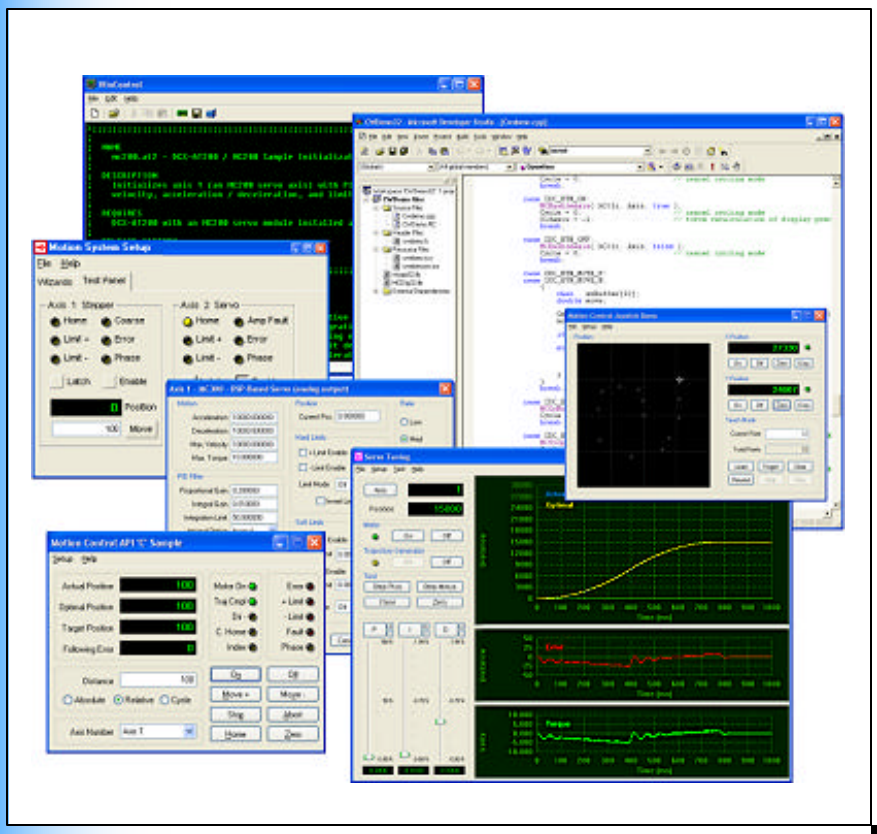
# Programming & Integration

## *An Introductory Guide to Motion Control Programming…*

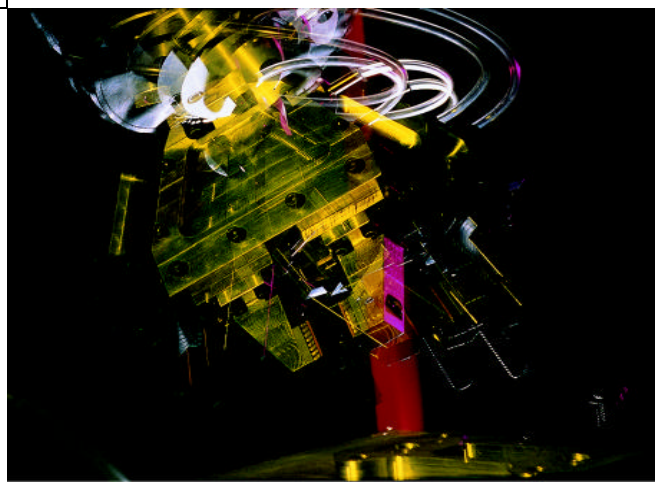


Precision MicroControl (PMC) motion controllers are successfully deployed in a wide variety of applications such as electronic assembly, electro-optics and semiconductor machine automation. PMC recognizes that no matter what the application, good software tools are a critical to the success of any automation project.

To speed system development, PMC offers a comprehensive and powerful high-level Motion Control Application Programming Interface (MCAPI), as well as our Motion Integrator™ suite of graphical setup, tuning and diagnostic utilities. These software tools are designed to help you get up-and-running quickly – whether your control program requires just a few simple motion commands, or the flexibility and power of a multi-threaded C++ host application.

