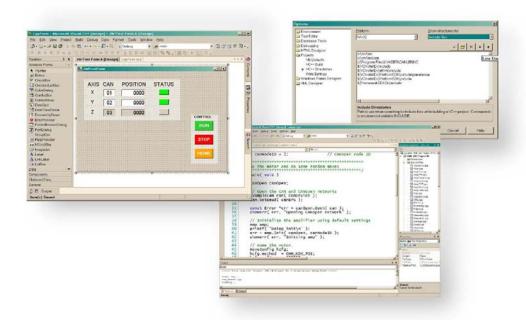


C++ Software Tools for CANopen Distributed Control



- Automatic Network Management of CAN Bus
- Point-to-Point and Coordinated Motion

CML is a collection of object-oriented C++ classes designed to simplify the integration of intelligent CANopen servo amplifiers and stepping motor drivers into a PC based, or embedded control architecture. The development of low-level code to control the network is eliminated. Communication-card interfacing, mapping PDO's, SDO data packing, synchronization and nodeguarding 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/Set parameters, & single-axis moves). For a multi-axis control, a coordinated set can be created by the linkage class.

- C++ Source Code for PC's or Embedded Systems
- Supports CANopen Servo & Stepper Drives

Complex, multi-axis moves are possible when the application program generates a sequence of points that define position, velocity, and time (PVT). The drives buffer the PVT points, perform a cubic polynomial interpolation algorithm and synchronously update commanded position to generate the path through N-dimensional 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 realtime operating system.







Description

WHAT IS CML?

CML is a collection of C++ classes that, when compiled in a C++ application, provide a programming interface to Copley Controls CANopen-enabled digital servo amplifiers and stepping motor drivers.

WHY USE CML?

CML adds CANopen functionality to C++ applications and eliminate the low-level coding to support communications over a CAN bus. In addition, CML eliminates the additional coding needed to support communication with devices operating under the CANopen protocol, the application layer that works over a CAN bus that is designed for motion control and other specialized types. CANopen devices have object-dictionaries that combine dedicated addresses for standard functions and other addresses for device-specific ones.

CML provides a high-level language interface to low-level functions that is efficient and robust. This greatly reduces development time and time-to-market. At the same time, it enables programmers to focus on their application development and to treat the CANopen interface simply as a library of objects that are ready to use.

HOW DOES IT WORK?

CML provides an object-oriented interface during program development. CML communicates with a CAN interface card via the interface driver provided by the manufacturer. At run-time, CML provides CAN bus control of Copley CANopen products by managing all of the low-level bus communications necessary to provide those services.

GENERAL SPECIFICATIONS

PRODUCT TYPE

C++ class source-code files

OPERATING SYSTEMS SUPPORTED

Microsoft Windows, Linux, or other POSIX compliant operating system

CANopen COMPLIANCE

CiA DSP-402, CANopen Device Profile for Drives & Motion Control CiA DS-401, CANopen Device Profile for Generic I/O Modules CiA 301, Application Layer and Communication Profile

HARDWARE REQUIREMENTS (MINIMUM)

Processor running POSIX compliant software

One CAN interface device

Copley Controls CANopen servo amplifier or stepping motor driver: Xenus, Accelnet, Stepnet

SOFTWARE REQUIREMENTS

Any C++ compiler for use with CML. Some examples are:

Microsoft Visual C++, C++ .NET, gcc

CME 2 Version 3.2 or higher; Copley Controls' application for

amplifier setup, tuning, and configuration

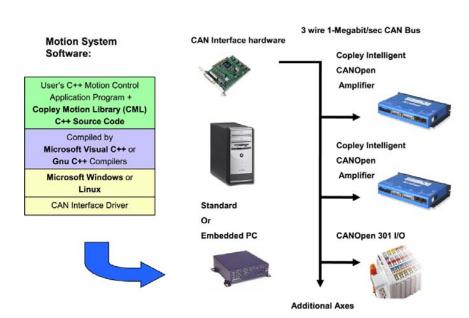
CANopen HARDWARE SUPPORTED

CAN bus interface products:

Kvaser, Ixxat, Vector, National Instruments

I/O products:

CML SYSTEM CONFIGURATION









Commonly Used Classes and their Member Functions and Variables

CARlogen Object (handles all CANopen Mouton DSP-402 standard CAN communications between the PC and the amplifiers) Profitime Any Profitime Any Profitime Earliant Institute Institute	General Function	Method/Property Name	Description	
EARlogen EARlogen Entitlative Institution Ampilitier Object (findindes all control and monitoring communications with an ampilitier using CANoppen metwork communications Ampilitier Indisalization Initiatize Disable Disable Disable the ampilitier Entitle Entitle Entitle Entitle Disable Disable the ampilitier Clear any latering fautile on the CANopen network with the appropriate CAN address and state communications Entitle Entitle Entitle Clear any latering fautile on the ampilitier Clear any latering fautile on the ampilitier Read the position growth automatic clear Position and Velocity Position Actual Read the position growth automatic clear Position and Velocity Position Carlon Cod Nome Position Point Move Support MoveAbs Settings Cod Nome	CANopen Object (handles all CANopen Motion DSP-402 standard CAN communications between the PC and the amplifiers)			
Ampilifer Object (nandles all control and monitoring communications with an ampilifer using CAROpen Motion DSP-402 compatible objects) Ampilifer Initialization Initializa	CANopen	PortName	The CAN card name and port to be used for CANopen network	
Ampillier Object (inandies all control and monitoring communications with an ampillier using CANopen Motion 1987-402 compatible objects) Ampillier Indistization		BitRate	The CANopen Bit Rate to be used	
Ampilifor Initialization Initializat		Initialize	Initialize the CANopen network commulcations	
Amplitier Modes and Status Information Position and Velocity Position and associate position error (definence between position command and actual position and associate position point move using the pre-configured profile settings Position and position and associate point-to-point move using the pre-configured profile settings Profile Petern an absociate point-to-point move using the pre-configured profile settings Profile Petern and absociate point-to-point move using the pre-configured profile settings Profile Petern and associate point-to-point move using the pre-configured profile settings Profile Petern and position and passed pre-configured profile settings Profile Petern Processing Coestations Object (for setting up motion profile) Profile Settings Object (for setting up motion profile) Profile Settings Object (for performing coordinated multi-ass motion) Profile Settings Object (for performing coordinated multi-ass motion) Profile Settings Position profile profile profile pro	Amplifier Object (handles all control and monitoring communications with an amplifier using CANopen Motion DSP-402 compatible objects)			
Enable E	Amplifier Initialization	Initialize		
Amplifier Modes and Status Information ClearFaults Clear any latching faults on the amplifier ReadEventSticky Read events with automatic clear Position and Velocity Position Actual Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the position error (difference between position command and actual position) Read the actual encoder position Read the actual encoder position Read the actual encoder position Read the actual encoder position error position error position or an error to ocour. Read the actual encoder position error positi	Amplifier Modes and Status Information	Disable	Disable the amplifier	
Clear Faults Clear any latching faults on the ampilier		Enable	Enable the amplifier	
Position and Velocity Position and Section Section and Section Section Section and Section Section Section and Section Section Section and Section Secti		ClearFaults	Clear any latching faults on the amplifier	
Position and Velocity Position Error Read the position error (difference between position command and actual position) Position Posit		ReadEventSticky	Read events with automatic clear	
PositionError PositionError PositionError PositionError PositionError PositionError PositionError Position Position Position Position Position Position Position PositionError Position PositionError Position PositionError Position Position PositionError Position Position PositionError	Position and Velocity	PositionActual	Read the actual encoder position	
Companies Comp		PositionError		
