

A3200 MotionPAC

Integrated PLC and Advanced Motion Controller

PLC totally integrated with advanced motion

Standards based: IEC 61131-3, PLCopen & .NET

30% to 50% reduction in development time

Global Tag database – define Tags once and then use everywhere

Easy to use diagnostics and debugging tools

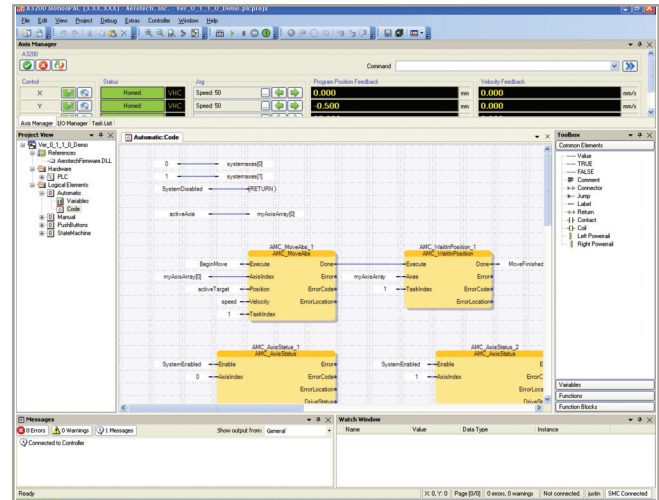
Automatically manages Tag and variable memory

I/O mapping tools

Offline simulation without any changes to program

All motion status and events are exposed in standard function block

The A3200 MotionPAC is a software-based Programmable Logic Controller (PLC) that is completely integrated with the A3200 motion controller. The MotionPAC can be used independently for full machine control or in conjunction with AeroBasic™ programs. MotionPAC complies with IEC 61131-3 and PLCopen. Users can program in Ladder Diagrams (LD), Function Block Diagrams (FBD) or Structured Text (ST). The MotionPAC's tight integration with the A3200 motion controller allows users to command asynchronous, coordinated, blended, geared or cammed motion directly from LD, FBD, ST or directly start, stop or share Tags with AeroBasic™ programs. Programmers can create shared variables and tags that are referenced by Name in all of the A3200 tools, Operator Interface, and existing AeroBasic™ programs as well as in .NET programs. MotionPAC tools like user definable libraries (LD, FBD, ST or .NET), integrated axis manager for readout and manual control, global Tag database, automatic memory management and virtual execution of the PLC increase your programming efficiency and quality. Standard fieldbuses, drive I/O, and high-end data acquisition are integrated into the MotionPAC and can be shared among all



MotionPAC is a powerful software tool that provides motion control completely integrated with PLC functionality.



MotionPAC is available as part of the Automation 3200 software-based machine controller package.

applications by Name. The MotionPAC's Integrated Automation will reduce programming and commissioning time by 30% to 50%.

The MotionPAC is ideal for machine control program development. The development environment includes an axis manager for axis readout and manual control during the development of LD, FBD or ST. This eliminates the need for separate readout and development applications. All of the usual hardware associations, task assignments and programs are organized in an easy-to-use tree structure.

MotionPAC DESCRIPTION

Toolboxes are provided for each language for drag and drop programming or just start typing on the palette and the Intellisense will assist in the programming. The programming palette is free-form meaning that function blocks and functions can be used on a ladder diagram rung as well, even motion function blocks, making programming easy.

Global Tags

Often integration of PLC and motion is difficult and time-consuming due to the lack of coordination between variables and Tags. MotionPAC's Global Tag Database eliminates this. All variables or Tags in the MotionPAC can be shared by name with all of the other applications in the A3200 including the CNC Operator Interface, Scope, Motion Composer, AeroBasic™ programs and even custom .NET programs. Not only are variables available by name, all of the memory management is done for the programmer automatically. Of course, the programmer is free to view the memory addresses or even assign addresses if desired, but it is not necessary.