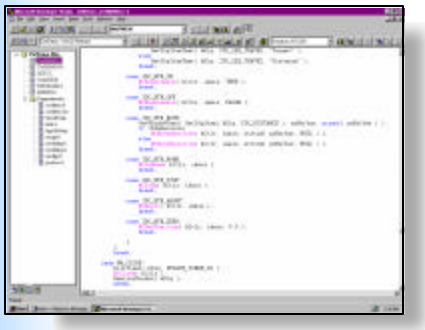
To further simplify system integration and eliminate any hidden costs, all software is included with PMC motion control cards at no extra charge. And to ensure that you get the most out of your programming investment, all functions and commands are compatible across the entire PMC product family.

- *Fully programmable in C /C++, Delphi & Visual Basic*
- *Drivers for Windows 98/2000/NT/XP and Linux*
- *Simple command-based programming also available*
- *Includes Motion Integrator™ setup & diagnostic tool suite*
- *Compatible across PMC controller family*

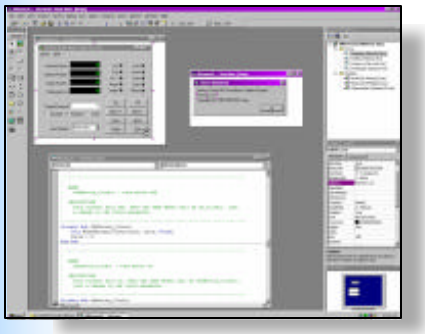


PMC
Precision MicroControl Corp.

## C-Programmable For High-Performance PC-Based Applications



**Microsoft Visual Developers Studio™**



**Visual Basic™ Development Environment**

## MCAPI - Motion Control Application Programming Interface

- Designed for OEM's with demanding PC-based control requirements
- Seamless integration with high-level languages
- Programmable in C/C++, Delphi, LabVIEW, Visual Basic
- Capable of multi-threading under Windows NT/2000/XP and Linux
- Clearly documented with comprehensive on-line help
- Many example programs with source code
- For RTOS support (QNX,VxWorx,VenturCom & others) contact PMC

For the experienced programmer, the MCAPI provides a standard set of functions that can be called from application programs written in C/C++ and other high level languages such as Visual Basic and Delphi (Pascal). In addition, the Motion VI Library provides the LabVIEW and BridgeVIEW programmer with graphical access to the MCAPI. For easy integration with Microsoft COM enabled software (Component Object Model supporting ActiveX components), a standard COM interface to the MCAPI is also provided.

The MCAPI is fully supported under Windows 982000/NT/XP and Linux For the Windows and Linux developer, the MCAPI is a Dynamic link Library (DLL) that communicates with the control card through 32-bit installable device drivers.

The functions of the MCAPI have names consistent with the operations they perform, making programs easy to read and practically self-documenting.

For example, the function call to move 1 axis to position 1000 would appear in a C program as:

MCMoveAbsolute( ctrlr,1,1000,0);

The function call to read the position of axis #8 into a variable is:

Position = MCGetPosition (ctrlr,8);

The Motion Control API (MCAPI) was designed from the ground up to include the power and flexibility that experienced C- programmers have come to expect. The design of the MCAPI includes a high degree of backwards- and cross-platfom compatibility. Code written four, five, or even six years ago will run on the latest versions of the MCAPI.