Point-to-Point Move Support MoveAbs Perform an absolute point-to-point move using the pre-configured profile settings MoveRel Perform a relative point-to-point move using the pre-configured profile settings MoveRel Perform a relative point-to-point move using the pre-configured profile settings WaltMoveDone Walts for the currently running move to finish, or for an error to occur Create Event Create an event that monitors amplifier events for specific conditions Unit Conversion Functions CountsPertUnit Conversion Functions CountsPertUnit Coreate an event that monitors amplifier events for specific conditions Profile Settings Object (for setting up motion profile) Profile Settings Object (for setting up motion profile) Profile Settings Profile Cestises the profile type (SCurve, Trap, Velocity) Gets/sets profile velocity value (velocity that the motor attempts to reach during the move) Profile Acc Gets/sets profile acceleration value (acceleration that the motor uses when sating the move) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for monitoring events from a given amplifier) WaltMoveDone Walt unit the multi axis move is complete. If the move does not complete by the time specified, the function will return Perent Object (for monitoring events from a given amplifier) Event Object (for monitoring events from a given amplifier) CopleyMotionLibrary Object (high level object that enables sophisticated debugging) CopleyMotionCommunications with a CANopen DS-401 compatible I/O module) Dipidal I/O DinSRead, DoutSWrife Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O AnfieRead, Ain16Wrife Read a group of 8 digital inputs or write a 16-bit analog output	Homing	GoHome	Go home executes the homing routine as specified in the home settings object	
Point-to-Point Move Support MoveRel Amplifier Event Processing WaitMoveDone Waits for the currently running move to finish, or for an error to occur Create Event Create an event that monitors amplifier events for specific conditions Unit Conversion Functions Counts PerUnit Counts PerUnit Create an event that monitors amplifier events for specific conditions Stores a scaling factor for converting between an amplifier's default units (encoder counts) and user-defined units Profile Settings Object (for setting up motion profile) Profile Settings Profile Type Profile Yel Cets/sets the profile yelo (Scurve, Trap, Velocity) Cets/sets the profile velocity value (velocity that the motor attempts to reach during the move) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing events from a given amplifier) WaitMoveDone Wait until the multi axis move is complete. If the move does not complete by the time specified, the function will return Start Stop Stop monitoring Debug logging Debug Level Debug Level Debug Level Debug Level Debug Level Din8Read, DoutsWirte Read a group of 8 digital inputs or write a 16-bit analog output	Quick Stop Support	HaltMove	Halts current move using pre-programmed halt mode	
MoveRel Perform a relative point-to-point move using the pre-configured profile settings Settings	Point-to-Point Move Support	MoveAbs		
Ampillier Event Processing CreateEvent Create an event that monitors ampillier events for specific conditions Unit Conversion Functions CountsPerUnit Stores a scaling factor for conventing between an ampillier's default units (encoder counts) and user-defined units Profile Settings Object (for setting up motion profile) Profile Settings Profile Ype Gets/sets the profile type (SCurve, Trap, Velocity) Profile Settings Profile Vel Profile Acc Gets/sets profile velocity value (velocity that the motor attempts to reach during the move) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing coordinated multi-axis motion) Linkage Object Methods MoveTo Multi-axis move. Target positions are passed as an array. Wait until the multi axis move is complete. If the move does not complete by the time specified, the function will return Event Object (for monitoring events from a given amplifier) Event Object (high level object that enables sophisticated debugging) Debug logging Debug Level Debug Level Gets/sets the profile accoleration value (acceleration that the motor uses when starting the move) Begins monitoring for an event to cour. Generates a single callback to a user subroutine when the chosen event occurs, or timeout has expired Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs 10 Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O DineRead, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Read a 16-bit analog input or write a 16-bit analog output		MoveRel		
