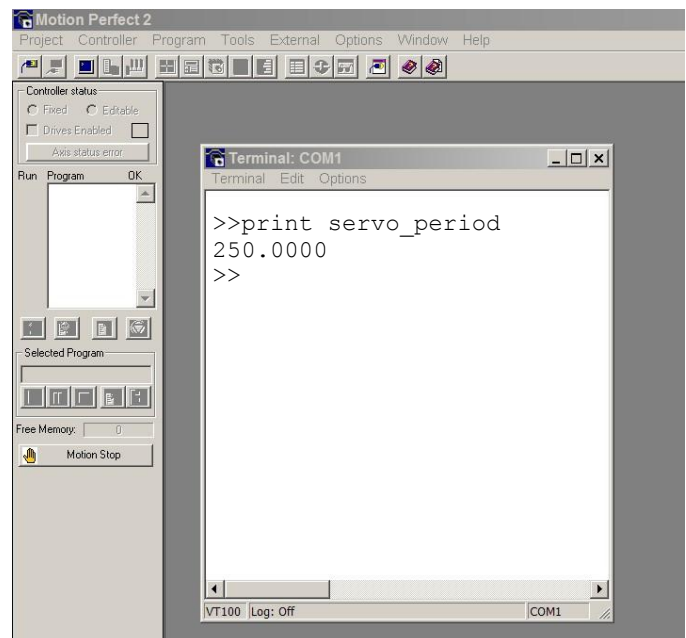


Fig 6 – Verifying SERVO_PERIOD after power cycle

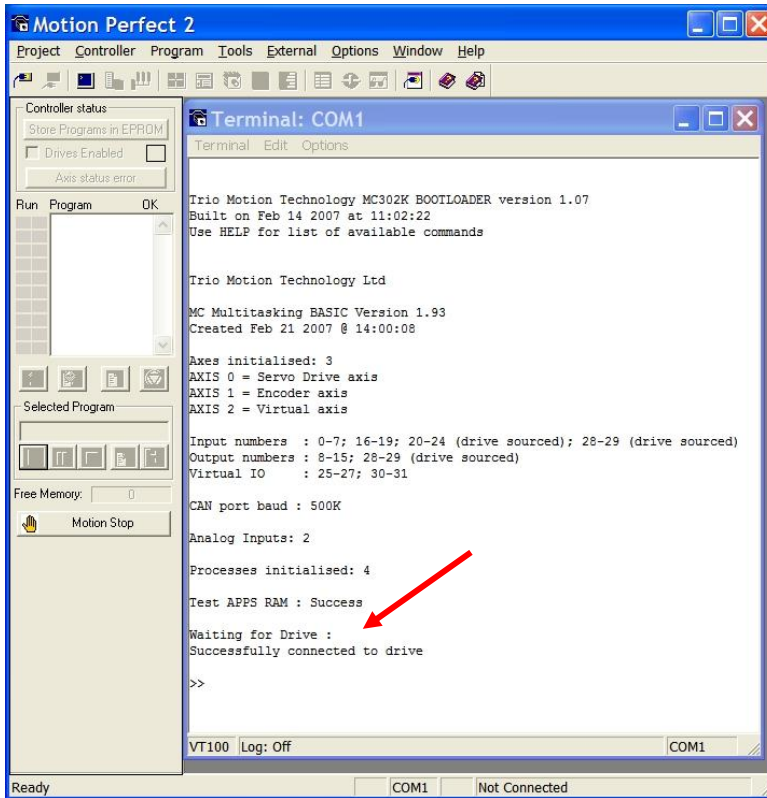


Fig 7 – Successfully configured system boot-up message

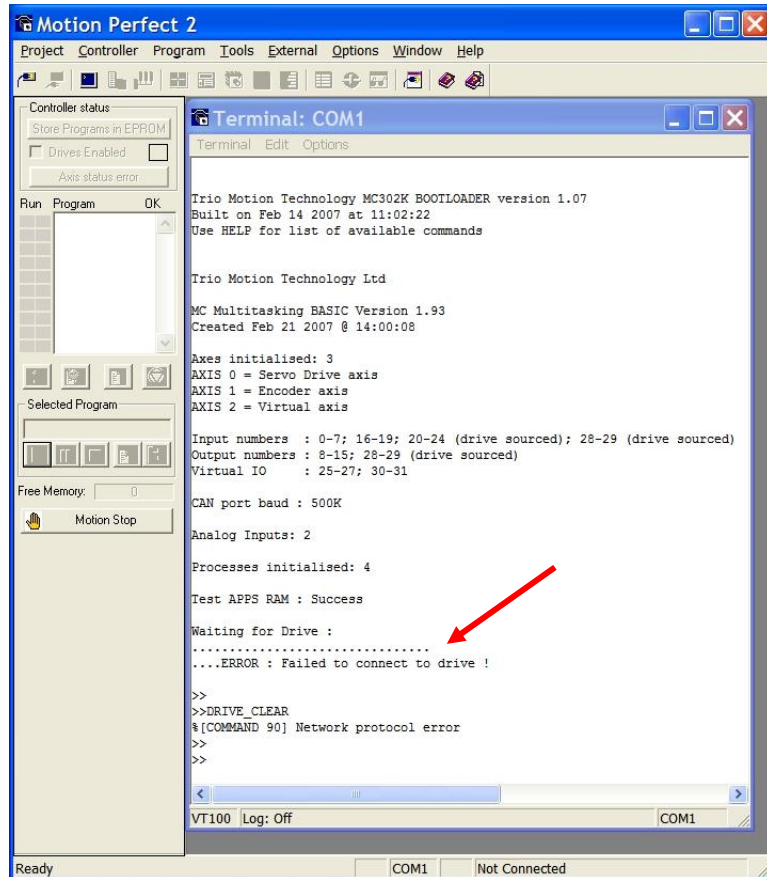
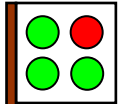


Fig 8 – Unsuccessful configured system boot-up message

Once the S300/S600/S700 drive has been set up correctly, any valid Trio BASIC command can be executed at the Command Line in *Motion Perfect2*. Use the sequence below to verify system setup.

```
>> SERVO=1
>> WDOG=1
>> SPEED AXIS (0)=5
>> FORWARD
>> CANCEL
```

4.0 MC302-K LEDs'



Top Left (green) – OK

The OK LED should be on all the time indicating the processor is healthy. The STATUS LED follows the drive enable (WDOG) function. ON=drive disabled, OFF=drive enabled. The STATUS LED also flashes when a motion following error occurs. DATUM(0) from within a Trio BASIC program will clear the motion error and LED flashing.

Bottom Left (green) – IN(0)

The IN(0) LED indicates the status of Input(0) on the MC302-K. ON indicates 24Vdc is present on Input(0); X12 connector. See the Trio BASIC “DISPLAY” command to change.

Bottom Right (green) – IN(1)

The IN(1) LED indicates the status of Input(1) on the MC302-K. ON indicates 24Vdc is present on Input(1); X12 connector. See the Trio BASIC “DISPLAY” command to change.

5.0 Trio BASIC Commands to the Drive (OPTIONAL)

Trio MC302-K uses the standard Trio BASIC command set for user programs. The full command set can be found in the Trio Technical Reference manual on the web site www.triomotion.com. In addition to the standard command set, the MC302-K uses drive-specific commands to allow reading and writing of drive parameters, and other functions from within a Trio BASIC application program.

MC302-K Drive Commands

DRIVE_READ (register, time, VR(x))

Purpose: To read a drive parameter for use in a Trio BASIC program.

A successful command will return a TRUE (-1), or otherwise FALSE (0) to the BASIC program.

register = Drive parameter number (see drive parameter list for complete listing).

The command returns the parameter value if it can be read from the drive.

time = Time in ms for the MC302-K to wait for a read request reply (default should be 100). Some drive parameters may take several hundred ms to return a value. Increase the *time* parameter to maximum the Trio BASIC should wait for a reply. If the *time* is too short, a FALSE (0) will be returned to the BASIC program. If a FALSE is returned, the *time* value should be lengthen to wait for a drive response.