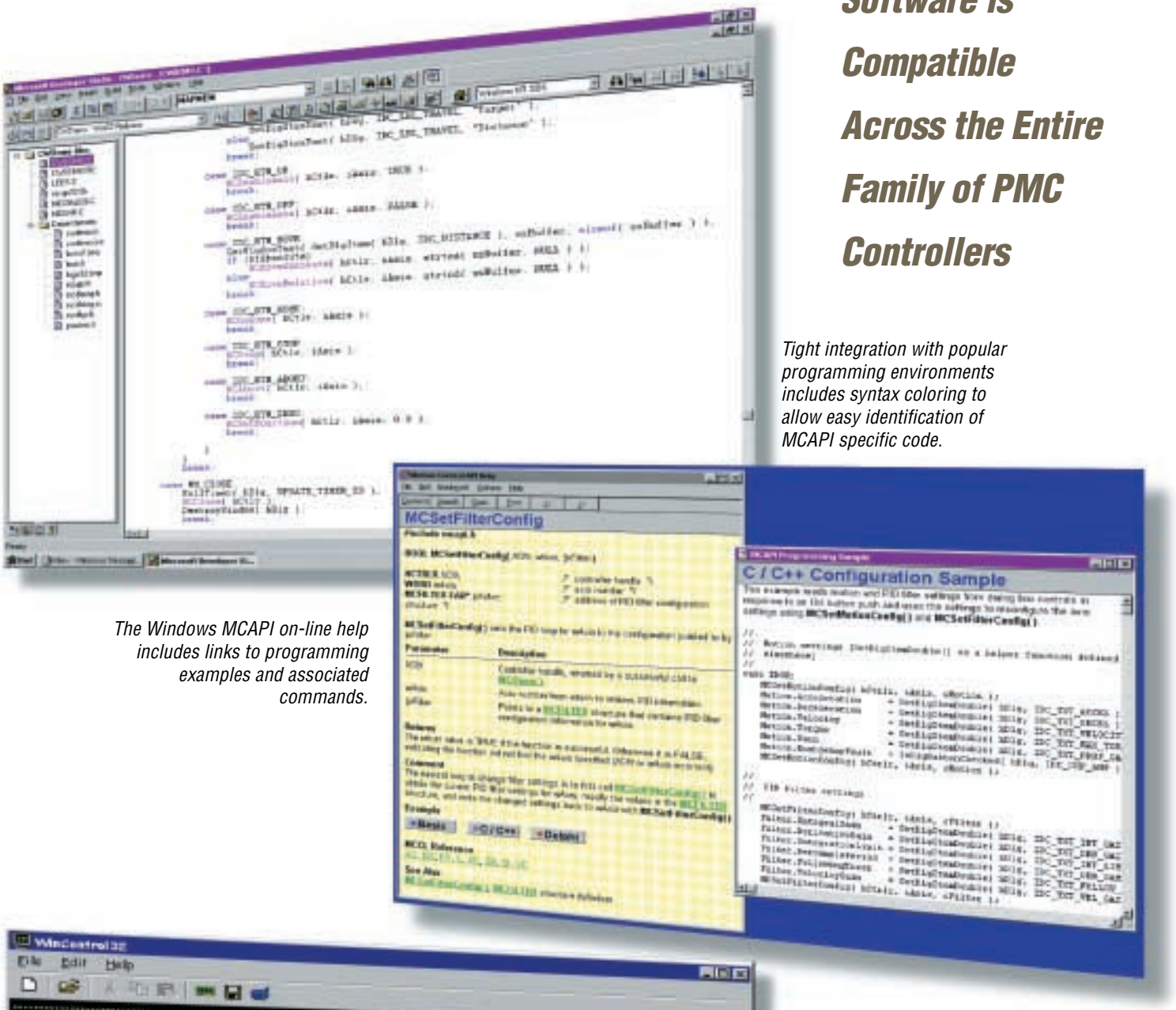
For high-performance motion control applications, communication speed is critical. The MCAPI uses the motion control card's dual ported memory and high-speed direct binary interface for optimum throughput.

