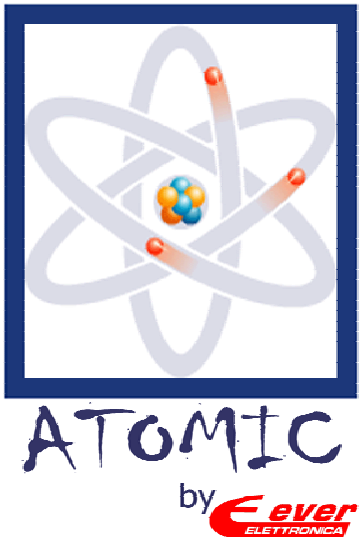
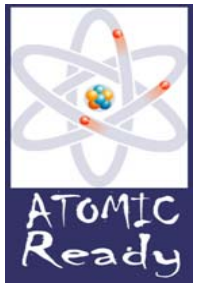
‘Atomic’
Development environment for customized applications



Introduction

- ✓ Atomic is a micro programming language designed for the ***'full digital'*** EVER SDM, MD, SW and SM family drives.
- ✓ The scope of this language is to give the user the freedom to create his own simple application without having to switch to more complex and more expensive drives.
- ✓ Atomic's philosophy is to have few but powerful instructions.
- ✓ The programming of Atomic is done by means of user friendly Personal Computer with the Atomic environment and an SDM, MD, SW or SM drives with 'Atomic Ready' logo and Atomic firmware (c0499).



Introduction

- ✓ The user program Atomic is developed by means of a simple software interface supplied by Ever Elettronica Co. for the windows operating system Personal Computer.

'Atomic' software interface

SDM, MD, SW and SM drives with 'Atomic' firmware

Hardware - Software components

Hardware and software components table compatible with 'Atomic':

Drives	Software kit
- SDMWD180vA133c0499 - SDMWD180vA143c0499 - SDMWA180vA133c0499 - SDMWT180vA133c0499	SDM_ATM-00
- SDMWD170vB231c0499 - SDMWD170vB242c0499 - SDMWA170vB231c0499 - SDMWA170vB242c0499 - SDMWA170v2231c0499 - SDMWA170v2242c0499 - SDMWA170v4231c0499 - SDMWA170v4242c0499	SDM_ATM-00
- SDMWA130vA136c0499	S130_ATM-00
- MDK3AxxxNvPA36c0499 - MDK3AxxxHvPA36c0499 - MDK3BxxxNvPA36c0499 - MDK3BxxxHvPA36c0499 - MDK3CxxxNvPA36c0499 - MDK3CxxxHvPA36c0499 - MDK3DxxxNvPA36c0499 - MDK3DxxxHvPA36c0499	MD_ATM-00
- SW1D4080N361-00c0499	SW_ATM-00
- SM2Ax60PNx43Ax0c0499 - SM2Ax60PNx43Bx0c0499 - SM2Ax60PNx43Dx0c0499 - SM2Ax60PNx43Ex0c0499	SM_ATM-00

The software kits include:

- the **CD ATOMIC** to install the PC environment
- the **RS232 cable** adapted to the requested drives

Technical specifications

✓ These are the specifications and the supported application functions :

- 15 powerful macro instructions
- 8 storable user applications
- 640 bytes of max program size
- 6 bytes of medium length instructions
- 32 user variables
- Supports 4 bytes integer numbers
- Access to all the controlling objects of the drives
- Access to all the I/O's of the drives
- Multitasking support up to 8 user tasks
- A medium execution time of <1 ms per instruction

Atomic advantages

- ✓ Atomic introduces a new way to create an own application:
  - Simple, fast and complete building of the user's cycle
  - A broad versatility for different application requirements
  - The user doesn't have to learn the instructions syntax
  - Functions on all drives of the SDM, MD, SW and SM series with c0499 firmware
  - Integrates all advantages of the f^4d^2 technology
  - Simplifies the debugging of the application
  - It benefits from all the power of the DSP's integrated in the drives
  - Complete support and training by EVER Elettronica's support dept.

Instructions set

Motor handling instructions:

Instruction	Description	Parameters
MOVE (1)	**Starts the motor according to the given parameters**	1) Type of movement (steps/target/homing...).
		2) [optional] Movement Parameter (steps #, position...).
		3) [optional] Profile velocity.
		4) [optional] Acceleration ramp.
		5) [optional] Deceleration ramp.
STOP (1)	**Stops the motor according to the given parameters**	1) Type of stop (no ramp,with ramp,with steps).
		2) [optional] Stop parameter (# of steps).

(1) The functionality of these instructions is the same as in the CANopen / Modbus SDM drives.

