POSITIONING AND CONTOURING CONTROL SYSTEM

MCU-3000, MCU-3100 AND MCU-3400C

OPERATING MANUAL / OM

Rev. 12/052015

1 Introduction

The boards MCU-3000, MCU-3100 and MCU-3400C are universal positioning and contouring control systems for machine tools, robots, handling gear and special purpose machines. Movements and process sequences can be automated through the entry of CNC programs and parameters.
The MCU-3000 or MCU-3100 is designed as a PCI expansion board for the IBM-AT computer range, IPC (Industrial Personal Computer) or compatibles, and is used to control 1 to 3 CNC axes with either servo or stepper motors. The MCU-3400C is a CompactPCI board with up to 4 axes.
This manual essentially describes the hardware and software components of the accessories included in the standard scope of delivery, and is divided into three parts:

- **OM**: Operating Manual
- **PM**: Programming and Reference Manual
- **CM**: Commissioning Manual

The *OM* describes the MCU-3000 or MCU-3100 hardware and software components available in the standard scope of delivery. The user is given an overview of the MCU-3000 or MCU-3100's functions, the operating modes and programming methods.

The Programming Manual describes the various types of programming involved. This section is rounded off by a command and reference list.

The Commissioning Manual describes how to install and commission all the system components contained in the scope of delivery.

For easier comprehension, we recommend that you work through the Operating Manual first.

2 System hardware

2.1 The boards MCU-3000, MCU-3100 and MCU-3400C

The MCU-3000 or MCU-3100 is a PCI plug-in board for AT or IPC (Industrial Personal Computer with AT interfaces) PCs or compatibles, and constitutes the control system’s intelligence.
The minimum configuration, comprising an MCU-3000 or MCU-3100, supports three CNC axis channels with either stepper or servo drives (DC or AC/EC). The maximum configuration supports eight axis channels. For this, a daughter board (OPMF) for 5 further CNC axes is plugged into the MCU-3000 or MCU-3100 using sandwich technology.

The MCU-3000 or MCU-3100 is accommodated on a short PCI plug-in board for IBM AT or compatibles. Communication between the PC and MCU-3000 or MCU-3100 takes place via the PCI bus. This ensures a high data throughput for outputting commands and reading in status information. The MCU-3000 or MCU-3100 address is managed by Plug+Play Bios of the PC.
The MCU-3400C is fitted on a 100 x 160 mm plug-in CompactPCI board with a 78-pin SUB-D connector. The notes in this manual apply to both boards. Any differences are highlighted.

2.1.1 The interface logic of the MCU-3000 or MCU-3100

A significant feature of the MCU-3000 or MCU-3100 board is the use of state-of-the-art PLD and FPGA technology.
FPGA modules are logical modules with a high level of functionality in service that can be reprogramd as often as required. These contain the logic for actual value acquisition by the measurement systems. This includes, *inter alia*, processing of many common incremental coders, processing of SSI absolute encoders, and the precise real-time enabled temporary storage of position values based on various conditions. Of course, customised adjustments can also be made to the application.
The complete I/O periphery is electrically isolated from the system electronics (PC logic). The interfacing of external components occurs according to current guidelines for industrial electronics.

2.1.2 Connection of the external system components

The external system components mentioned in [section 2.2], such as power amplifiers, incremental encoders, limit switches, inputs and outputs, are interfaced through a 50-pin SUB-D pin connector at the front of the MCU-3000 or MCU-3100 board.
Three CNC axes can be connected to this connector with either AC or DC motors.
The MCU-3000 or MCU-3100 provides three analog channels with an output voltage range of +/-10 V and 16-bit resolution. These channels are electrically isolated from the MCU-3000 or MCU-3100’s digital power supply (PC power supply) and are used to control commercial power amplifiers that are connected as speed or current regulators.
Three stepper and directional signals, each with an antivalent signal level and a control voltage of 5 V (RS422) are provided to drive stepper motor amplifiers.
Three pulse acquisition channels, each with 32-bit register width, are used for position acquisition using incremental encoders or linear scales. Pulses are acquired either for TTL sensors or for sensors with symmetrical outputs (RS422). The quadrature signals generated by these sensors are electronically quadrupled and are also electrically isolated from the system electronics. The maximum pulse input frequency is 2 MHz. The MCU-3000 or MCU-3100 can also evaluate a reset pulse. The RS422 input logic of the MCU-3100 is provided with line break detection.