Unit Conversion Functions CountsPerUnit Conversion Functions CountsPerUnit Conversion Functions CountsPerUnit Conversion Functions Profile Settings Object (for setting up motional profile) Profile Settings Object (for setting up motional profile) Profile Settings Profile Settings Profile Settings Profile Settings Profile Settings Profile Settings Profile Vel Gets/sets the profile type (SCurve, Trap, Velocity) Gets/sets profile velocity value (velocity that the motor attempts to reach during the move) Profile Acc Gets/sets profile acceleration value (acceleration that the motor uses when starting the move) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for menitoring coordinated multi-axis motion) Linkage Object (for monitoring coordinated multi-axis motion) Event Object (for monitoring events from a given amplifier) Event Object (for monitoring events from a given amplifier) Event Object (for monitoring events from a given amplifier) Event Object (for monitoring events from a given amplifier) Event Object (high level object that enables sophisticated debugging) Debug logging Debug Level Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs Coplex (Controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Analog I/O Ain16Read, Ain16Write Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O Ain16Read, Ain16Write		WaitMoveDone	Waits for the currently running move to finish, or for an error to occur	
Profile Settings Object (for setting up motion profile) Profile Settings Object (for setting up motion profile) Profile Settings Profile Settings Object (for setting up motion profile) Profile Settings Object (for performing coordinated multi-axis motion) Linkage Object (for monitoring events from a given amplifier) Event Object (for monitoring events from a given amplifier) Start Begins monitoring for an event to occur. Generates a single callback to a user subroutine when the chosen event occurs, or timeout has expired subroutine when the chosen event occurs, or timeout has expired from and to all the amplifiers along with actions and faults that are useful in debugging monitor programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Ain16Read, Ain16Write Read a group of 8 digital inputs or write a 16-bit analog output	Amplifier Event Processing	CreateEvent	Create an event that monitors amplifier events for specific conditions	
Profile Settings Profile Vel Velocity value (velocity that the motor attempts to reach during he move) Profile Vel Profile Veloceteration value (acceleration value (acceler	Unit Conversion Functions	CountsPerUnit		
Profile Settings Profile Vel Gets/sets profile velocity value (velocity that the motor attempts to reach during the move) ProfileAcc Gets/sets the profile acceleration value (acceleration that the motor uses when starting the move) Linkage Object (for performing coordinated multi-axis motion) Linkage Object Methods	Profile Settings Object (for setting up motion profile)			
Profile Settings Profile Acc Gets/sets the profile acceleration value (acceleration that the motor uses when starting the move) Linkage Object (for performing coordinated multi-axis motion) Linkage Object Methods Initialize Initializes a linkage object by assigning Amplifiers to it MoveTo Multi-axis move. Target positions are passed as an array. Wait until the multi axis move is complete. If the move does not complete by the time specified, the function will return Event Object (for monitoring events from a given amplifier) Event Object Start Stop Stop monitoring CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging Debug Level Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Analog I	Profile Settings	ProfileType	Gets/sets the profile type (SCurve, Trap, Velocity)	
Linkage Object (for performing coordinated multi-axis motion) Linkage Object (for performing coordinated multi-axis motion) Linkage Object Methods Initialize Initializes a linkage object by assigning Amplifiers to it MoveTo Multi-axis move. Target positions are passed as an array. Wait until the multi axis move is complete. If the move does not complete by the time specified, the function will return Event Object (for monitoring events from a given amplifier) Event Object Start Begins monitoring for an event to occur. Generates a single callback to a user subroutine when the chosen event occurs, or timeout has expired Stop Stop monitoring CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs 10 Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O DinBRead, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Read a 16-bit analog input or write a 16-bit analog output		ProfileVel		
Initialize Initialize Initializes a linkage object by assigning Amplifiers to it MoveTo Multi-axis move. Target positions are passed as an array. Wait until the multi axis move is complete. If the move does not complete by the time specified, the function will return Event Object (for monitoring events from a given amplifier) Event Object Start Begins monitoring for an event to occur. Generates a single callback to a user subroutine when the chosen event occurs, or timeout has expired Stop Stop monitoring CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O Ain16Read, Ain16Write Read a 16-bit analog input or write a 16-bit analog output		ProfileAcc		