VR(x) = This is the VR variable location in the MC302-K that holds the returned value from the drive. In Trio BASIC, a VR variable is global and accessible by all programs. The drive parameter value is stored in *VR(x)* . “X” is a value from 0 to 1023 specifying the location.

Example 1: Read back the actual drive current (I) parameter 88 in a program, and store the value in VR(55):

```
IF DRIVE_READ(88,100,55)=TRUE THEN PRINT "Drive Current=",VR(55)
```

Example 2: Generally, a single read attempt will return most drive parameters. For more critical drive parameters, or continuous reads, a multiple read loop and test can be implemented.

```
WHILE TRUE
  FOR tries= 1 to 10
    IF DRIVE_READ(88,100,55)=TRUE THEN
      PRINT"Drive Current=",VR(55)
      WA(1000)  `Get Drive current every 1sec
    ELSE
      PRINT"Drive did not reply"
    ENDIF
  NEXT tries

WEND  `End main loop
```

DRIVE_CLEAR

Purpose: To reset and clear S600 / 300 drive faults (warm drive boot).
A successful command will return a TRUE (-1), or otherwise FALSE (0) to the BASIC program.

When the drive is in a fault condition, DRIVE_CLEAR will run the drive's own error reset procedure. This is the same as the drive's CLRFAULT parameter (24). The DRIVE_CLEAR command will wait up to 2000ms for a drive response, and will return either a TRUE (-1) or FALSE (0) to the BASIC program.

The DRIVE_CLEAR command should only be used in fault recovery routines in a Trio BASIC program. The example below can be used to check for and clear an FE (Following Error) fault during runtime:

```
WHILE TRUE  `Error Handler Program
  .
  IF AXISSTATUS=256 OR AXISSTATUS=258 THEN  `Check for FE fault
    GOSUB reset_routine
    PRINT #5,"All Clear..."
  ENDIF
WEND  `End program loop

Reset_routine:
  DATUM(0)  `Clear FE fault in MC302-K
  DRIVE_CLEAR  `Reset drive faults
  IF DRIVE_CLEAR=TRUE THEN
    WA(1000)  `Wait in ms
    WDOG=OFF  `Cycle enable (WDOG) to the drive...
    WA(50)
    WDOG=ON  `Enable drive
    WA(50)
    SERVO=ON  `Close position loop in MC302-K
  ELSE
    GOTO reset_routine  `Try again..
  ENDIF
RETURN
```

DRIVE_EEPROM

Purpose: Performs a "SAVE" function within the drive to its EEPROM.

A successful command will return a TRUE (-1), or otherwise FALSE (0) to the BASIC program.

The DRIVE_EEPROM can be used in a Trio BASIC program to save drive parameters to the drive's flash memory. This is the same command as the drive's SAVE function (Save to EEPROM button in the Drive GUI). Note some drive parameters require a power cycle in the drive for activation.

DRIVE_RESET (0)

Purpose: Reset the communications link between the MC302-K and the drive.

The DRIVE_RESET is typically not required for normal operation, but can be used in fault recovery routines to re-establish communications between the MC302-K and the drive.

A successful command will return a TRUE (-1), or otherwise FALSE (0) to the BASIC program.

The MC302-K and drive run through an initialization process following power-up during which they move through communication phases until phase 6 (normal operation) is reached. It will not be possible to control the drive-motor if they fail to reach this final phase. The communication phase process is completely transparent to the user and occurs at initial power up. The final phase will always be reached unless there is hardware/software issue between the Drive and MC302-K Controller, resulting in a "Network Protocol Error" (BASIC_ERROR "90").

The DRIVE_RESET(0) command will reset the communications link back to "0" working up to phase "6".

DRIVE_HOME

Purpose: Performs the drive's internal home sequence.

A successful command will return a TRUE (-1), or otherwise FALSE (0) to the BASIC program.

When the DRIVE_HOME is encountered in a Trio BASIC program, the drive will begin its internal homing sequence. The mode of homing will be based on the settings of the drive's DREF, NREF, VREF, IN1MODE, and REFMODE parameters. The Trio BASIC program will hold on the DRIVE_HOME line until the drive completes the homing sequence (when the drive's "Motion Task Active" bit is cleared).