What are the benefits of Global Tags?

- Often programming takes place in multiple languages – PLC and motion or even .NET. In MotionPAC the PLC programmer defines the Global Tags and they are available in all languages (even .NET or C) by name.
- Global Tags are available in the Operator Interface by name without any memory management required.
- When commissioning a machine, users can plot MotionPAC variables on the same plot as motion variables by name.
- MotionPAC variables can be used in G-code or AeroBasic™ programs by name.

Each of these features eliminates programming errors and minimizes debugging and commissioning time.

MotionPAC programmers can create their own libraries from Function Blocks, Functions, Ladder Diagrams,

The screenshot displays the A3200 MotionPAC software interface. The main workspace shows a ladder logic diagram for a motion sequence. The diagram includes logic for enabling axes, homeing axes, and starting a run cycle. Key components include:

- Enable Axes:** Triggered from push button #3 or from the Operator Interface.
- Home Axes:** Triggered from push button #2 or from the Operator interface.
- Run Cycle:** Triggered from push button #1 or from the operator interface.
- M2_Timer (TON):** A timer block with IN, ET, and Q outputs, and PT (T#02s) input.
- AMC_Enable_1:** A function block with Execute, Done, and Error outputs, and Aves input.

 The interface also features an Axis Manager (A3200) with Control and Status sections, a Project View showing the file structure, an I/O Manager for Axis Y, and a Toolbox with various function blocks and variables. The status bar at the bottom indicates 'X: 0, Y: 0', 'Page [0/0]', '0 errors, 0 warnings', 'Online', 'Running', 'jprofeta', and 'SMC Connected'.

MotionPAC is a powerful software tool that provides motion control completely integrated with PLC functionality.

MotionPAC SPECIFICATIONS and DESCRIPTION

MotionPAC Specifications		
Standards	IEC 61131-3; PLCopen	
Programming Languages	Ladder Diagrams (LD), Function Block Diagrams (FBD), Structured Text (ST), .NET (C#, VB.net), G-code, AeroBasic	
Global Tag Database	Define Tags and variables in MotionPAC for use in Operator Interface and all other applications by Name	
PLC Tasks	16 tasks	
Task Properties	All tasks have Watchdog timer, association with multiple programs (limited by memory) and 18 priority levels	
System Properties	Hot, Cold, Warm Start, Hot Swap of Programs on next scan	
Languages	English; German	
Debugging	Breakpoints, Step Into, Step Over, Step Out	
Simulation	Run program in simulated mode with no code changes; simulate virtual hardware or actual hardware	
Axes	32 axes of coordinated motion can be controlled from LD, FBD, ST or .NET	
Position Modes	Absolute, incremental, dynamic trajectory correction	
Motion Types	Independent Motions	Point-to-point incremental; target position or velocity; velocity profiles; time based; free run
	Coordinated Motions	RS-274 standard G-code motion including linear, circular, helical and spherical interpolation, cutter compensation, normalcy, parts rotation, mirroring, path retrace, polar transformations and cylindrical transformations, scaling
	Electronic Gearing	Electronic master/slave gearing, cam profiling with cubic splining
	Advanced Features	High-speed registration, multi-dimensional error mapping and orthogonality correction, autotuning, backlash compensation, gantry algorithms

Structured Text or combinations of each. These libraries can be referenced and reused in every application. This programming style – code reuse of tested components – aids users in writing quality programs faster.

Motion Capabilities

The motion capabilities include asynchronous, coordinated and blended motion. Moves can be specified as relative or absolute in user units (either English or metric). Other motion commands include electronic gearing, camming, end point modification and more. Fully compliant CNC or G code also can be called from the PLC program. For more details on the motion capabilities integrated into MotionPAC, see Aerotech's Motion Composer software description.

The complete motion state vector and status data is made available as a function block. Programmers can monitor all aspects of the motion directly in their LD, FBD or ST and take action based on motion events without additional programming overhead.

