



Copley Motion Libraries

Copley Motion Libraries (CML) combine with Copley CANopen servo and stepper drives for rapid development of a multi-axis control system. CML links into your C++ application program running on a PC or embedded processor. Trajectory generation, network management and CANopen message formatting are taken care of automatically.

Key Features

- Supports CANopen servo and stepper drives
- C++ source code for PCs or embedded systems
- Network management is automatic
- Point-to-point and coordinated motion

Motion System Software

User's C++ Motion Control
Application Program plus
Copley Motion Library (CML)

Compiled by
Microsoft Visual ++ or
Gnu C++ Compilers

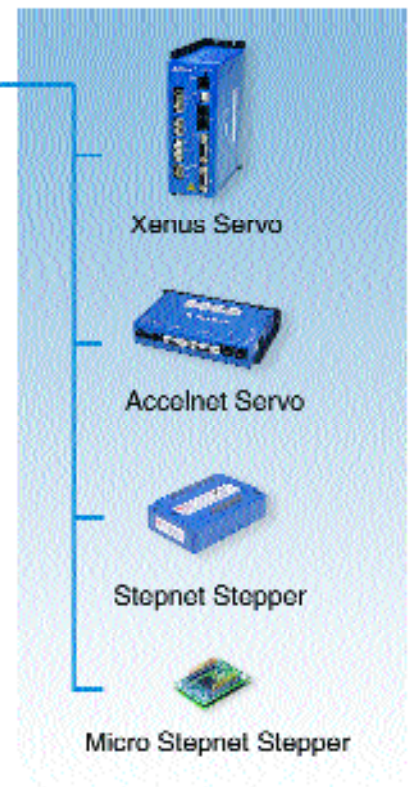
Microsoft Windows or
Linux

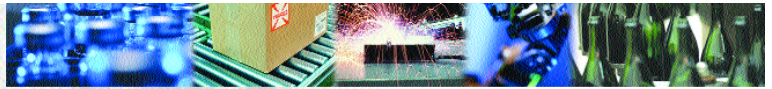
CAN Interface Driver

Controller Hardware



Distributed Drives





COPLEY MOTION LIBRARIES (CML)

RAPID DEVELOPMENT

Copley Motion Libraries (CML) are a collection of object-oriented C++ programs designed to simplify the integration of intelligent CANOpen amplifiers into a PC based or embedded control architecture. The development of low-level code to control the network is eliminated. CAN card interfacing, PDO mapping, SDO data packing, synchronization and node guarding are taken care of automatically by a few simple commands.

The application programmer has direct access to CANopen DS-402 compliant motion functions (enable, homing, get and set parameters, single axis moves). For multi-axis control, a coordinated set can

be created by the linkage class. Simply pass position coordinates and CML generates a stream of PVT (position/velocity/time) points. The drives buffer the points, perform a cubic polynomial interpolation and synchronously update commanded position to generate the path through space.

CML is designed to enable the C++ programmer to create motion applications on a range of operating systems and processor boards. It can be used on a PC using the Microsoft Windows or Linux operating systems. CML can also run on an embedded processor with any real-time operating system.

COMMAND LIBRARY SUMMARY

- Network configuration and startup
- Generation of CANopen messages (SDO, PDO etc.)
- PDO configuration and mapping to drive objects
- Sync message and clock synchronization
- Heartbeat configuration and activation
- Node guarding configuration and activation
- Network error management
- Set and get parameters
- Send profile parameters
- Execute profiles
- Send PVT vectors and manage PVT buffer
- Handle faults and errors
- Configure and execute homing
- Download firmware
- Transfer motor files
- Transfer amplifier data structure (model and serial number, power ratings etc.)
- Read and write CANopen I/O modules