The actual position of the measurement system can be temporarily stored in registers using the reset pulse or a digital input. These measurement methods allow simple encoder verification, setting of reference marks or realtime-enabled position latch for measurement machines.
As an alternative to the above-mentioned incremental encoders, all standard SSI absolute encoders can also be evaluated.
The MCU-3000 or MCU-3100 supports the connection of 16 optically decoupled inputs and 8 optically decoupled outputs. The inputs and outputs are not grouped specific to the axes. However, through software planning of the inputs and outputs, axis-specific assignment can be assigned specific functions, such as a limit switch function, a reference switch function and power amplifier enable, etc. All inputs and outputs which are not assigned special functions are freely programmable.
A watchdog circuit ensures safe operating states, even in exceptional situations. A flash memory is used to record various operating parameters. This includes, for example, the output setpoint status after system power-up.

2.2 External system components

Due to the digital signal processing function and the standard setpoint and actual value interfaces that are available on the MCU-3000 or MCU-3100, the MCU-3000 or MCU-3100 can be used with different motor types and power amplifiers, irrespective of the output range.
The external components are selected to suit the application concerned, after taking into account the performance class, functionality, safety considerations and cost-efficiency constraints.
All commercial power amplifiers can be controlled using a +/-10 V setpoint value channel. The power amplifiers can be connected as speed controller or as current amplifiers.
Both stepper and servo motors can be selected. All motors with a position feedback feature are designated as servo motors. This category includes brushless (AC or EC) and brush-gear equipped DC motors, or hydraulic motors. A position acquisition function is required to position these motor types. In general, rotary transducers or linear scales are used for this purpose. To control stepper motor power output stages, the system provides pulse and directional signals. In contrast to a servo drive, the stepper motor drives can be run without a position feedback feature.
Additional external system components can be linked to the MCU-3000 or MCU-3100 via digital inputs or outputs. Software planning of these inputs and outputs makes it possible to implement limit switches, reference switches, emergency stop, amplifier enable and other functions.

2.3 Types of power amplifiers

Various types of power output stages can be used to provide the energy for the electrical drive. All the types listed below can be controlled directly by the MCU-3000 or MCU-3100.
One of the features determined by the different types of power amplifiers is the characteristic control response of the controlled systems, which has to be allowed for when setting the filter parameters [CM / section 6.2].
Note: Most power amplifiers, except stepper motor power amplifiers, are usually controlled through the analog setpoint value channels of the MCU-3000 or MCU-3100 board. Pulse and directional signals, and their inverted pulse trains are available for the use of stepper motor power amplifiers. This interface, however, is sometimes also implemented on servo systems with digital power amplifiers. In this case, a positioning control must be dimensioned on the power amplifier. The MCU-3000 or MCU-3100 must then be planned as for a stepper motor.

2.3.1 Speed controllers

These are the usual commercially available units used for controlling the speed of a DC or brushless DC motor. The input variable here is usually a voltage of +/- 10 V, corresponding to a speed of +/- maximum speed. The speed actual value is fed directly to the speed controller, usually as a tacho signal. When the amplifier is enabled, the speed controller will develop a holding torque even without a setpoint value signal, i.e. the motor axis will drift away more or less slowly, due to the input offset. The input offset can normally be set, but is temperature-dependent.