.NET and PLCs

While MotionPAC is completely IEC 61131-3 compliant, it also can be programmed entirely with .NET. Programmers can write custom .NET programs that run in real-time. The programs can be packaged as function block libraries and included in any program as a function block or structured text. The Global Tags created in MotionPAC are available by Name to both the PLC and the .NET programmer. Now PLC programmers and .NET programmers can collaborate in the same environment and use the programming language of their choice.

Protecting Intellectual Property

When protecting intellectual property, the .NET library is ideal. All the functional interface information is available to other programmers or a customer for easy use and maintenance in a PLC language, but the algorithms and methods used are hidden in the compiled real-time .NET code.

Simulation Environment

Programmers can completely simulate the program execution on the PC without changing any of the code. Virtual hardware can be associated with the system so that the programs can be executed in a virtual environment. This coupled with the MotionPAC's debugging tools such as breakpoints, Step Into, Step Over, Step Out functions, forcing I/O, as well as watch windows for tags and variables assist in full featured program debugging and system commissioning in less time.

Fieldbuses

Fieldbus I/O such as Modbus TCP or EtherCAT Tags are also shared globally among all applications and programming languages. While the address of the I/O must be assigned to a physical input, the memory management between applications and programming languages is done automatically for the programmer. The I/O assignment is made easier by the I/O Mapping Tool that displays all of the I/O in the system and the programmer drags and drops I/O points to the Global Tags for association.

MotionPAC DESCRIPTION

Standard features like Cold, Warm, Hot Start and Hot Swapping Programs on next scan are available. MotionPAC can be online and monitoring the program execution with all variables, Tags and state information shown in real-time directly on the program. User Administration is provided to allow full access to the machine or restrict specific functions to certain users. Context sensitive Help is integrated into MotionPAC along with message windows for any errors, faults or system messages.

MotionPAC has 16 tasks available for program execution. Multiple programs can be attached to each task, again aiding in modular programming. Each task has a priority (18 levels), scan time and watchdog timer that can be assigned to provide programmers with exceptional flexibility.

File Management

Programmers can perform file management directly from a FB or ST program. Now system and machine status and production information can be automatically stored for later retrieval and assessment.

MotionPAC programs can be exported to XML documents for consumption by other PLCopen compliant editors. PLCopen XML documents can also be imported from other PLCopen compliant editors.

Multi-Lingual Support

The MotionPAC development environment is multi-lingual. The menus and programming tools are available in English and German, with other languages forthcoming.

MOTION COMMANDS

Motion Function Blocks

- MoveAbsolute
- MoveRelative
- MoveSuperimposed
- MoveVelocity
- Home
- Stop
- PositionProfile
- MoveContinuous
- Halt
- CamIn
- CamOut
- GearIn
- GearOut
- Phasing
- GearInPos
- CamTableSelect
- Blended Motion Uses AeroBasic™ Program

Administrative Function Blocks

- Power
- ReadStatus
- ReadAxisError
- ReadParameter
- ReadBoolParameter
- WriteParameter
- WriteBoolParameter
- ReadActualPosition
- Reset
- PassiveHome
- AbortTrigger
- ReadDigitalInput
- ReadDigitalOutput
- WriteDigitalOutput
- GroupReadConfiguration
- GroupEnable
- GroupDisable
- GroupReadActualPosition
- GroupReadActualVelocity
- GroupReadActualAcceleration
- GroupReset
- PathSelect
- SetPosition
- SetOverride
- ReadActualVelocity
- ReadActualTorque
- DigitalCamSwitch

