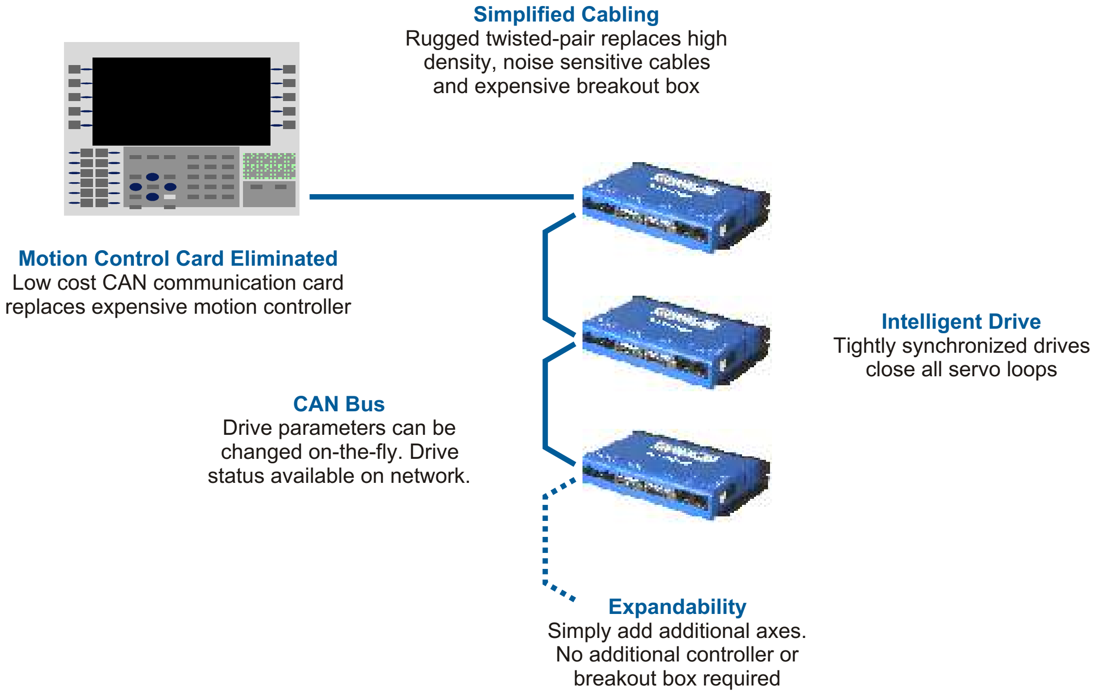
Copley Controls Corp.

Copley Motion Systems LLC

CANopen Distributed Control

The Distributed Control Advantage



Simplified Cabling
Rugged twisted-pair replaces high density, noise sensitive cables and expensive breakout box

Motion Control Card Eliminated
Low cost CAN communication card replaces expensive motion controller

Intelligent Drive
Tightly synchronized drives close all servo loops

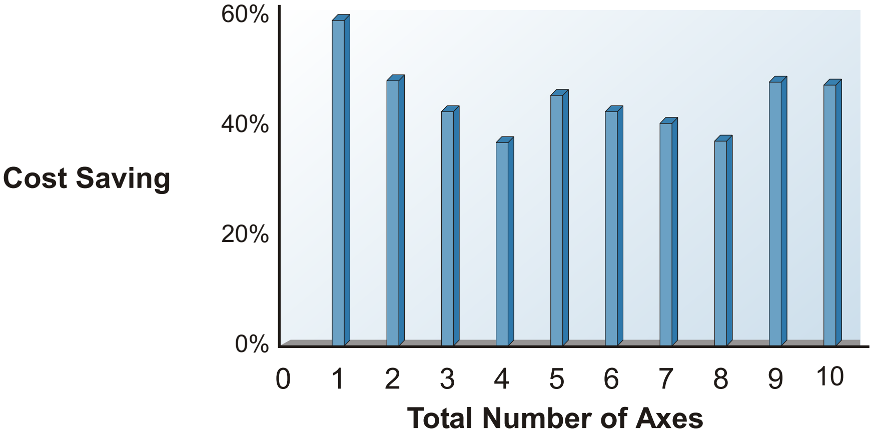
CAN Bus
Drive parameters can be changed on-the-fly. Drive status available on network.