## Partial Listing of MCAPI Functions

| | |
|---|---|
| MCEnableDigitalIO( ctrlr, channel, state ); | Change output state of digital I/O channel |
| MCSetJogConfig( ctrlr, axis, lpjog ); | Set jog configuration |
| MCEnableGearing( ctrlr, axis, masis, ratio, state ); | Configure and enable / disable gearing |
| MCDirection( ctrlr, axis, dir ); | Set direction of travel |
| MCEnableAxis( ctrlr, axis, state ); | Turn axis on or off |
| MCGetPosition( ctrlr, axis ); | Get current position |
| MCGetStatus( ctrlr, axis ); | Get status value from axis |
| MCGetVelocity( ctrlr, axis ); | Get velocity value |
| MCGo( ctrlr, axis ); | Start motion |
| MCGoHome( ctrlr, axis ); | Start axis or axes moving to home position |
| MCLearnPosition( ctrlr, axis, index ); | Store current position to point memory |
| MCMoveAbsolute( ctrlr, axis, index ); | Move to an absolute position |
| MCMoveRelative( ctrlr, axis, distance ); | Move a relative distance from current position |
| MCSetAcceleration( ctrlr, axis, rate ); | Set acceleration value |
| MCSetDeceleration( ctrlr, axis, rate ); | Set deceleration value |
| MCSetServoOutputPhase( ctrlr, axis, select ); | Set servo output phase |
| MCGetScale( ctrlr, axis, scale_factors ); | Get current scale factors |
| MCSetScale( ctrlr, axis, scale_factors ); | Set scale factors for unit conversion |
| MCSetFilterCongig( ctrlr, axis, lpflf ); | Set PID filter values |

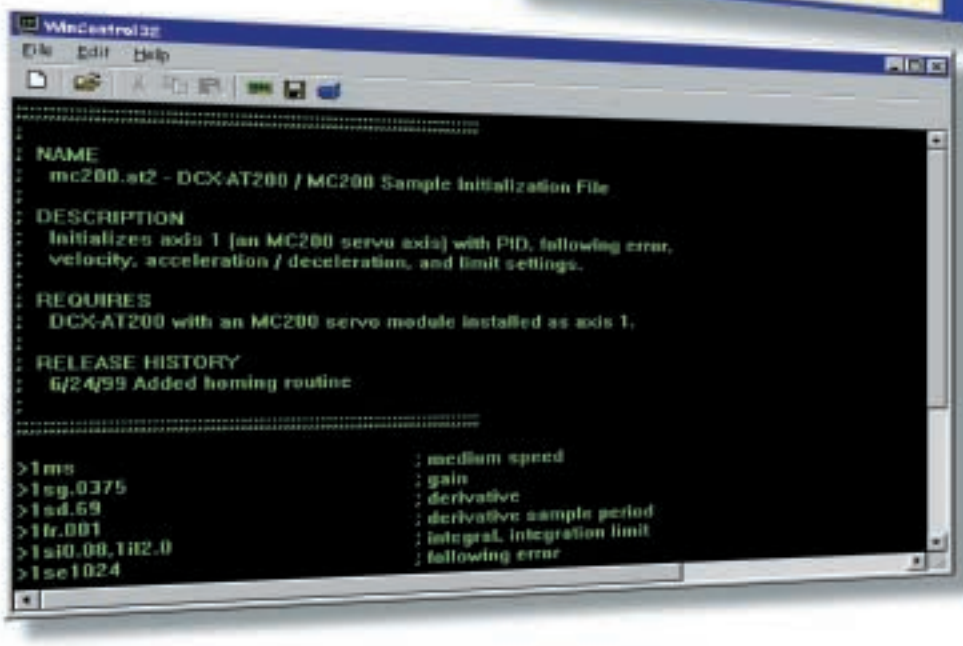
*A representative sample of the more than 130 MCAPI functions. Contact PMC for an up-to-date list.We continually add functions to support new features.*

## *Software is Compatible Across the Entire Family of PMC Controllers*




*Tight integration with popular programming environments includes syntax coloring to allow easy identification of MCAPI specific code.*

*The Windows MCAPI on-line help includes links to programming examples and associated commands.*



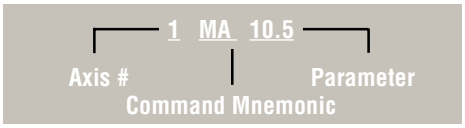

*WinControl is a powerful yet easy to use Windows based terminal emulator for MCCL programming. MCCL commands can be entered from the keyboard or downloaded from an ASCII text file.*

## MCCL – Motion Control Command Language

- Intuitive, easy-to-use commands
- Store & execute multiple programs on-board
- Solve any application, from basic to the most complex
- Ideal for prototyping and embedded control applications

PMC's motion control cards can execute more than 175 MCCL commands, allowing you to perform a wide variety of tasks with a simple on-board command language. Setting a motor's maximum speed, moving a motor to a specific position, or even reporting the current position are just some of the operations that can be performed using the MCCL commands.