2.3.2 Current amplifiers

In contrast to the speed controller, the current amplifier does not need a tacho signal as an actual value feedback from the motor. The input setpoint value here has the significance of an armature current. Many manufacturers offer their power amplifiers as current amplifiers or speed controllers. The commercially available speed controllers can usually be quickly and easily converted into a current amplifier. Please follow the instructions of the manufacturer concerned. The amplification factor should be set so as to ensure that the setpoint maximum current corresponds to an input setpoint value of 10 V.

2.3.3 Voltage amplifiers

In contrast to the speed controller, the voltage amplifier does not need a tacho signal as an actual value feedback from the motor. The input setpoint value here has the significance of an armature voltage. In practice, voltage amplifiers are not of major importance and are at most used for small ratings when the motor current is limited by the armature resistance. Note that the available acceleration decreases at higher speeds.

2.3.4 Stepper motor power amplifiers

Stepper motors usually have 2, 3 or 5 phases or windings to be controlled. In order to ensure a rotary motion, current has to flow in the individual motor windings in a defined cycle. All the usual stepper motor power output stages available on the market will generally satisfy these requirements, and are controlled with the pulse (one motor step per pulse) and directional (specifies the direction of rotation of the motor shaft) input signals. The above-mentioned current flow results from this input information.

2.4 Position latch functionality

2.4.1 Hardware latch strobe

The MCU-3000 or MCU-3100 has various functions for temporarily storing position actual values via a hardware signal. On servo systems with an incremental encoder interface, the hardware latch (latch strobe) is provided for this. A fast digital input is assigned to each axis channel. Activating this input results in latching the current counter state in the position register LP (see PM, Section 4.4.81). At the same time, the “STRBL” bit in the register of the digital inputs is set on the relevant axis channel. Detection of the hardware signal is edge-triggered, i.e. even short pulses (up to approx. 50 ns) are reliably detected. The actual latch process is practically delay-free (< 200 ns). The “STRBL” signal bit must be manually reset if necessary by writing in the register of the digital inputs (SAP programming) or using the PCAP function “rdigi”. The polarity of the edge detection can be inverted in mcfg.

2.4.2 Software latch strobe

In addition to this hardware function, a digital input of the input group of an axis channel can also be declared as a software latch input (in the mcfg.exe program). In this case, the relevant input is called statically once for each sampling interval. This function can also be used on stepper motor systems and on pulse acquisition systems other than incremental encoders, such as SSI. However, this is a pure software function, i.e. a delay of up to one sampling interval is possible between the pulse edge and the latch time. It is also possible that short pulses (< 1 sampling interval) will not be detected. The detection of a latch signal is shown in the "LPSF" bit of the axis-specific AXST register. By reading out the latched position value in the position register LP, this bit is automatically deleted.

Warning: If the hardware and software position latch are used at the same time on one axis, conflicts may arise because the position register LP is always used to read out the latch position. On servo systems with incremental encoder interfaces, the hardware latch functionality must therefore always be used, because in this case, it is always active and cannot be deactivated.

2.4.3 Zero trace hardware latch strobe

On servo systems with an incremental encoder interface, there is also an option of a zero-trace latch. Activating the zero-trace signal on the incremental encoder causes latching of the current counter state in the position register LPNDX (see PM, Section 4.4.82). At the same time, the "NDXL" bit is set in the register of the digital inputs on each axis channel. Detection of the hardware signal is edge-triggered, i.e. even short pulses (up to approx. 50 ns) are reliably detected. The actual latch process is practically delay-free (< 200 ns). The "NDXL" signal bit must be manually reset if necessary by writing in the register of the digital inputs (SAP programming) or using the PCAP function "rdigi". The polarity of the edge detection can be inverted in mcfg.

3 Introduction to how to operate and program the MCU-3000 or MCU-3100

The MCU-3000 or MCU-3100 can be operated and programd in several different ways. All tools and drivers are customized for the 32-bit Windows operating systems (Windows NT, 2000, XP, Vista and Windows 7).

