Elmo
Motion Control

Distributed Multi-Axis Solution

Maestro - Multi-Axis Manager

Elmo's *Maestro* is a **network-based multi-axis motion manager** that operates in conjunction with Elmo intelligent servo drives to provide a full multi-axis motion control solution. The *Maestro* and the *SimplIQ* servo drives share the motion processing workload in a distributed motion control architecture.

The *Maestro* operates as a **Multi-Axis Motion Manager** to:
- coordinate motion between various axes in synchronized interpolated mode
- integrate event handling into motion control procedures

The *Maestro* operates as a **CANopen Network Node Master** for:
- Network management (NMT)
- Heartbeat
- Clock synchronization
- Network Configuration

The *Maestro* operates as a **Ethernet - CAN gateway**

The *Maestro* operates as a File Administrtator for:
- Firmware – Maestro and intelligent drives
- Multi-Axis User Applications – Maestro and intelligent drives
- System Resources

The *Maestro* operates as a **Multi-Axis Motion Analysis & Development tool**:
- Multi-Axis recording and analysis tools
- Multi-Axis application development environment

The *Maestro* can be monitored through a web browser

Hardware	Stand-alone
Operating System	Real Time OS with Elmo's *SimplIQ* Kernel
Programming	• Elmo *SimplIQ* extended language • Elmo Studio (IDE) • IEC 61131-3 (mid-2005)
Debugging	On-line monitoring of all axes state variables, get/set values, controller values, on-line axis tuning, axis variables forcing, recording
Number of Axes	• Up to 126 axes per CAN bus • 2 ~ 16 Interpolated axes depending upon bus load
Axis Types	• Intelligent servo drives (Elmo) and/or • CANopen DSP-402
Execution Time (multi-thread)	• Interpolated points: 1 mSec *see graphs on last page*
Axis Functions	Standard: start/stop/reset/ reference, velocity, position Special: master-slave cascading, electronic gearings, interpolated position

Protocols	• CANopen: DS 301, DSP 401 , DSP 402 • Telnet, HTTP
I/O System	• On board isolated 16 DI, 16 DO, 4 AI • CANopen DS 401 • Incremental encoder 20 Mega-Counts/ Second
Connectivity	Variable access via *Maestro* API
Master	• Heartbeat • NMT Synchronization
Interface	• RS-232 • Fast Ethernet (10/100 Mbps) • 2 CAN bus
Power Supply	24 VDC
Processor	• 300 MHz, Pentium compatible • 16 bit ISA (PC104 standard)
Memory	• 64 MB Internal flash (expandable to 128 MB) • 64 MB RAM (expandable to 128 MB)
Diagnostic LEDs	power, LAN speed, LAN activity, Flash access
Dimensions	153 mm x 107 mm x 51 mm
Weight	400 grams (14 ounces)
Oper. Temp.	0°C ~ 40°C

Distributed Network Architecture for Multi-Axis Applications

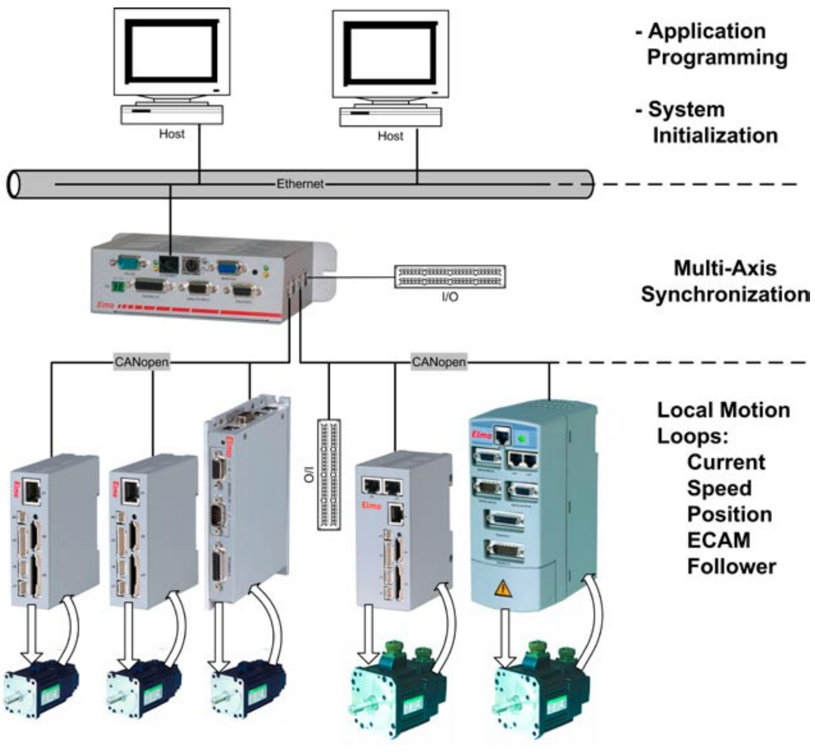
Why Maestro ?