MoveTo Multi-axis move. Target positions are passed as an array.	Linkage Object (for performing coordinated multi-axis motion)			
WaitMoveDone Wait until the multi axis move is complete. If the move does not complete by the time specified, the function will return Event Object (for monitoring events from a given amplifier) Start Start Begins monitoring for an event to occur. Generates a single callback to a user subroutine when the chosen event occurs, or timeout has expired Stop Stop monitoring CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages for debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Read a 16-bit analog input or write a 16-bit analog output	Linkage Object Methods	Initialize	Initializes a linkage object by assigning Amplifiers to it	
Event Object (for monitoring events from a given amplifier) Event Object Start Start Stop Stop Stop monitoring CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Analog I/O Ain16Read, Ain16Write the time specified, the function will return Begins monitoring for an event to occur. Generates a single callback to a user subroutine when the chosen event occurs, or timeout has expired Stop Stop monitoring Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs Read a group of 8 digital inputs or write a group of 8 digital outputs Read a 16-bit analog input or write a 16-bit analog output		MoveTo	Multi-axis move. Target positions are passed as an array.	
Event Object Start Start Stop Stop monitoring for an event to occur. Generates a single callback to a user subroutine when the chosen event occurs, or timeout has expired Stop Stop monitoring CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Read a 16-bit analog input or write a 16-bit analog output		WaitMoveDone		
Event Object Stop Stop monitoring CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O Read a 16-bit analog input or write a 16-bit analog output	Event Object (for monitoring events from a given amplifier)			
CopleyMotionLibrary Object (high level object that enables sophisticated debugging) Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O Ain16Read, Ain16Write Read a 16-bit analog input or write a 16-bit analog output	Event Object	Start		
Debug logging DebugLevel Gets/sets the debug message level. When enabled will record CAN messages from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O Ain16Read, Ain16Write Read a 16-bit analog input or write a 16-bit analog output		Stop	Stop monitoring	
Debug logging DebugLevel from and to all the amplifiers along with actions and faults that are useful in debugging motion programs IO Object (controls communications with a CANopen DS-401 compatible I/O module) Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O Ain16Read, Ain16Write Read a 16-bit analog input or write a 16-bit analog output	CopleyMotionLibrary Object (high level object that enables sophisticated debugging)			
Digital I/O Din8Read, Dout8Write Read a group of 8 digital inputs or write a group of 8 digital outputs Analog I/O Ain16Read, Ain16Write Read a 16-bit analog input or write a 16-bit analog output	Debug logging	DebugLevel	from and to all the amplifiers along with actions and faults that are useful in	
Analog I/O Ain16Read, Ain16Write Read a 16-bit analog input or write a 16-bit analog output	IO Object (controls communications with a CANopen DS-401 compatible I/O module)			
	Digital I/O	Din8Read, Dout8Write	Read a group of 8 digital inputs or write a group of 8 digital outputs	
Event driven I/O ICreateEvent Create an event that monitors I/O events for specific conditions	Analog I/O	Ain16Read, Ain16Write	Read a 16-bit analog input or write a 16-bit analog output	
	Event driven I/O	ICreateEvent	Create an event that monitors I/O events for specific conditions	

Copley Controls, 20 Dan Road, Canton, MA 02021, USA Web: http://www.copleycontrols.com

Tel: 781-828-8090 Fax: 781-828-6547

Page 3 of 4





CML Applications

Laboratory Automation



Metrology



Conformal Coating



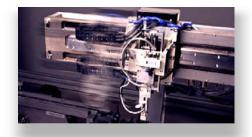
ActiveX, Microsoft, Visual Basic, Visual C++, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. LabVIEW is a registered trademark of National Instruments Corporation. Other product and company names mentioned herein may be the trademarks of their respective owners.



Textile Cutting



Pick & Place



Rev 3.01_tu 11/29/2011