Each MCCL command can easily be identified by a two-letter mnemonic. The letters are easy to remember because they relate to the function the command performs. The format for all commands is the same as the example below:




This command causes axis 1 to Move Absolute to position 10.5. By placing commas (,) between multiple commands, they can all be issued at the same time to initiate synchronized multi-axis motion.

The MCCL language includes commands for conditional execution branching and looping. Using these commands, complex control operations can be implemented in user-written "macro" routines. Multiple commands can be linked, permanently stored in the card's memory as a macro command, and used at any time. Macro commands can be written to perform any motion, from a simple homing routine, to controlling an entire machine without the intervention of a host computer.

As with the Windows version of the MCAPI, the MCCL commands can be sent to the card via three different interfaces:

- PC bus (ISA or PCI using the terminal emulator software utilities)
- RS-232/422 serial port
- IEEE-488 interface

With a terminal emulator utility running on the host PC, typing on the keyboard transfers one character at a time to the motion control card. Any response from the card will be displayed on the host computer screen. Motion control commands can also be placed in an ASCII text file and downloaded to the card.

### Partial Listing of MCCL Command Set

#### Setup Commands

| | |
|---|---|
| PP | Profile Parabolic |
| PS | Profile S curve |
| SA | Set Acceleration |
| SD | Set Derivative Gain |
| SE | Stop on Follow Error |
| SG | Set Prop. Gain |
| SH | Step Half/Micro |
| SI | Set Int. Gain |
| SQ | Set Torque |
| SS | Set Slave ratio |
| SV | Set Velocity |
| US | User Scale |
| VA | Vector Acceleration |
| VD | Vector Deceleration |
| VG | Velocity Gain |
| VO | Velocity Override |
| VV | Vector Velocity |

#### Motion Commands

| | |
|---|---|
| CM | Contour Mode |
| CP | Contour Path |
| CR | Arc Center Relative |
| FE | Find Edge |
| FI | Find Index |
| GH | Go Home |
| GO | Start in Velocity Mode |
| MA | Move Absolute |
| MF | Motor Off |
| MN | Motor On |
| MR | Move Relative |
| PM | Position Mode |
| QM | Torque Mode |
| SM | Set Master |
| SN | Synchronization On |
| ST | Stop |
| VM | Velocity Mode |

#### Reporting Commands

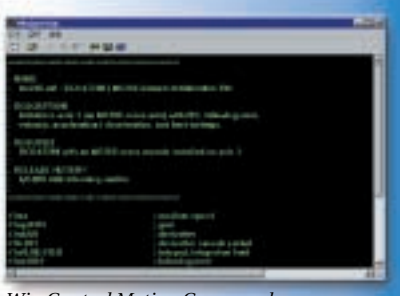
| | |
|---|---|
| AT | Tell Pos. Aux. Encoder |
| TF | Tell Following Error |
| TO | Tell Optimal Position |
| TP | Tell current Position |
| TS | Tell Servo Status |
| TT | Tell Target Position |
| TX | Tell Cont. Count |
| TZ | Tell Index Position |

#### Macro Commands

| | |
|---|---|
| ET | Escape Task |
| GT | Generate Task |
| MC | Macro Call |
| MD | Define as Macro |
| MJ | Macro Jump |
| RM | Reset (clear) Macros |
| TM | Tell Macro |