Lower system cost with network communications:
· less wiring · less space · less noise

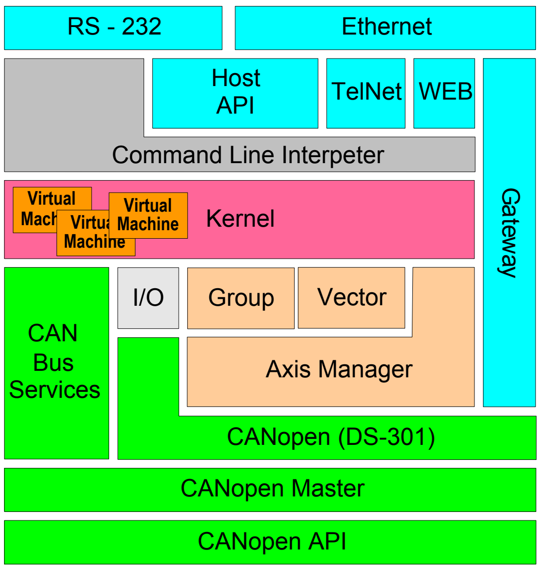
Lower hardware cost because real-time motion management tasks are off-loaded from the host

Lower development cost due to:
- · transparent CANopen communications
- · comprehensive integrated development environment (IDE) and testing tools
- · flexible integration of intelligent nodes (Drives, I/O, Encoders, HMI's) using industry standard protocols

Lower maintenace costs due to:
- · remote drive and motor diagnostics
- · less hardware to maintain
- · centralized firmware, application and resource handling
- · short machine setup



Functional Block Diagram



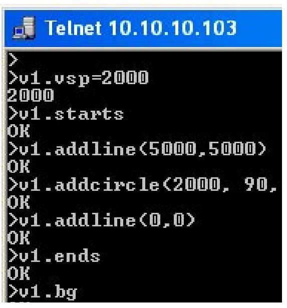
The functions in the diagram are organized as follows:

- **Host Communications Services -**
  provides standard communications services
- **Command Line Interpreter -**
  enables line-by-line program execution
- **The Kernel -**
  executes programs individualy or simultaneously
- **Motion Management -**
  motion is defined relative to:
  Axes, **Vectors** and **Groups**
- **CANopen Communication Services -**
  provides standard CANopen NMT services

Command Line Interpreter (CLI)

CLI commands that are sent to the *Maestro* are either executed by the *Maestro* itself or are forwarded directly to the specified axis for immediate execution. The CLI currently supports the following commands:

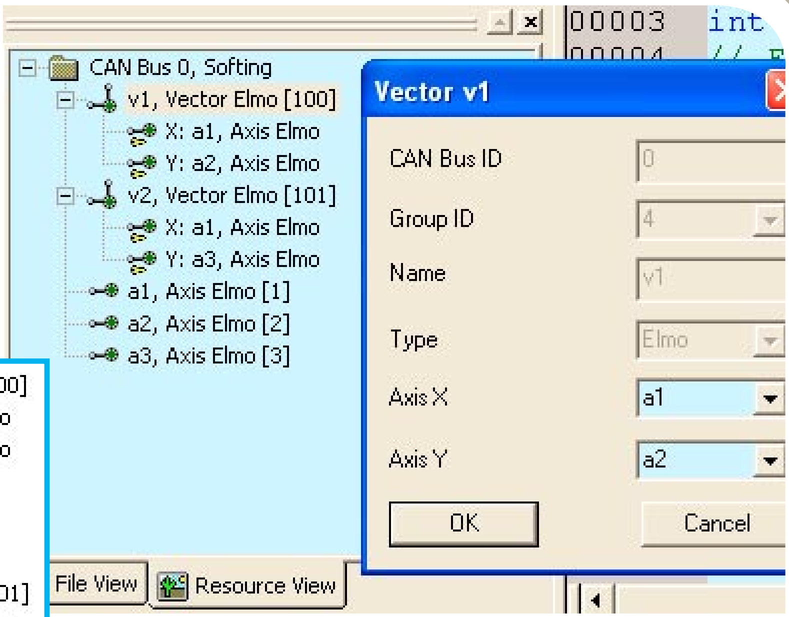
- Initialization commands
- Commands for collecting information
- Axis commands
- Vector commands
- Group commands