IEC 61131-3 LADDER DIAGRAM

- VALUE
- TRUE
- FALSE
- COMMENT
- CONNECTOR
- JUMP
- LABEL
- RETURN
- CONTACT
- COIL
- LEFT POWERRAIL
- RIGHT POWERRAIL

MotionPAC DESCRIPTION

IEC 61131-3 FUNCTIONS BLOCKS

- CTD
- CTU
- CTUD
- F_TRIG
- R_TRIG
- RS
- SR
- TOF
- TOF_R
- TON
- TON_R
- TP
- TP_R

FUNCTIONS FOR STRING AND VARIABLE MANAGEMENT

- ABS
- ACOS
- ADD
- ADD_T_T
- AND
- ASIN
- ATAN
- B_BCD_TO_DINT
- B_BCD_TO_INT
- B_BCD_TO_LINT
- B_BCD_TO_SINT
- BCD_TO_DINT
- BOOL_TO_BYTE
- BOOL_TO_DINT
- BOOL_TO_DWORD
- BOOL_TO_INT
- BOOL_TO_LINT
- BOOL_TO_LREAL
- BOOL_TO_LWORD
- BOOL_TO_REAL
- BOOL_TO_SINT
- BOOL_TO_UDINT
- BOOL_TO_UINT
- BOOL_TO_ULINT
- BOOL_TO_USINT
- BOOL_TO_WORD
- BYTE_TO_BOOL
- BYTE_TO_BYTE
- BYTE_TO_DINT
- BYTE_TO_DWORD
- BYTE_TO_INT
- BYTE_TO_LINT
- BYTE_TO_LREAL
- BYTE_TO_LWORD
- BYTE_TO_REAL
- BYTE_TO_SINT
- BYTE_TO_STRING
- BYTE_TO_UDINT
- BYTE_TO_UINT
- BYTE_TO_ULINT
- BYTE_TO_USINT
- BYTE_TO_WORD
- CONCAT
- COS
- B_BDC_TO_DINT
- B_BDC_TO_INT
- B_BDC_TO_LINT
- B_BDC_TO_SINT
- DELETE
- DINT_TO_BOOL
- DINT_TO_BYTE
- DINT_TO_DWORD
- DINT_TO_INT
- DINT_TO_LINT
- DINT_TO_LREAL
- DINT_TO_LWORD
- DINT_TO_REAL
- DINT_TO_SINT
- DINT_TO_STRING
- DINT_TO_UDINT
- DINT_TO_UINT
- DINT_TO_ULINT
- DINT_TO_USINT
- DINT_TO_WORD
- DIV
- DIV_T_AI
- DIV_T_AN
- DIV_T_R
- DIVTIME
- DWORD_TO_BOOL
- DWORD_TO_BYTE
- DWORD_TO_DINT
- DWORD_TO_INT
- DWORD_TO_LINT
- DWORD_TO_LREAL
- DWORD_TO_LWORD
- DWORD_TO_REAL
- DWORD_TO_SINT
- DWORD_TO_STRING
- DWORD_TO_UDINT
- DWORD_TO_UINT
- DWORD_TO_ULINT
- DWORD_TO_USINT
- DWORD_TO_WORD
- EQ
- EQ_STRING
- EXP
- EXPT
- FIND
- GE
- GE_STRING
- GT
- GT_STRING
- INSERT