3.1 Manual operation

The MCU-3000 or MCU-3100 is manually operated using the *mcfg.exe* utility program, which is part of the TOOLSET software [TSW] of the MCU-3000 or MCU-3100 [section 2.2].
This offers the user a multitude of operating types for controlling the MCU-3000 or MCU-3100, including manual operation of the complete axis system.

3.2 Programming with a PC application program (PCAP)

The user uses a high-level programming language such as *C* or *Pascal* to create a user program for use on the PC. With the help of function libraries, this application program is used to execute mcug3.dll commands on the MCU-3000 or MCU-3100 via a 32-bit DLL driver. This DLL in turn requires a hardware driver (mini-port driver), which enables accesses via the PCI bus. The PC user program is responsible for the coordinated sequence of the individual axis systems in this operating type. The PC Application Program shall be referred hereinafter as PCAP.
Commands are transferred using pre-defined commands. These in turn have been implemented as DLL functions for 32-bit Windows. There is a source or header file and, if necessary, corresponding lib. files for the above-mentioned programming languages.
From the user's perspective, the DLL functions merely represent a functional extension of the respective programming language. The actual “intelligence” of the functions is in the DLL driver or in the control system.

3.3 Programming as a stand-alone system (SAP)

Another option is to create programs for the so-called stand-alone operating mode. This operating mode enables an operating program that has already been loaded onto the MCU-3000 or MCU-3100 to be executed automatically, that is, without support from a PCAP. This means that the PC can handle other tasks. This Stand-alone Application Program shall be referred to hereinafter as SAP.
The SAP is created by the user with the aid of an editor or the CNC-Edit editor, which is integrated in the *mcfg.exe* utility program. The syntax for this user program is similar to that for *Pascal* and permits simple and flexible program creation. Once the user program has been created, an Autocode file is generated using the *NCC* compiler, which is available both as a command line compiler [section 4.3] and integrated in the mcfg.exe utility program. This autocode file can be transferred to the MCU-3000 or MCU-3100 and executed there automatically without support from the PC. Here, it has to be made sure that for each control type the corresponding CNC file is compiled. The file type results automatically from the type of the system file SYSTEM.DAT which is provided during compiling.
If synchronisation with a PCAP running in parallel is needed, this can be implemented using commonly pre-defined variables, which are accessible in read and write modes by the PC and the MCU-3000 or MCU-3100 alike.