Telnet 10.10.10.103

```
>
>v1.vsp=2000
2000
>v1.starts
OK
>v1.addline(5000,5000)
OK
>v1.addcircle(2000, 90,
OK
>v1.addline(0,0)
OK
>v1.ends
OK
>v1.bg
```

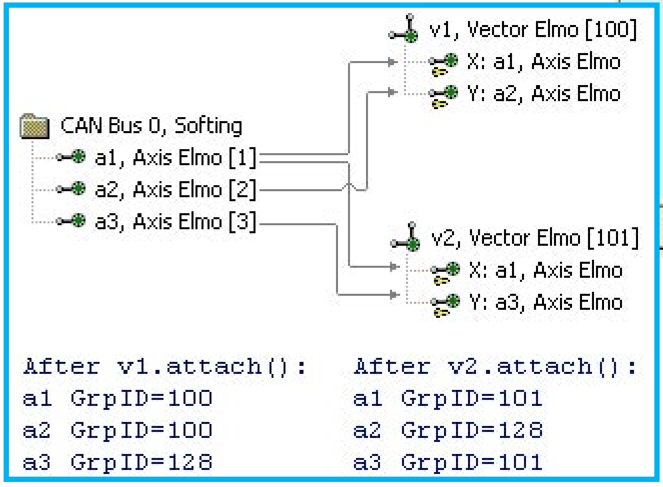
Designating Resources

The *Maestro* is designed to manage movement. Its "perspective" of the world is based upon motor axes which perform the movement either independently or in combination. Therefore, an application dveloper's first job is to define the elements that make up the *Maestro*'s "world". These elements are: **Axes**, **Vectors** and **Groups**.



```
CAN Bus 0, Softing
  v1, Vector Elmo [100]
    X: a1, Axis Elmo
    Y: a2, Axis Elmo
  v2, Vector Elmo [101]
    X: a1, Axis Elmo
    Y: a3, Axis Elmo
  a1, Axis Elmo [1]
  a2, Axis Elmo [2]
  a3, Axis Elmo [3]
```

Vector v1	
CAN Bus ID	0
Group ID	4
Name	v1
Type	Elmo
Axis X	a1
Axis Y	a2



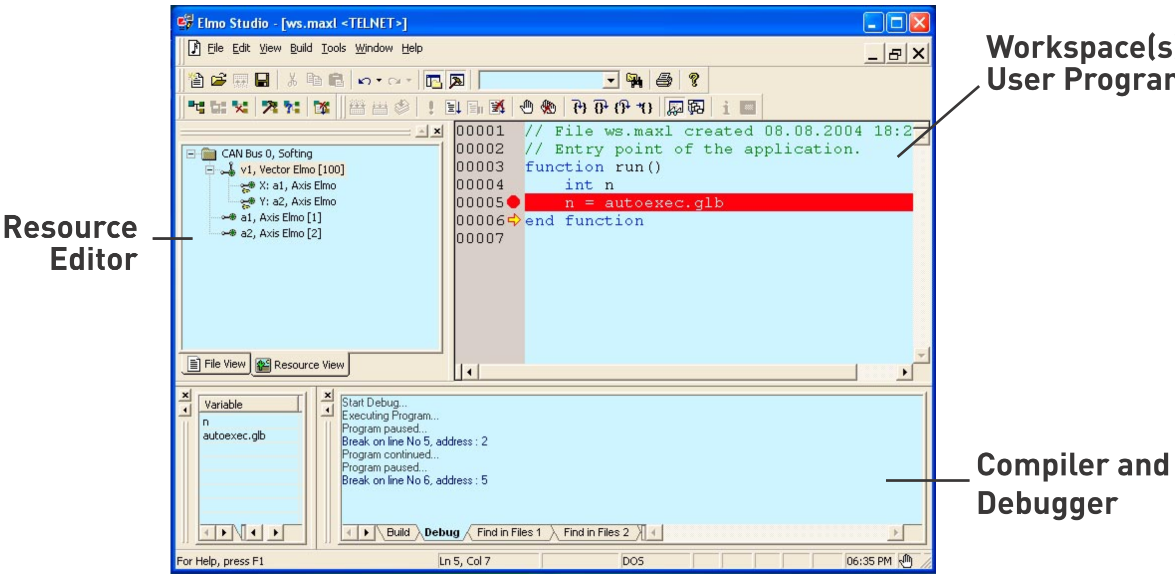
```
After v1.attach():    After v2.attach():
a1 GrpID=100          a1 GrpID=101
a2 GrpID=100          a2 GrpID=128
a3 GrpID=128          a3 GrpID=101
```