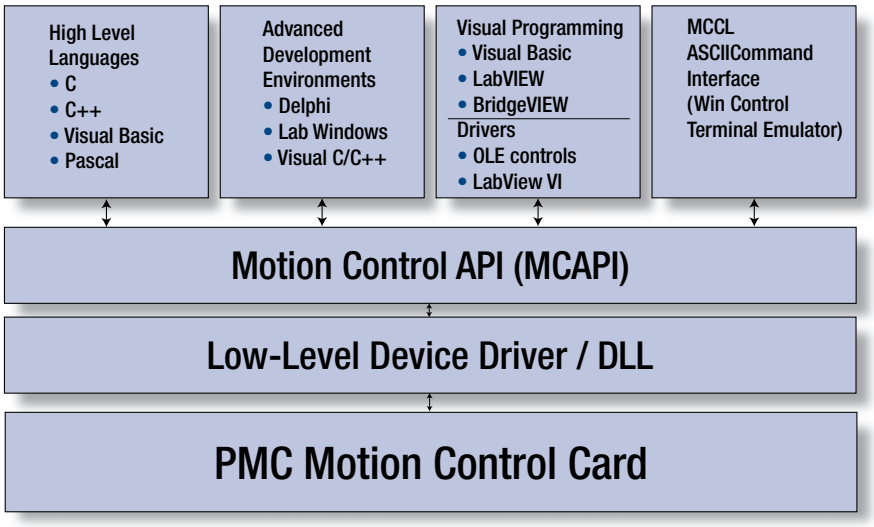
*A representative sample of the more than 175 MCCL commands.*

## Easy-to-Use Motion Command Language for Fast Prototyping & Embedded Control



*Win Control Motion Command Terminal Emulator*

# Motion Contol Programming Options

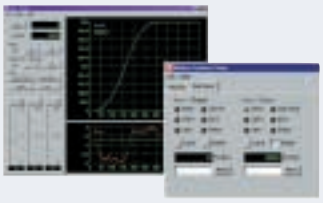



Motion control applications can run on the host PC, the PMC controller, or both.

- ***Host-based application programs:*** For sophisticated applications running on a host PC, the Motion Control API (MCAPI) provides seamless integration between PMC motion control systems and popular high-level programming languages.
- ***Embedded application programs:*** For applications and routines running on-board the PMC controller, the Motion Control Command Language (MCCL) is easy-to-use and frees the host PC for other tasks.

Software included with each controller includes many high-level programming examples and pre-built Windows dialog boxes with source code, as well as extensive online help and PMC's Motion Integrator™, a comprehensive suite of setup, tuning and diagnostic tools.

*Get you controller installed and running in hours instead of days with PMC's graphical and intuitive Motion Integrator™ setup, tuning & diagnostic programs.*

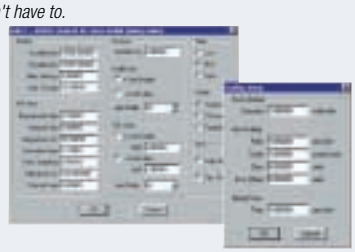


A Servo Tuning and a System Setup dialog box

*Two of the many high-level programming examples - available with fully annotated source code*

"C" and Visual Basic Sample Programs

*Motion Dialog boxes are pre-built for all common setup tasks, so you can incorporate them into your application program with a single function call. They are already written, so you don't have to.*



Servo Setup & Scaling Motion Dialogs

*A complete VI Library with detailed motion control icons is provided for LabVIEW programmers*

LabVIEW programming examples

## *Powerful API & Software Tools Speed System Development*

*PMC's Motion Control API for Windows 95/98/NT/2000 includes complete function libraries for:*

- ***Visual C/C++***
- ***Visual Basic***
- ***Borland Delphi***
- ***LabVIEW***
- ***BridgeVIEW***
- ***Watcom C/C++***

*PMC continuously develops new software tools and provides custom software solutions to qualified OEM's. Contact a PMC application engineer to discuss your requirements.*

# Motion Control Programming Examples

PMC provides programming support for DOS, Windows, and stand-alone applications. Below are two motion control programming examples. Each example shows how to program the same motion using MCCL commands and MCAPI functions.

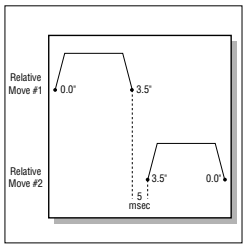
## Basic Positioning

Any or all axes can be easily programmed for independent motion with a minimum of commands.