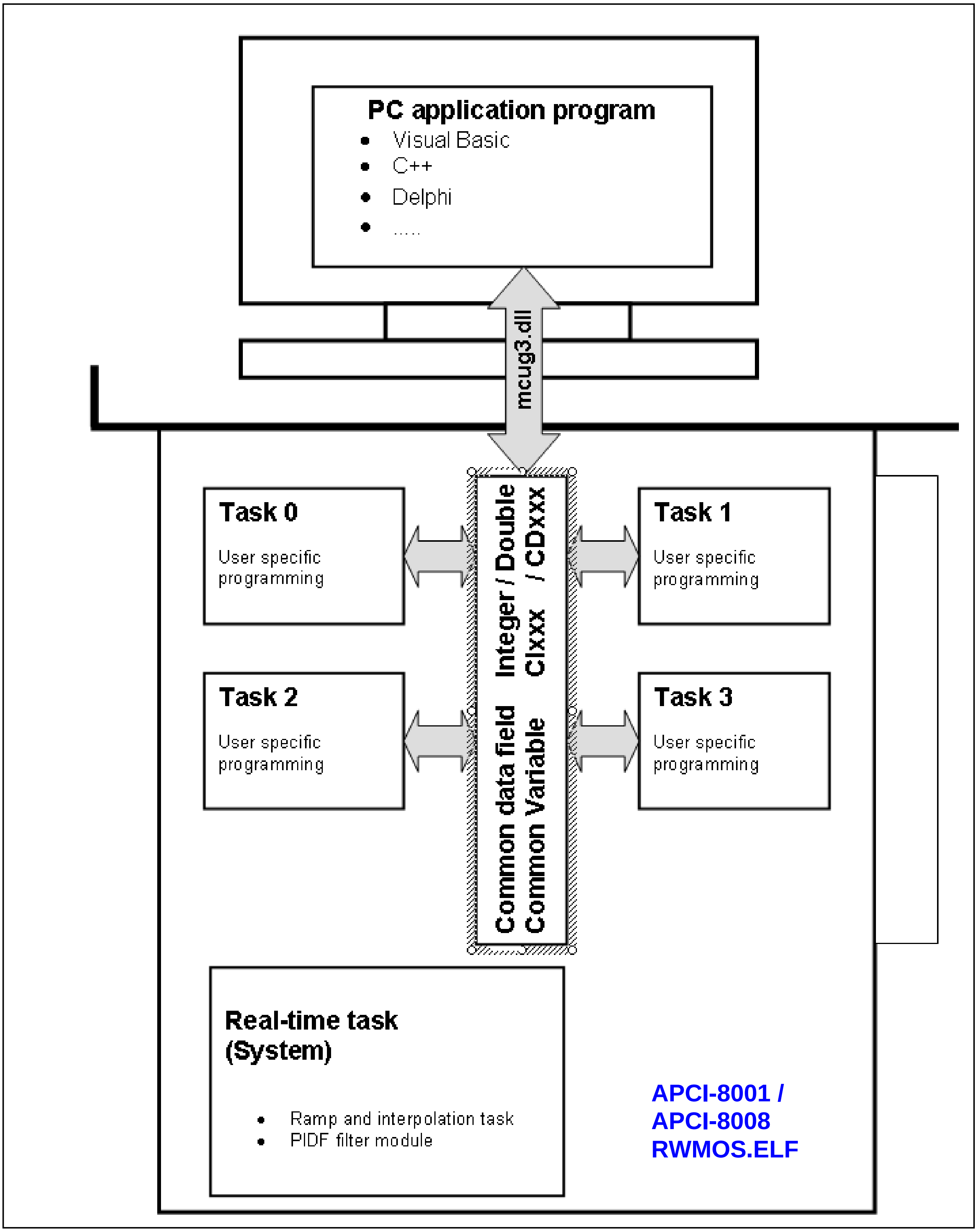
4 System software

This section describes the utility programs for the TOOLSET software [TSW] of the MCU-3000 or MCU-3100 contained in the standard scope of delivery. This information is important for correct operation of the control system and for the creation of user programs.

4.1 The rw_MOS operating system software (rwmos.elf)

The *rw_MOS* operating system software is located in the rwmos.elf file and contains the executable machine code for the MCU-3000 or MCU-3100 CPU system. *rw_MOS* contains various software modules, which are required for the functionality of the control system. For a more detailed understanding of this, the most important software modules are explained in the following sections for interested users. For each control type, an appropriate RWMOS.ELF file is required, i.e. these files cannot be exchanged for the different control versions.

MCU-3000 / MCU-3100 / MCU-3400C: Operating system



4.1.1 PIDF filter module

This module controls the motors to bring them to a desired setpoint position or speed using a digital PIDF filter. This filter is modelled on the analog filter by processing at short, identical intervals. The control cycle time is set to 1.28 ms for all axis channels, but can be varied within certain constraints (100µs – 5ms). All actual value data that is required for the filter computation are also read in and processed in this scan cycle. Setpoint values are then output on the corresponding setpoint value channels.

Note: The PIDF filter is used both for servo and stepper motors. That is, the respective control loop must also be closed before using the axes in stepper motors. Here, the position is controlled at the actual output step pulse. An external position feedback feature is not required.