Resource:	Axes, groups and vectors make up a set of *Maestro* resources.
Axis:	Drive and motor pair that controls motion along a specific axis.
Vector:	A set of dependent axes designed to implement vector motion. The axes require accurate timing and strict synchronization.
Group:	A set of axes that the *Maestro* can send the same command to simultaneously. Each axis can implement motion independently.

The *Maestro* uses **Primitives** to define all types of motion. *Maestro* primitives are **Lines**, **Circles** and **Poly-lines**. Combined, these primitives can form even the most complex paths. When the *Maestro* is in control, motion does not stop at each primitive's boundary, instead motion remains continuous.

Maestro's User Programming Language (UPL)

The UPL is a compiled language, designed for real-time execution of motion-control tasks. The user writes a program in an IDE (Integrated Development Environment) and then compiles the program. A successfully compiled program can then be downloaded to the *Maestro*, from where it can be edited and debugged using the (good old) Elmo Studio.



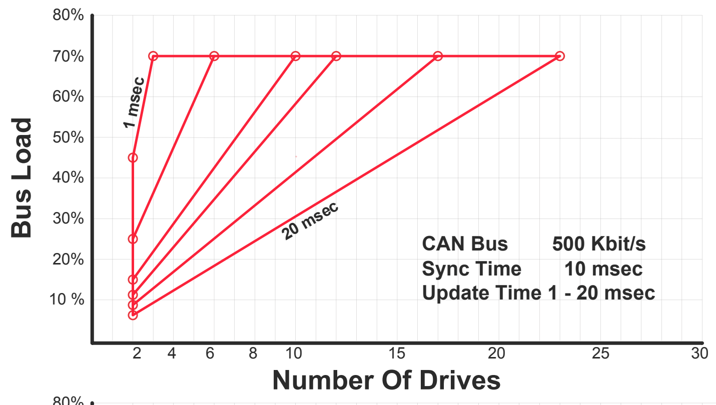
```
00001 // File ws.maxl created 08.08.2004 18:2
00002 // Entry point of the application.
00003 function run()
00004     int n
00005     n = autoexec.glb
00006 end function
00007
```

```
n
autoexec.glb
```

```
Start Debug...
Executing Program...
Program paused...
Break on line No 5, address : 2
Program continued...
Program paused...
Break on line No 6, address : 5
```

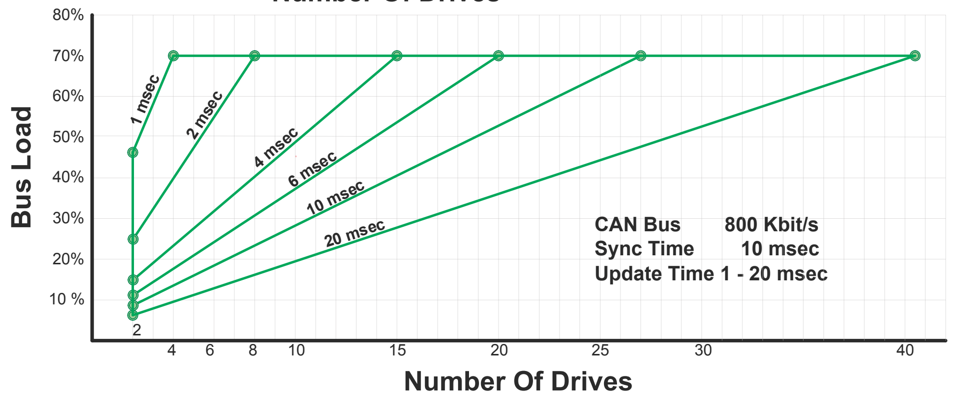
Elmo Studio (Integrated Development Environment) for Single and Multi-Axis Application Development