DRIVE_WRITE(register, value [,time])

Purpose: To write a value to a drive parameter from a Trio BASIC program.

A successful command will return a TRUE (-1), or otherwise FALSE (0) to the BASIC program.

`register` = Drive parameter number.

`value` = The number or expression to be written to the drive parameter.

`[time]` = Optional time in ms for the MC302-K to wait for write request reply [Default=100].

Generally, the default time delay of 100ms is adequate when writing most drive parameters. If a FALSE (0) is returned from a DRIVE_WRITE command, the `<time>` option should be used to lengthen the time the MC302-K will wait for a drive response. For example, with `<time>` set to 3000 the DRIVE_WRITE command will wait "up to" 3 seconds for a response.

Example 1: Set the drive's peak current value using IPEAK drive parameter (110), where "n" is the value of current to be set.

```
DRIVE_WRITE(110,n)
```

DRIVE_INTERFACE

Purpose: Low-level communications link between the MC302-K and the drive.

The DRIVE_INTERFACE is typically not needed, but provides useful direct access to the DPR in the drive regardless of communication status between the MC302-K and the drive. Even catastrophic drive errors such as F32 "System Error" can be read back using function mode 5, letting a Trio BASIC program determine the drive's status.

DRIVE_INTERFACE(5,ERRCODE_Byte)

Returns either the Most Significant Word (MSW) when ERRCODE_byte=0; and the Least Significant Word (LSW) when ERRCODE_byte=1. This is the 32-bit value ERRCODE that is provided by the drive, with 1 bit per fault raised by the drive. A 0 indicates that the fault is not present and a 1 indicates that it is. Bit 0 indicates the status of F01 and bit 31 indicates the status of F32. For example, if faults F29 and F04 are present then DRIVE_INTERFACE(5,0) would return 4096 (or hex 1000) and DRIVE_INTERFACE(5,1) would return 8.

Drive Error bit mapping would be as follows:

F1 to F16 - Byte 1 – (LSW)

Example: With fault F4 set in the drive, the DRIVE_INTERFACE(5,1) will return 8.

Bit#	Bit set	Value Returned
0	0	1
1	0	2
2	0	4
3	1	8
4	0	16
5	0	32
6	0	64
7	0	128
8	0	256
9	0	512
10	0	1024
11	0	2048
12	0	4096
13	0	8192
14	0	16384
15	0	32768

F17 to F31 - Byte 0 – (MSW) "F29" example

Example: With fault F29 set in the drive, the DRIVE_INTERFACE(5,0) will return 4096.

Bit#	Bit set	Value Returned
16	0	1
17	0	2
18	0	4
19	0	8
20	0	16
21	0	32
22	0	64
23	0	128
24	0	256
25	0	512
26	0	1024
27	0	2048
28	1	4096
29	0	8192
30	0	16384
31	0	32768

6.0 Input and Output Mapping

The following table lists the MC302-K and drive inputs and outputs, their location and syntax to use within a Trio BASIC program.