MotionPAC DESCRIPTION

- INT_TO_BOOL
- INT_TO_BYTE
- INT_TO_DINT
- INT_TO_DWORD
- INT_TO_INT
- INT_TO_LINT
- INT_TO_LREAL
- INT_TO_LWORD
- INT_TO_REAL
- INT_TO_SINT
- INT_TO_STRING
- INT_TO_UDINT
- INT_TO_UINT
- INT_TO_ULINT
- INT_TO_USINT
- INT_TO_WORD
- L_BDC_TO_DINT
- L_BDC_TO_INT
- L_BDC_TO_LINT
- L_BDC_TO_SINT
- LE
- LE_TRING
- LEFT
- LEN
- LIMIT
- LIMIT_DINT
- LIMIT_INT
- LIMIT_LINT
- LIMIT_LREAL
- LIMIT_SINT
- LIMIT_STRING
- LINT_TO_B_BCD
- LINT_TO_BOOL
- LINT_TO_BYTE
- LINT_TO_DINT
- LINT_TO_DWORD
- LINT_TO_INT
- LINT_TO_LREAL
- LINT_TO_LWORD
- LINT_TO_REAL
- LINT_TO_SINT
- LINT_TO_STRING
- LINT_TO_UDINT
- LINT_TO_UINT
- LINT_TO_ULINT
- LINT_TO_USINT
- LINT_TO_WORD
- LN
- LOG
- LREAL_TO_BOOL
- LREAL_TO_BYTE
- LREAL_TO_DINT
- LREAL_TO_DWORD
- LREAL_TO_INT
- LREAL_TO_LINT
- LREAL_TO_LWORD
- LREAL_TO_REAL
- LREAL_TO_SINT
- LREAL_TO_STRING
- LREAL_TO_UDINT
- LREAL_TO_UINT
- LREAL_TO_ULINT
- LREAL_TO_USINT
- LREAL_TO_WORD
- LT
- LT_STRING
- LWORD_TO_BOOL
- LWORD_TO_BYTE
- LWORD_TO_DINT
- LWORD_TO_DWORD
- LWORD_TO_INT
- LWORD_TO_LINT
- LWORD_TO_LREAL
- LWORD_TO_REAL
- LWORD_TO_STRING
- LWORD_TO_SINT
- LWORD_TO_UDINT
- LWORD_TO_UINT
- LWORD_TO_ULINT
- LWORD_TO_USINT
- LWORD_TO_WORD
- MAX
- MAX_DINT
- MAX_INT
- MAX_LINT
- MAX_LREAL
- MAX_REAL
- MAX_SINT
- MAX_STRING
- MID
- MIN
- MIN_DINT
- MIN_INT
- MIN_LINT
- MIN_LREAL
- MIN_REAL
- MIN_SINT
- MIN_STRING
- MOD
- MOVE
- MUL
- MUL_T_AI
- MUL_T_AN
- MUL_T_R
- MULTIME
- NE
- NE_STRING
- NEG
- NOT
- OR

MotionPAC DESCRIPTION

- REAL_TO_BOOL
- REAL_TO_BYTE
- REAL_TO_DINT
- REAL_TO_DWORD
- REAL_TO_INT
- REAL_TO_LINT
- REAL_TO_LREAL
- REAL_TO_LWORD
- REAL_TO_STRING
- REAL_TO_SINT
- REAL_TO_UDINT
- REAL_TO_UINT
- REAL_TO_ULINT
- REAL_TO_USINT
- REAL_TO_WORD
- REPLACE
- RIGHT
- ROL
- ROL_BYTE
- ROL_DWORD
- ROL_LWORD
- ROL_WORD
- ROR
- ROR_BYTE
- ROR_DWORD
- ROR_LWORD
- ROR_WORD
- SEL
- SEL_TO_BOOL
- SEL_TO_BYTE
- SEL_TO_DINT
- SEL_TO_DWORD
- SEL_TO_INT
- SEL_TO_LINT
- SEL_TO_LREAL
- SEL_TO_LWORD
- SEL_TO_REAL
- SEL_TO_STRING
- SEL_TO_SINT
- SEL_TO_UDINT
- SEL_TO_UINT
- SEL_TO_ULINT
- SEL_TO_USINT
- SEL_TO_WORD
- SHL
- SHL_BYTE
- SHL_DWORD
- SHL_LWORD
- SHL_WORD
- SHR
- SHR_BYTE
- SHR_DWORD
- SHR_LWORD
- SHR_WORD
- SIN
- SINT_TO_BOOL
- SINT_TO_BYTE
- SINT_TO_DINT
- SINT_TO_DWORD
- SINT_TO_INT
- SINT_TO_LINT
- SINT_TO_LREAL
- SINT_TO_LWORD
- SINT_TO_REAL
- SINT_TO_STRING
- SINT_TO_SINT
- SINT_TO_UDINT
- SINT_TO_UINT
- SINT_TO_ULINT
- SINT_TO_USINT
- SINT_TO_WORD
- SQRT
- STRING_TO_BOOL
- STRING_TO_BYTE
- STRING_TO_DINT
- STRING_TO_DWORD
- STRING_TO_INT
- STRING_TO_LINT
- STRING_TO_LREAL
- STRING_TO_LWORD
- STRING_TO_REAL
- STRING_TO_SINT
- STRING_TO_UDINT
- STRING_TO_UINT
- STRING_TO_ULINT
- STRING_TO_USINT
- STRING_TO_WORD
- SUB
- SUB_T_T
- TAN
- TIME_TO_DINT
- TIME_TO_STRING
- TRUNC
- TRUNC_DINT
- TRUNC_INT
- TRUNC_LINT
- TRUNC_SINT
- UDINT_TO_BOOL
- UDINT_TO_BYTE
- UDINT_TO_DINT
- UDINT_TO_DWORD
- UDINT_TO_INT
- UDINT_TO_LINT
- UDINT_TO_LREAL
- UDINT_TO_LWORD
- UDINT_TO_REAL
- UDINT_TO_SINT
- UDINT_TO_STRING
- UDINT_TO_UINT
- UDINT_TO_ULINT