Instructions set

Arithmetical instructions:

Instruction	Description	Parameters
ADD	**Performs an addition between two (objects/user variables/numerical constants) storing the result into an (objects/user variable)** **Destination = 1st operand + 2nd operand**	1) Destination (User Variable/Object). 2) First Operand (Numerical Constant,User Variable,Object). 3) Second Operand (Numerical Constant,User Variable, Object).
SUB	**Performs a subtraction between two (objects/user variables/numerical constants) storing the result into an (objects/user variable)** **Destination = 1st operand - 2nd operand**	1) Destination (User Variable/Object). 2) First Operand (Numerical Constant,User Variable,Object). 3) Second Operand (Numerical Constant,User Variable, Object).
MUL	**Performs a multiplication between two (objects/user variables/numerical constants) storing the result into an (objects/user variable)** **Destination = 1st operand * 2nd operand**	1) Destination (User Variable/Object). 2) First Operand (Numerical Constant,User Variable,Object). 3) Second Operand (Numerical Constant,User Variable, Object).
DIV	**Performs a division between two (objects/user variables/numerical constants) storing the result into an (objects/user variable)** **Destination = 1st operand / 2nd operand**	1) Destination (User Variable/Object). 2) First Operand (Numerical Constant,User Variable,Object). 3) Second Operand (Numerical Constant,User Variable, Object).

Instructions set

Application flow control instructions:

Instruction	Description	Parameters
WAIT	**Arrests the user application execution until the specified event occurs**	1) Event (Time delay, Motor running/at standstill, Input open/close).
		2) [optional] Time, input #.
TEST	**Changes the next instruction pointer depending on the test result**	1) Type of Test (Input open/close, Object/User Variable < > = !=).
		2) Object/User Variable.
		3) [optional] Comparison Value.
		4) Line number if the test result is true.
JUMP	**Changes the next instruction pointer**	1) Line number.
CONTEX _SWITCH	**Switches to a different user application stored in the drive**	1) Application number.
END	**End of user application**	None.

Instructions set

Set instructions:

Instruction	Description	Parameters
SET	**Sets the destination equal to the source**	1) Destination (User Variable, Object, Digital Output).
		2) Source (Constant value,User Variable or Object).

Boolean instructions:

Instruction	Description	Parameters
BOOL	**Performs a Boolean Operation : (AND, OR, Shift Right, Shift Left)**	1) Destination (User Variable, Object).
		2) First operand (User Variable or Object).
		3) Second operand (Constant value,User Variable or Object).

Application execution

At switch-on the SDM drive:

- retrieves the startup parameters (Settings, motor currents, ramps...) stored in NVRAM checking if a valid user application is stored;
- starts the program execution from the first line.

During the normal functioning the drive:

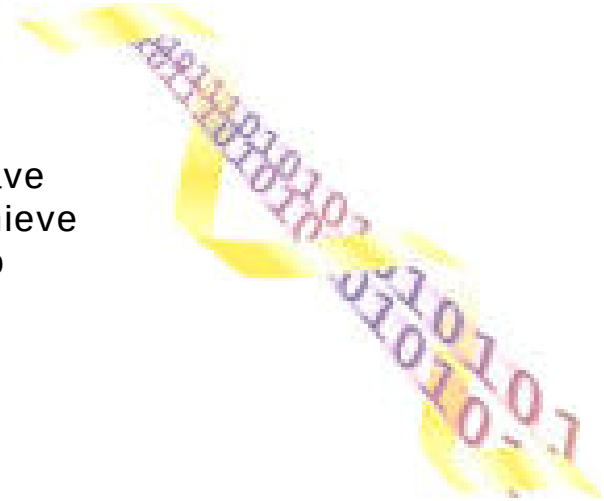
- **runs** the user application;
- **continues to act** as a CANopen/MODBUS standard slave and all the checkings enabled in the 'Drive_Working_Settings' object are active.
- **detects** if there is an emergency condition (hardware or software), in this case the user application is arrested until the emergency situation is cleared.

Language syntax introduction

The Atomic software development environment is a Microsoft Windows(tm) application that enables to build the user application by means of wizards and dialog boxes so it will not be necessary to write a single line of instruction.