TABLE 3 – MC302-K and Drive I/O Mapping

Hardware	I/O point	Function	Connector Location	Trio BASIC Syntax (x = 1 or 0)
MC302-K	0	24Vdc INPUT	DB25 – Pin 25	IN(0)
MC302-K	1	24Vdc INPUT	DB25 – Pin 24	IN(1)
MC302-K	2	24Vdc INPUT	DB25 – Pin 23	IN(2)
MC302-K	3	24Vdc INPUT	DB25 – Pin 22	IN(3)
MC302-K	4	24Vdc INPUT	DB25 – Pin 21	IN(4)
MC302-K	5	24Vdc INPUT	DB25 – Pin 20	IN(5)
MC302-K	6	24Vdc INPUT	DB25 – Pin 19	IN(6)
MC302-K	7	24Vdc INPUT	DB25 – Pin 18	IN(7)
MC302-K	8	24Vdc INPUT	DB25 – Pin 17	IN(16)
MC302-K	9	24Vdc INPUT	DB25 – Pin 16	IN(17)
MC302-K	10	24Vdc INPUT	DB25 – Pin 15	IN(18)
MC302-K	11	24Vdc INPUT	DB25 – Pin 14	IN(19)
MC302-K	8	24Vdc OUTPUT	DB25 – Pin 3	OP(8,x)
MC302-K	9	24Vdc OUTPUT	DB25 – Pin 4	OP(9,x)
MC302-K	10	24Vdc OUTPUT	DB25 – Pin 5	OP(10,x)
MC302-K	11	24Vdc OUTPUT	DB25 – Pin 6	OP(11,x)
MC302-K	12	24Vdc OUTPUT	DB25 – Pin 8	OP(12,x)
MC302-K	13	24Vdc OUTPUT	DB25 – Pin 9	OP(13,x)
MC302-K	14	24Vdc OUTPUT	DB25 – Pin 10	OP(14,x)
MC302-K	15	24Vdc OUTPUT	DB25 – Pin 11	OP(15,x)
S600 [S300] S700	SW1 Analog-In 1	Analog INPUT 1	X3 – Pins 4, 5 [3, 4] X3B – Pins 9, 10	AIN(0)
S600 [S300] S700	SW2 Analog-In 2	Analog INPUT 2	X3 – Pins 6, 7 [5, 6] X3B – Pins 11, 12	AIN(1)
S600	MONITOR1	Analog OUTPUT 1	X3 – Pins 8, 10	DRIVE_WRITE(154,value)
S600	MONITOR2	Analog OUTPUT 2	X3 – Pins 9, 10	DRIVE_WRITE(155,value)
S600 [S300] S700	DIGITAL IN1	24Vdc INPUT	X3 – Pin 11 [8] X3A – Pin 2	IN(20)
S600 [S300] S700	DIGITAL IN2	24Vdc INPUT	X3 – Pin 12 [9] X3A – Pin 3	IN(21)
S600 [S300] S700	PSTOP DIGITAL IN3	24Vdc INPUT	X3 – Pin 13 [10] X3A – Pin 4	IN(22)
S600 [S300] S700	NSTOP DIGITAL IN4	24Vdc INPUT	X3 – Pin 14 [11] X3A – Pin 5	IN(23)
S600 [S300] S700	DIGITAL OUT1	24Vdc OUTPUT	X3 – Pin 16 [13] X3A – Pin 6	OP(28,x)
S600 [S300] S700	DIGITAL OUT2	24Vdc OUTPUT	X3 – Pin 17 [14] X3A – Pin 7	OP(29,x)
S600 [S300] S700	ENABLE	24Vdc INPUT	X3 – Pin 15 [12] X3A – Pin 1	AND function with WDOG=ON. The ENABLE is also IN(24).

The MC302-K I/O can be expanded using the standard Trio CANbus modules (part number P316 and P325).
Note: Connector Location pin numbers in [...] are for the S300 drive.

Example MC302-K Test Program:

```
'MC302-K ServoStar 300/600/700 Quick Test Program

DATUM(0) 'Clear any Following Errors.
WA(100)
SERVO=ON      'Turn on position loop in MC302-K
WDOG=OFF      'Cycle Drive Enable..
WA(100)
WDOG=ON

UNITS=65536 'default unit of revs
SPEED=10 'revs per sec
ACCEL=100
DECEL=100

BASE(0)
` Note position loop gains are scaled based on motor cts/rev (see PGEARI).
` For PGEARI=65536, use the gains below.
` For PGEARI=1048576 divide ALL gain values by 10.
`
P_GAIN=0.5 'Position loop gains in MC302-K.
I_GAIN=0.0
D_GAIN=0.5
VFF_GAIN=32
FE_LIMIT=0.5 'revs of motor
DEFPOS(0) 'reset position to "0"

start:
  FOR x=1 TO 10
    MOVE(2) 'Make incremental index moves..
    WAIT IDLE 'Wait for move to complete
    WA(100) 'Dwell in ms
  NEXT x
  WA(500)

  MOVEABS(0) 'Move back to absolute "0" position
  WAIT IDLE
  OP(28,ON)      'Set drive DIGITAL OUT1 on
  WA(1000)
  OP(28,OFF)     'Set drive DIGITAL OUT1 off
GOTO start
```