4.1.2 Ramp and interpolation module

The motors are brought to a desired target position using ramp functions with defined acceleration and pre-defined maximum velocity. This task is handled by the ramp and interpolation module. This module is synchronised with the PIDF filter module and is also executed once per scan cycle (1.28 ms) for all axes.
Another function handled by this module is to synchronise several axes in order to enable a path or space curve to be processed in interpolated mode. There are various interpolation procedures: linear, circular, helical and spline.
There are no restrictions regarding the selection of the axes involved in an interpolation procedure in the standard version (up to 3 axes) or in the multi-axis version (more than 3 axes).
Normally, the velocity course is trapezoidal and parameterised according to the acceleration, maximum velocity and target velocity. When S-profiles are selected, an s-shaped speed course with trapezoidal acceleration course can be set.
Another important feature of the ramp and interpolation module is the second-order dynamic response module. The ramp generation feature implemented in many conventional positioning control systems demands a movement stop in the desired target position. This restriction has been eliminated by implementing this dynamic response module. The user can program any target velocity for the desired target position. The axis system is moved to the desired target position, taking into account the maximum velocity and maximum acceleration, and reaches the programd target velocity at precisely this point. This method enables movement profiles of any desired complexity to be generated.
If you use S-shaped acceleration profiles, please note the following: When a traverse profile is interrupted, for example by another Jog traverse command call, the currently reached acceleration value is increased from 0 again.

4.1.3 PC interface module

This module is the basis for the PC application programming (PCAP programming) and executes the functions and commands called by the user. A wide range of commands is available. These commands are called from the PC Application Program, which is created in a high-level programming language.

4.1.4 Stand-alone CNC module

This module is the basis for stand-alone application programming (SAP programming), and is one of the most powerful characteristics of the MCU-3000 or MCU-3100. Up to four stand-alone user programs (SAP programs) can be created using the *NCC* compiler provided in the scope of delivery. These are processed automatically in multi-tasking mode by the four CNC tasks. Since the drive monitoring function can also be handled completely by an SAP program, the PC is available for other jobs.

4.1.5 Booting rw_MOS

The MCU-3000 or MCU-3100 *rw_MOS* operating system software must be transferred to the MCU-3000 or MCU-3100 each time the system is powered up, and triggers a system initialisation routing (hardware reset). This load operation is required at least once per system start and is concluded within a few seconds. The advantage of booting compared with a ROM-resident operating system is that customer-specific changes to the MCU-3000 or MCU-3100 operating program can be easily incorporated. As soon as the boot operation has been completed, the MCU-3000 or MCU-3100 is ready for operation.

4.2 The *mcfg.exe* utility program

The *mcfg.exe* utility program is executed as a 32-bit MDI application for Windows and offers a user-friendly environment for developing SAP programs, as well as a powerful commissioning, diagnosis and configuration interface.
A detailed description is given in the respective manuals or sections. However, the configuration of the system data is described in brief below, as this is required for the first time the control system is commissioned.
The *mcfg.exe* utility program requires the *system.dat* and *rwmos.elf* files. The *system.dat* file contains the user-specific settings for the respective control system. During installation, a default file from the installation CD is used or generated by the *sysgen.exe* utility program.
When you start the *mcfg.exe* file, all the axis channels for which the *cef* error flat is set are displayed. This error results from data inconsistency between the system file (*system.dat*) and the MCU-3000 or MCU-3100 board. The error can be cleared by saving in the active [System Data] window. A detailed description of the program “mcfg.exe” can be found in the MCFG manual.

4.3 The *ncc.exe* utility program

Using the *ncc.exe* command line compiler, Windows 32 systems can generate stand-alone application programs (SAP programs) directly from the DOS level (input request). The compiler is completely identical to the NCC compiler incorporated in *mcfg.exe*, and is called as follows:

NCC filename [Option]

The text file with the name *filename* must contain an SAP program. The ending *“.src*” is automatically assumed as the file extension name. The various options [*Option*] are explained below and must be separated from each other by blanks.