*Move axis 1, wait for 5 msec, return to starting position*

**MCCL**

```
1SV15,1SA5, 1DS5                ; Set Axis #1 vel., Accel. & Decel
1MR3.5,1WS0.005,1MR-3.5         ; Move axis #1 3.5", wait 5 msec,
                                ; Move-3.5"
```




*5 msec setting time between relative moves*

**MCAPI using "C"**

```
MCSetVelocity( 1, 15.0 );       // set axis 1 velocity
MCSetAccleration( 1, 5.0 );     // set axis 1 acceleration
MCSetDeceleration( 1, 5.0 );    // set axis 1 deceleration
MCMoveRelative( 1, 3.5 );       //  move axis 1 by 3.5 inches
MCWaitForStop( 1, 0.005 );      //  wait 5msec after axis 1 stops
MCMoveRelative( 1, -3.5 );      //  move axis 1 by -3.5 inches position
```

**MCAPI using "basic"**

```
Call MCSetVelocity( 1, 15.0 )       set axis 1 velocity
Call MCSetAccleration( 1, 5.0 )     set axis 1 acceleration
Call MCSetDeceleration( 1, 5.0 )    set axis 1 deceleration
Call MCMoveRelative( 1, 3.5 )       move axis 1 by 3.5 inches
Call MCWaitForStop( 1, 0.005 )      wait 5 msec after axis 1 stops
Call MCMoveRelative( 1, -3.5 )      move axis 1 by 3.5 inches
```

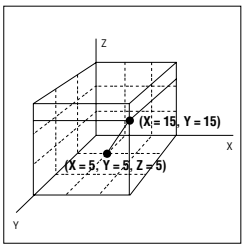
## Linear and Circular Contouring

PMC's motion control cards support simultaneous Linear and/or Circular Contouring. Any number of axes (as many as eight) with any combination of Servo and Stepper motors can be used. The on-board CPU will compute points on the path at intervals of 125 µsec, while still allowing on-the-fly changes of feed rate, acceleration, deceleration, and PID parameters. For example:

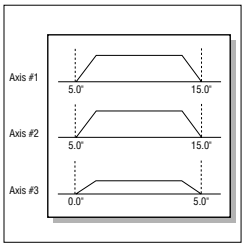
*3D Linear Interpolated move*

**MCCL**

```
1VV10,1VA50,1VD50               ; Set Vector Vel.,Accel., & Decel.
1CM1,2CM1,3CM1                  ; Contour Mode Axes1, 2, & 3
1CP1,1MA15,2MA15,3MA5           ; Contour Move (CP1=
                                ; Linear Interpolation) Absolute Move
                                ; to Axis 1=15.0", 2=15.0", 3=5.0"
```




*3-dimensional Linear Interpolated move*




*Velocity Profile for three axes Linear Interpolated move*

**MCAPI using "basic"**

```
Contour.VectorVelocity = 10.0           ' set contour velocity
Contour.VectorAccel = 50.0              ' set contour acceleration
Contour.VectorDecel = 50.0              ' set contour deceleration
Call MCSetContourConfig( 1, Contour)    ' initialize axis with contour
                                        ' settings
Call MCSetOperatingMode( 1, 1, CONTOUR ) ' axis one is in control
Call MCSetOperatingMode( 2, 1, CONTOUR ) ' axis two linked to axis one
                                        ' motion
Call MCSetOperatingMode( 3, 1, CONTOUR ) ' axis three linked to
                                        ' axis two
Call MCContourPath( 1,  LINEAR, " 1MA15,2MA15,3MA5" )
```

# Motion Control Integration and Diagnostic Tools

To assist the machine builder, powerful software tools are included with the Windows MCAPI. For the programmer, there is a wealth of commented sample source code. And our Motion Integrator™ suite of setup, tuning and diagnostic programs will help you get your system up and running in no time.

*We continuously develop new software tools. Please consult our factory for the latest available software.*

www.pmccorp.com

PMC Motion Control
Precision MicroControl Corp.

For Robotics & Machine Automation


2075-N Corte del Nogal
Carlsbad, CA 92009, USA
**Tel 760-930-0101** • Fax 760-930-0222

Information: info@pmccorp.com
Sales: sales@pmccorp.com
Tech Support: support@pmccorp.com
Web: www.pmccorp.com