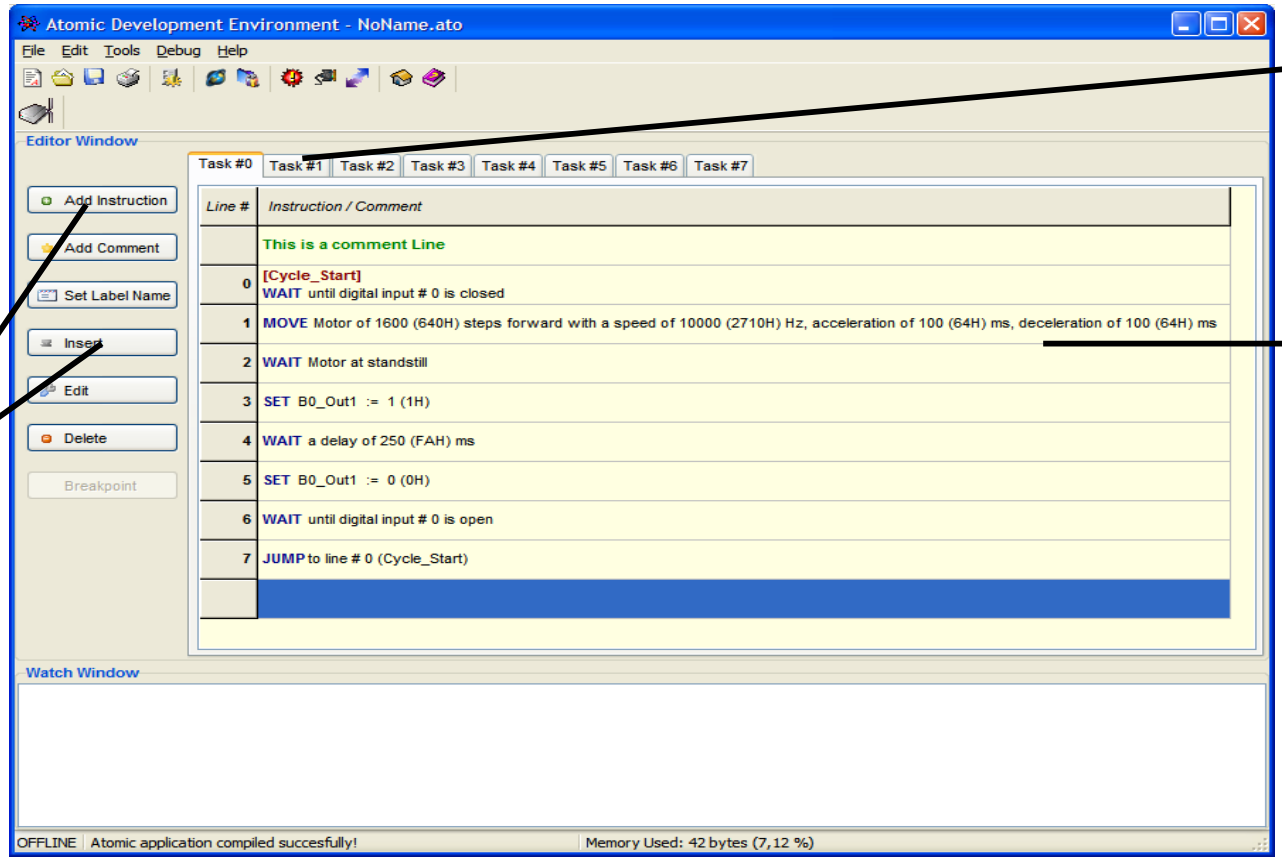
This means that the user doesn't have to learn the syntax of the Atomic instructions that are completely handled by the Personal Computer software.

The configuration windows have been designed in order to achieve a fast compilation time and to offer versatile instructions.



Language syntax

The user program written in Atomic appears as in the example below:



Tab of user task

User's application instructions area

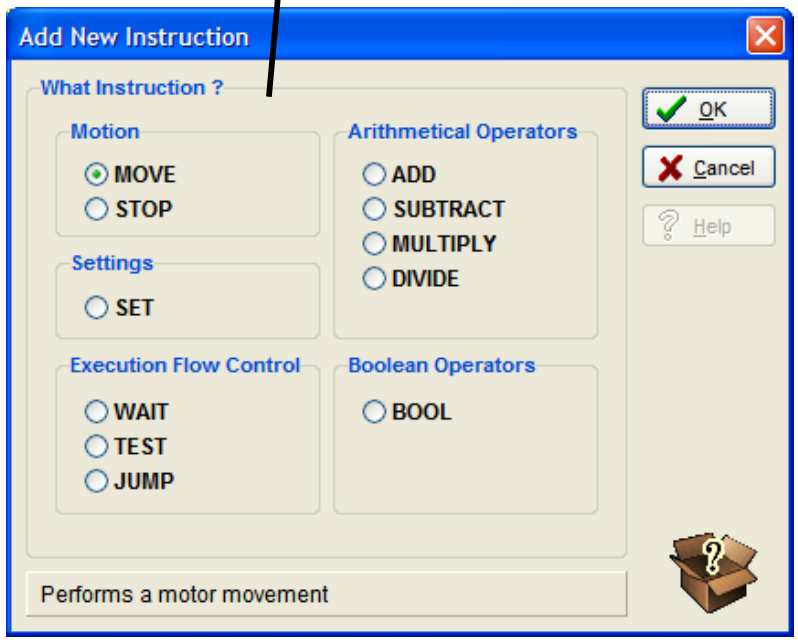
Buttons to add and edit the macroinstructions