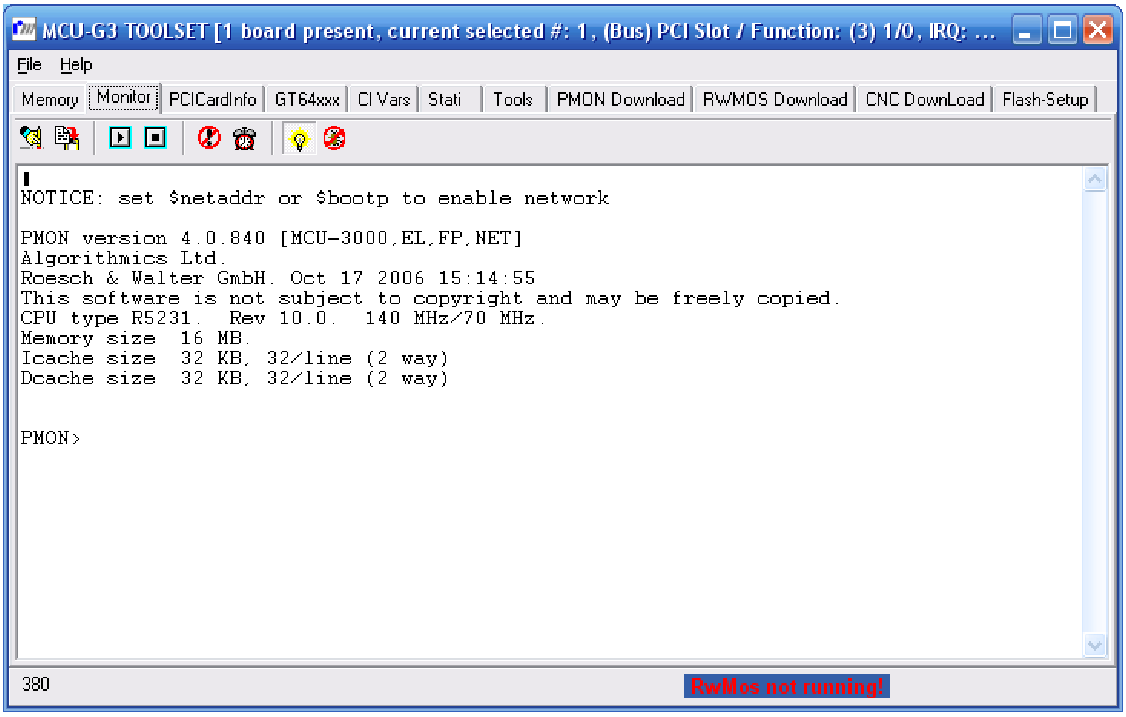
If error-free compilation has proved possible, then an autocode file with the file name *filename.cnc* is created in the current directory. If an error occurs, compilation will be aborted and the faulty line number will be displayed on the screen, together with an error text.

NCC command line compiler options [*Option*]

Option	Function
FS	(Full System) The system does not check how many axes are actually available in the system, i.e. up to 18 different axes designators can be referenced. The default names of these axes designators are A1 .. A18.
SC	(Syntax Check) The system executes only a compiler run, without generating a CNC files. This is primarily done for syntax checking the SAP source text program.
TSK x	(Task Selection, x = 0..3) The autocode file is generated for Task x. But if the SAP program contains a {$TASK} compiler command, the TSK x parameter is ignored.