MotionPAC DESCRIPTION

- UDINT_TO_USINT
- UDINT_TO_WORD
- UINT_TO_BOOL
- UINT_TO_BYTE
- UINT_TO_DINT
- UINT_TO_DWORD
- UINT_TO_INT
- UINT_TO_LINT
- UINT_TO_LREAL
- UINT_TO_LWORD
- UINT_TO_REAL
- UINT_TO_SINT
- UINT_TO_STRING
- UINT_TO_UDINT
- UINT_TO_ULINT
- UINT_TO_USINT
- UINT_TO_WORD
- ULINT_TO_BOOL
- ULINT_TO_BYTE
- ULINT_TO_DINT
- ULINT_TO_DWORD
- ULINT_TO_INT
- ULINT_TO_LINT
- ULINT_TO_LREAL
- ULINT_TO_LWORD
- ULINT_TO_REAL
- ULINT_TO_SINT
- ULINT_TO_STRING
- ULINT_TO_UDINT
- ULINT_TO_UINT
- ULINT_TO_USINT
- ULINT_TO_WORD
- USINT_TO_BOOL
- USINT_TO_BYTE
- USINT_TO_DINT
- USINT_TO_DWORD
- USINT_TO_INT
- USINT_TO_LINT
- USINT_TO_LREAL
- USINT_TO_LWORD
- USINT_TO_REAL
- USINT_TO_SINT
- USINT_TO_STRING
- USINT_TO_UDINT
- USINT_TO_UINT
- USINT_TO_ULINT
- USINT_TO_WORD
- W_BCD_TO_DINT
- W_BCD_TO_INT
- W_BCD_TO_LINT
- W_BCD_TO_SINT
- WORD_TO_BOOL
- WORD_TO_BYTE
- WORD_TO_DINT
- WORD_TO_DWORD
- WORD_TO_INT
- WORD_TO_LINT
- WORD_TO_LREAL
- WORD_TO_LWORD
- WORD_TO_REAL
- WORD_TO_SINT
- WORD_TO_STRING
- WORD_TO_UDINT
- WORD_TO_UINT
- WORD_TO_ULINT
- WORD_TO_USINT
- XOR

MotionPAC ORDERING INFORMATION

Ordering Example

A3200-IA	-MACHINE	-4 AXES	-DYNAMIC CONTROLS TOOLBOX	-MOTIONPAC	-MAINTENANCE-1-0
Software	License	Number of Axes	Controller Options	Motion Composer Options	Maintenance
A3200-IA	-MACHINE	-2 AXES -4 AXES -6 AXES -8 AXES -10 AXES -12 AXES -14 AXES -16 AXES -32 AXES	-FIVE AXIS CONTOURING -GALVO VTC AND GRC -DYNAMIC CONTROLS TOOLBOX -ENHANCED THROUGHPUT MODULE	-MOTION DESIGNER -CNC OPERATOR INTERFACE -LABVIEW -MOTIONPAC	-MAINTENANCE-y-mm