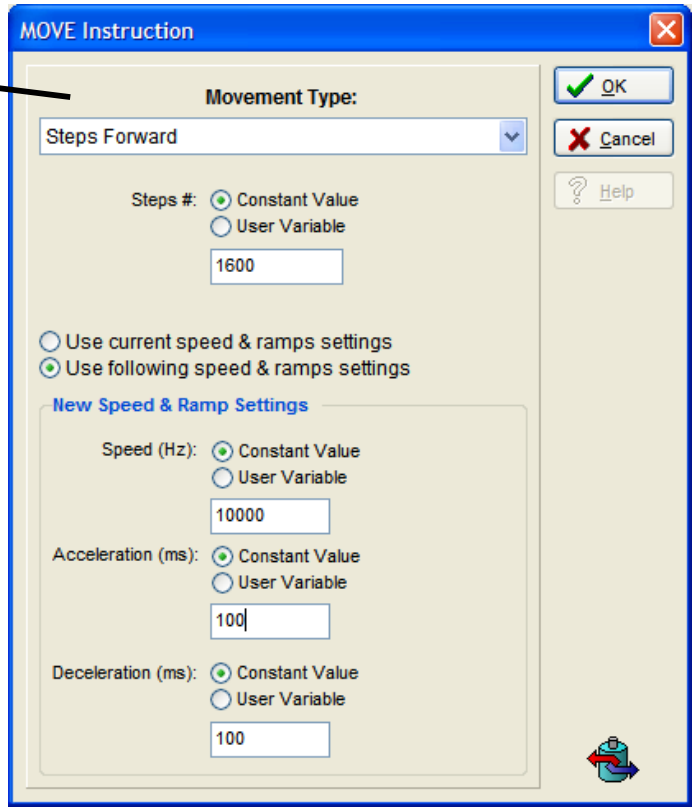
Language syntax

The function buttons to add or edit macro instructions open the following windows:

To add a new instruction

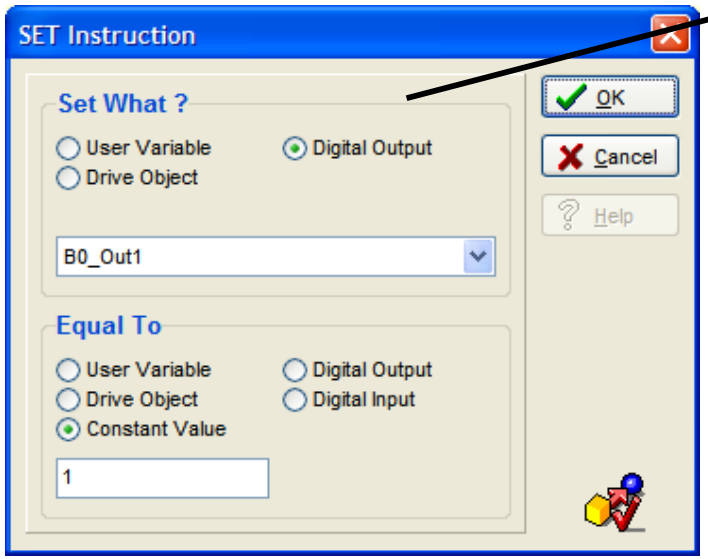
To set the parameters of a MOVE instruction



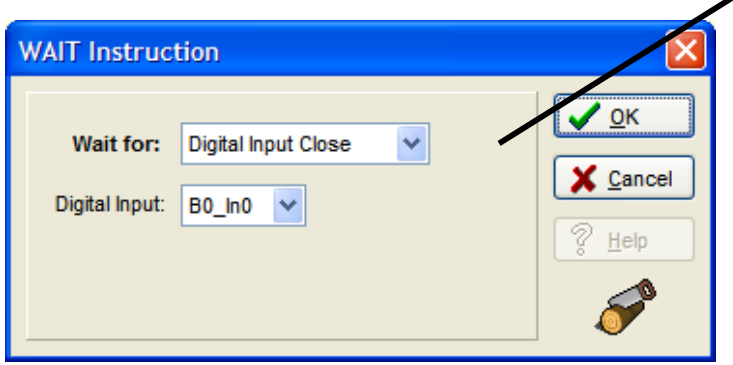


Language syntax

The function buttons to add or edit macro instructions open the following windows:

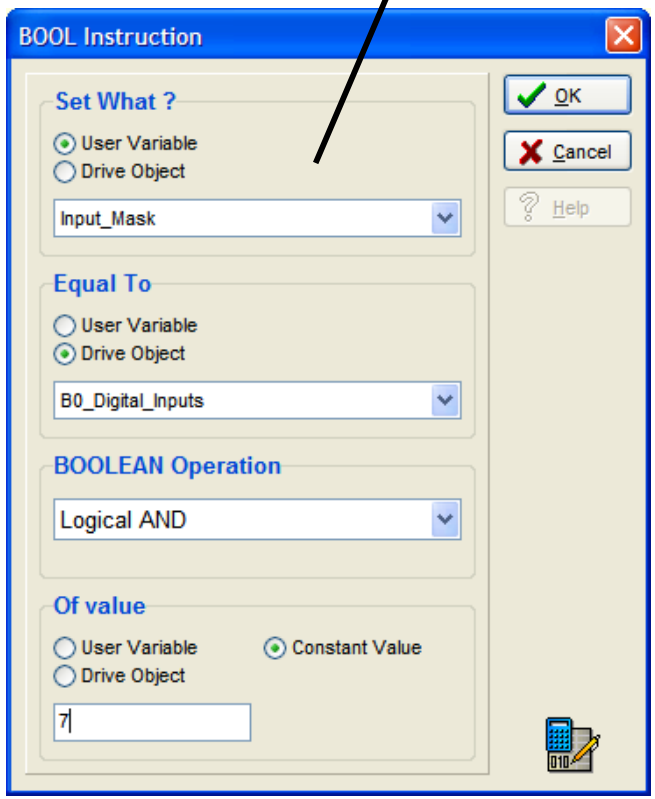


To set the parameters of a SET instruction



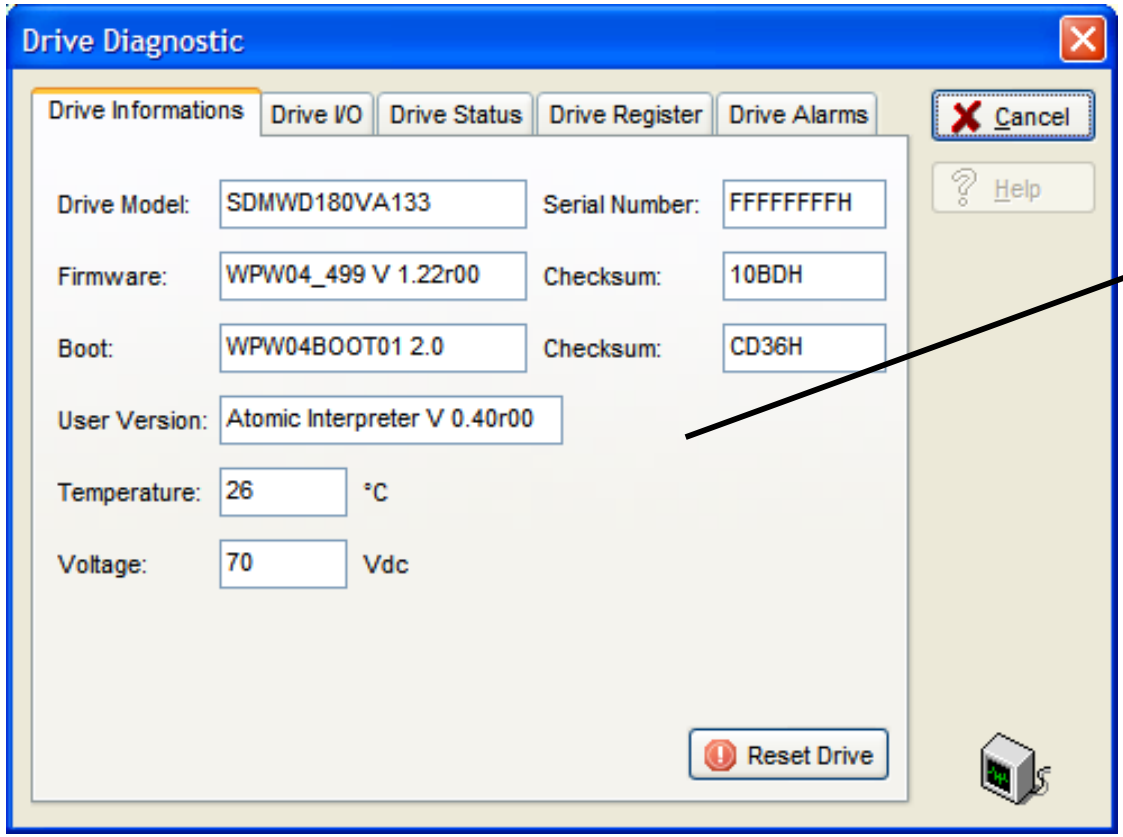
To set the parameters of a WAIT instruction