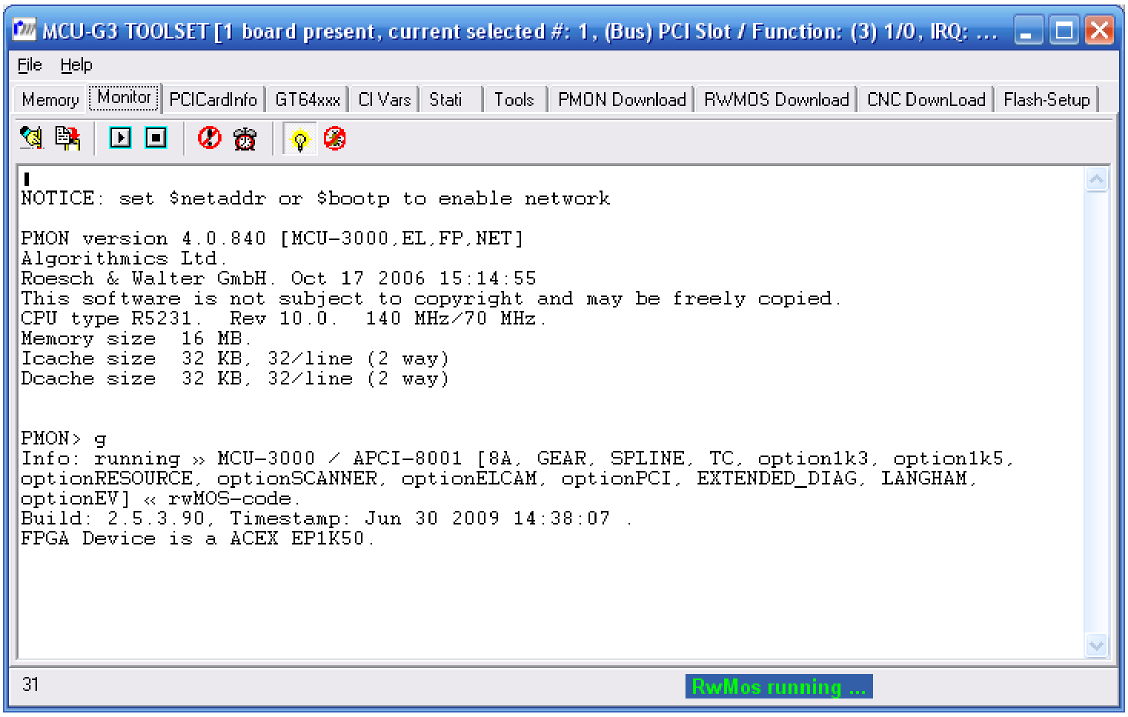
4.4 The *fwsetup.exe* utility program

The fwsetup program is a monitoring program for direct access to the control module. In an unbooted state, information about the bootloader PMON on the control unit can be shown here. The environment variables for configuring the control unit can also be shown and changed. As a result, memory areas can be managed along with axis specifications and the sampling rate set.
Figure: fwsetup "Monitor" screen when the control unit is unbooted



When the control unit is booted, RWMOS.ELF outputs messages on the properties and options which can be verified on the "Monitor" screen. This screen can provide helpful information concerning error states and programming problems in particular, because RWMOS-internal error messages or notes can be displayed here.
RWMOS-internal runtimes and communication states are shown on the "Stati" tab. Notes on using fwsetup.exe for specific use cases, in the case of reinstallation in particular, are described in the Programming Manual (PM).

Figure: fwsetup "Monitor" screen when the control unit is booted


MCU-G3 TOOLSET [1 board present, current selected #: 1, (Bus) PCI Slot / Function: (3) 1/0, IRQ: ...
File Help
Memory
Monitor
PCICardInfo
GT64xxx
CI Vars
Stati
Tools
PMON Download
RWMOS Download
CNC DownLoad
Flash-Setup
NOTICE: set $netaddr or $bootp to enable network
PMON version 4.0.840 [MCU-3000,EL,FP,NET]
Algorithmics Ltd.
Roesch & Walter GmbH. Oct 17 2006 15:14:55
This software is not subject to copyright and may be freely copied.
CPU type R5231. Rev 10.0. 140 MHz/70 MHz.
Memory size 16 MB.
Icache size 32 KB, 32/line (2 way)
Dcache size 32 KB, 32/line (2 way)
PMON> g
Info: running » MCU-3000 / APCI-8001 [8A, GEAR, SPLINE, TC, option1k3, option1k5,
optionRESOURCE, optionSCANNER, optionELCAM, optionPCI, EXTENDED_DIAG, LANGHAM,
optionEV] « rwMOS-code.
Build: 2.5.3.90, Timestamp: Jun 30 2009 14:38:07 .
FPGA Device is a ACEX EP1K50.
31
RwMos running ...