Automation 3200 Software

A3200-IA A3200: Full installation of Automation 3200 Controller and selected software components on a new system. Full part number includes options listed below. Pricing is summation of selected products. Maintenance (software update) included in price for one year from date of purchase. A3200 is intended for deployment on desktop or industrial PCs running Windows XP (32 bit) or Windows 7 (32/64 bit) operating systems. Please refer to A3200 recommended PC configuration at www.aerotech.com

MOTION COMPOSER STANDARD: Includes the Integrated Development Environment, Parameter Editor, Tuning Scope, System Diagnostics, and System Maintenance. Includes the following software options:

License

-MACHINE Provides the ability to write, compile, execute, debug programs in AeroBasic or G-code; full access to .NET 2.0 and C Library; access full diagnostics, fault, and status information; access and set I/O, registers, and variables; collect, analyze, and save data; view files from machine for analysis and record keeping; connect PC to machine directly with FireWire® card; connect PC to machine remotely through Ethernet TCP/IP; upgrades can be installed (firmware or controller) using Loader; 2 axes of software motion engine; simulate trajectory on PC; installs RTX on the PC; includes A3200-MC Standard

Axes

-2 AXES Software only motion, robotics, and I/O controller; provides 2 axes coordinated motion and 4.5 m FireWire® cable

-4 AXES Software only motion, robotics, and I/O controller; provides 4 axes coordinated motion and 4.5 m FireWire® cable

-6 AXES Software only motion, robotics, and I/O controller; provides 6 axes coordinated motion and 4.5 m FireWire® cable

-8 AXES Software only motion, robotics, and I/O controller; provides 8 axes coordinated motion and 4.5 m FireWire® cable

-10 AXES Software only motion, robotics, and I/O controller; provides 10 axes coordinated motion and 4.5 m FireWire® cable

-12 AXES Software only motion, robotics, and I/O controller; provides 12 axes coordinated motion and 4.5 m FireWire® cable

-14 AXES Software only motion, robotics, and I/O controller; provides 14 axes coordinated motion and 4.5 m FireWire® cable

-16 AXES Software only motion, robotics, and I/O controller; provides 16 axes coordinated motion and 4.5 m FireWire® cable

-32 AXES Software only motion, robotics, and I/O controller; provides 32 axes coordinated motion and 4.5 m FireWire® cable

Note: Old part number – NMotion SMC-nn

MotionPAC ORDERING INFORMATION

Controller Options

-FIVE AXIS CONTOURING Note: Old part number – CNC-5	More than 4 axes of coordinated motion with a single motion command
-GALVO-VCT AND GRC Note: Old part number - Nmark-VCT or Nmark-GRC	Galvo programming interface for vector-type marking applications; requires ScanLabs RTC-4 Galvo control card
-DYNAMIC CONTROLS TOOLBOX	Includes Harmonic Cancellation and Gain Scaling
-ENHANCED THROUGHPUT MODULE	Includes setup and monitoring screens of the ETM module; included in the price of the hardware modules sold separately

Motion Composer (MC) Options

-MOTION DESIGNER Note: Old part number – A3200-MD	The Motion Designer is an add-on software component to the Digital Scope that provides the ability to create, import, run, and evaluate motion profiles (trajectories)
-CNC OPERATOR INTERFACE Note: Old Part Number – Nview MMI	Man-machine interface CNC software for Windows XP/Vista; includes user manuals, technical manuals, and cable drawings for all related equipment
-LABVIEW -MOTIONPAC	Includes LABVIEW 8.2 VI samples Integrated PLC and advanced motion controller