To set the parameters of a BOOL instruction



System diagnostic

The diagnostic button opens the following window:

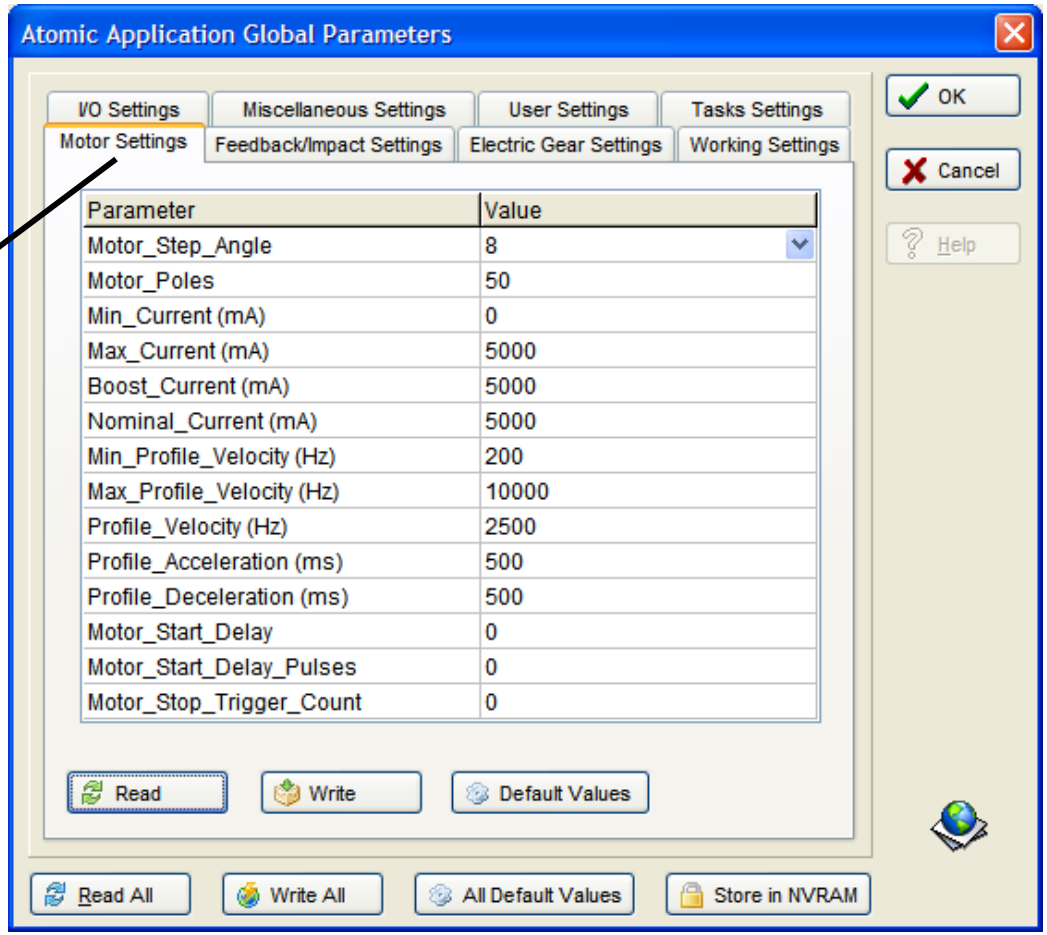


Information about the system in use

System configuration

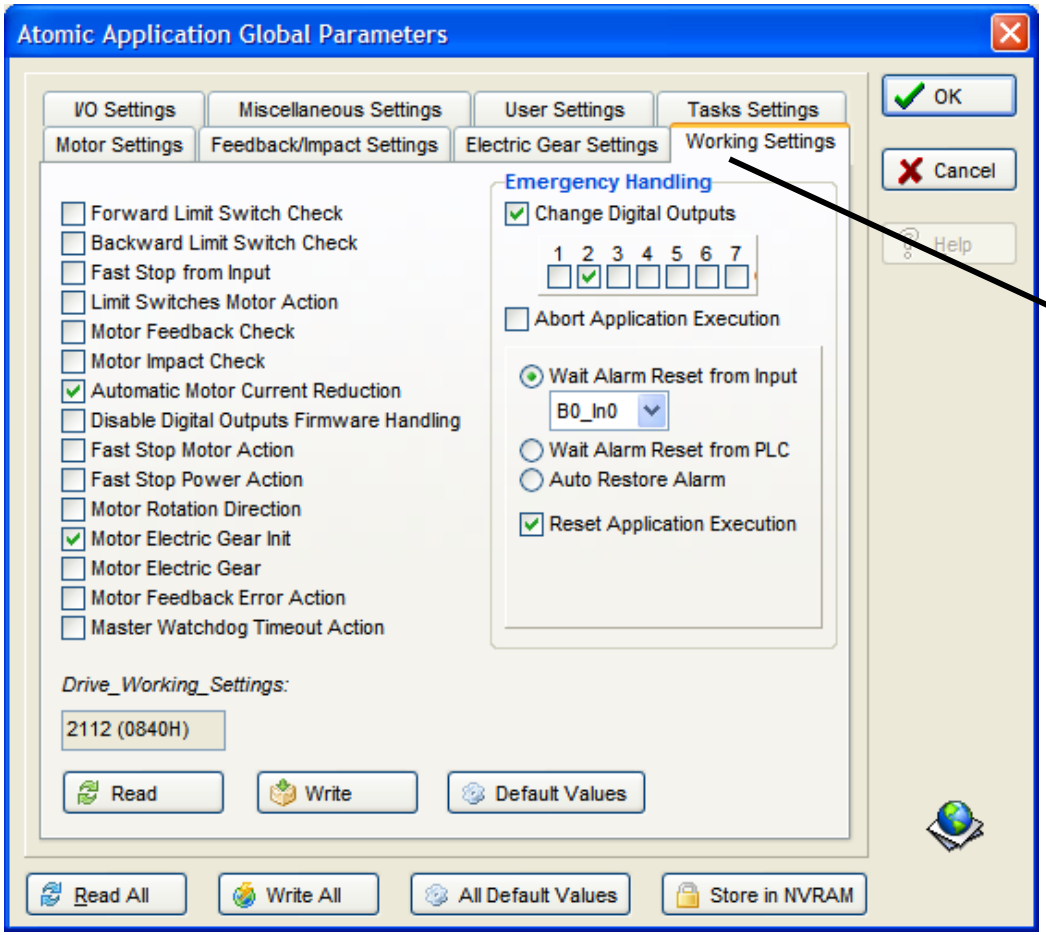