Network Performance in Synchronized Axis Mode



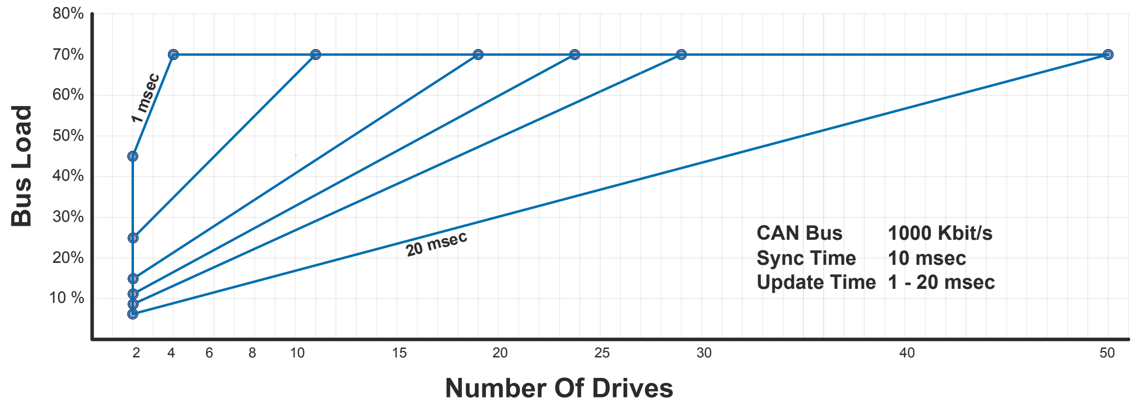
The *Maestro* can manage up to 126 axes on a standard application. However, fewer axes can be managed on applications requiring fast updates.

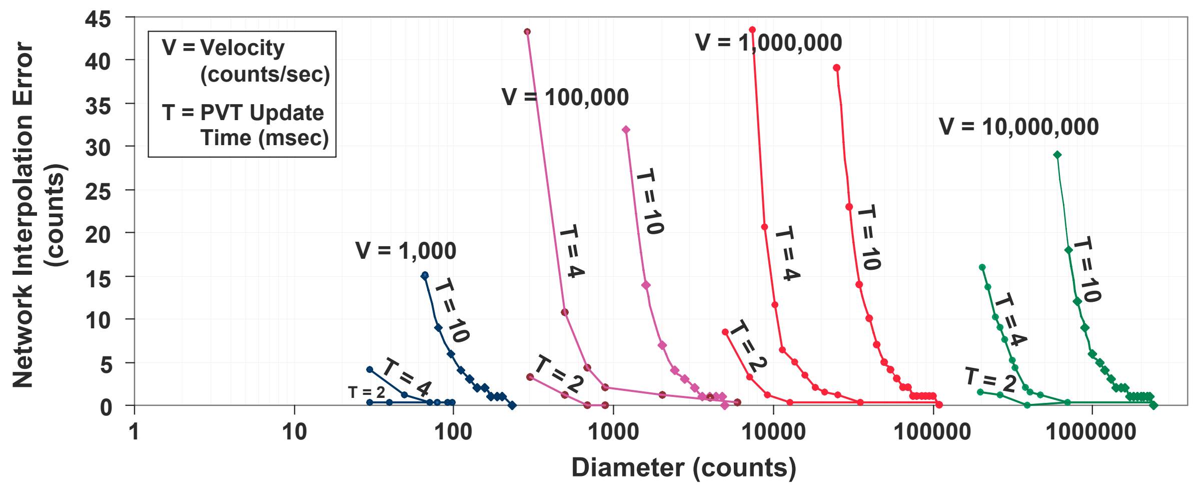
The Bus Load graphs show how many drives can be managed as a function of update time and bus speed.



For example, the *Maestro* can manage 4 axes on an application that requires an update every 1 msec running at a 70% load on an 800 Kbit/s bus.

On the other hand, the *Maestro* can manage up to 50 axes on an application that requires an update every 20 msec running at a 70% load on an 1000 Kbit/s bus.





The graph above can be used as a yardstick to determine the number of updates needed to produce an acceptable path. In this case, the number of updates is a function of the diameter, speed and maximum allowable interpolation error.

For example, if the required velocity is 100,000 counts/sec along a 1000 count diameter path, and the allowable interpolation error is 10 counts, then the *Maestro* must perform an update every 4 msec.

Elmo Motion Control Inc.
1 Park Drive, Suite 12
Westford, MA 01886
USA
Tel: (978) 399-0034
Fax: (978) 399-0035

Elmo Motion Control GmbH
Steinbeisstrasse 41
D-78056, Villingen-Schwenningen
Germany
Tel: +49 (07720) 8577-60
Fax: +49 (07720) 8577-70

For updates about the *Maestro* visit our web site:
www.elmomc.com