Thanks to the following window it is possible to set the start drive parameters:

Tab that shows the start drive parameters



System configuration

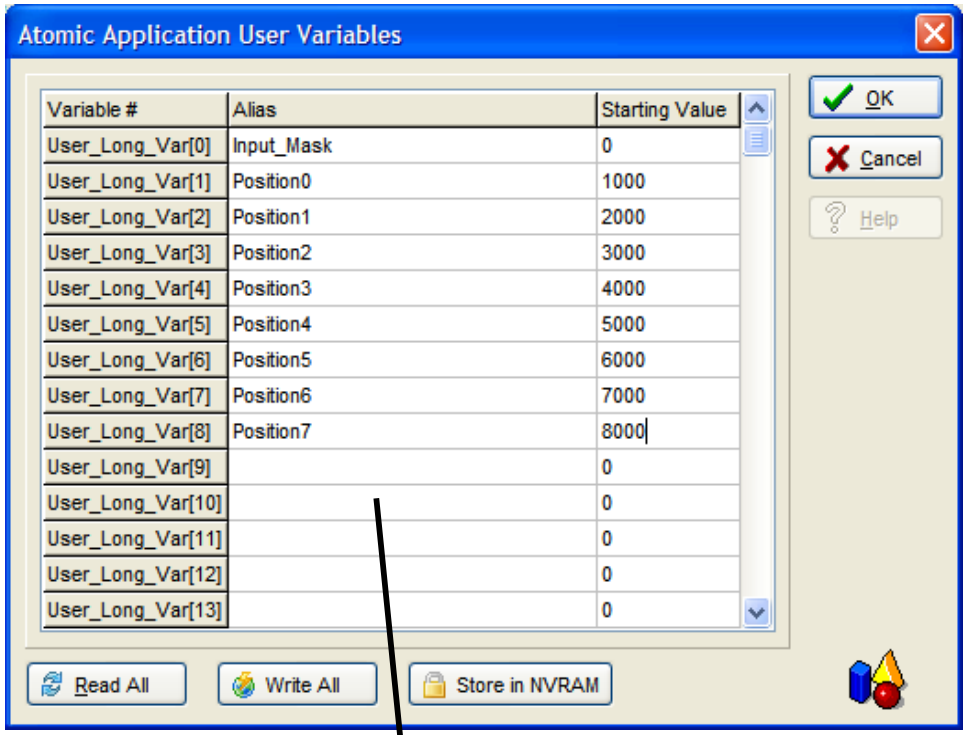
Thanks to the following window it is possible to set the start drive parameters:



Tab to set the start drive parameters

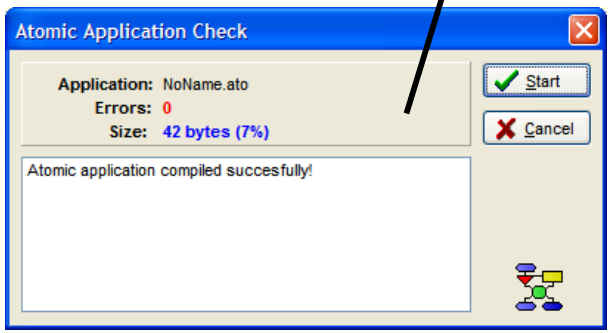
System configuration

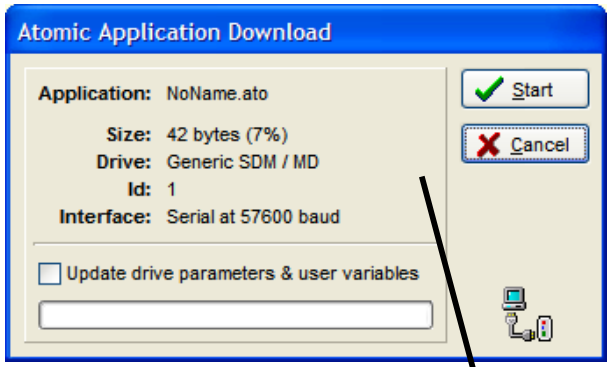
Other utility windows:



Window to manage the user variables

Window to control the user's application language syntax

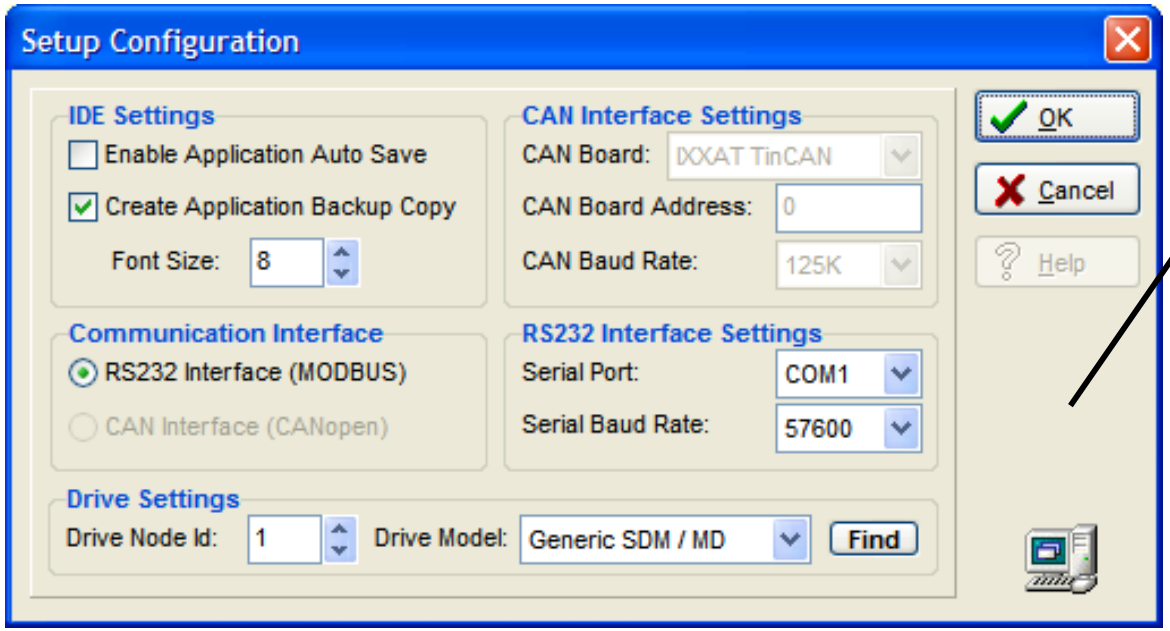




Window to transfer the program application to the drive

PC environment configuration

Other utility windows:



Window to set the Atomic software parameters