Expandability
Simply add additional axes. No additional controller or breakout box required

CANopen distributed control solves the major problems of multi-axis motion: cost, cabling and startup-time. Intelligent drives close all servo loops and interface to motor feedback and local I/O. A single, low cost, twisted-pair cable connects the drives. Noise sensitive analog control signals are eliminated along with expensive cable harnesses and breakout boxes.

A low-cost CAN communication card in the Central Processing Unit (CPU) replaces the traditional dedicated motion controller. Software development is simplified. System cost is reduced and reliability, serviceability and expansion flexibility are improved. Drive parameters can be changed on-the-fly and complete system status is available on the network.

CANopen Savings versus Traditional Control



CAN and CANopen

CANopen is an Application Layer built on the lower level CAN Data Link Layer. CANopen is an open, international standard for motion. It defines the messages used to control and configure the network and devices on the network:

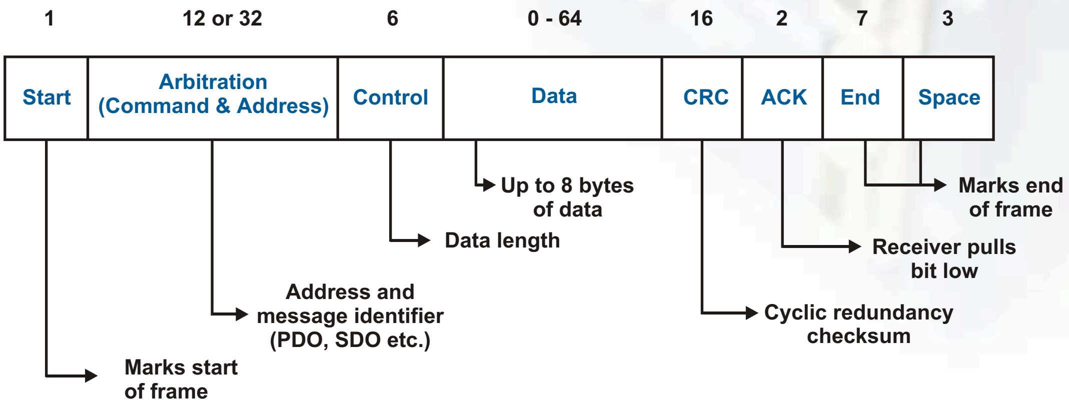
- How the network is managed - startup, network status
- Device configuration - parameter setting
- Device synchronization
- Real-time data transfer
- Event handling

The ability to tightly synchronize drives makes CANopen an excellent choice for high performance motion control. CANopen provides flexibility in how multiple axes are controlled. Drives on the network can operate in one of two position modes:

Profile Mode
For point-to-point moves
CPU sends move distance, velocity, accel/decel
Moves in multiple drives can be executed simultaneously

PVT Mode
For contouring
CPU sends series of points defined by position/velocity/time
Drive uses cubic polynomial interpolation for best-fit curve

CAN, the Data Link Layer, defines the low-level structure of data packets, device addressing, error checking and network arbitration. CAN was originally developed for automotive applications and is proven to be rugged and reliable in hostile environments. Data is transmitted over the network in the CAN frame shown below.



CAN accommodates master/slave communication and also peer-to-peer. CAN is deterministic and collisions are non-destructive. If two devices on the network initiate communication at the same time, the higher priority message is uncorrupted and goes through. The device with the lower priority message stops transmitting and tries again later.

SDOs and PDOs

The majority of network communication is handled by Service Data Object messages (SDOs) and Process Data Object messages (PDOs). SDOs are master/slave communications used primarily for drive configuration. PDOs are used for real-time control and data transfer. A PDO can be sent peer-to-peer or broadcast to all devices. Transmission can be synchronous, at programmed intervals, or triggered by an asynchronous event.

Both SDOs and PDOs access the "Object Dictionary" in a network device. Objects can be thought of as defining functions (e.g. execute move) that the device can perform or parameters (e.g. position loop gain, actual current) that the device can send or receive.

SDOs use part of the available 8 bytes in the CAN data field to address an object in a drive. If the object data is 4 bytes or less, only one CAN frame is required. If the data is longer, it can be sent in chunks of 7 bytes in subsequent frames. Each frame sent by the master (controller) must be acknowledged by a message from the slave (drive). An SDO requires a minimum transmission of 2 CAN frames.

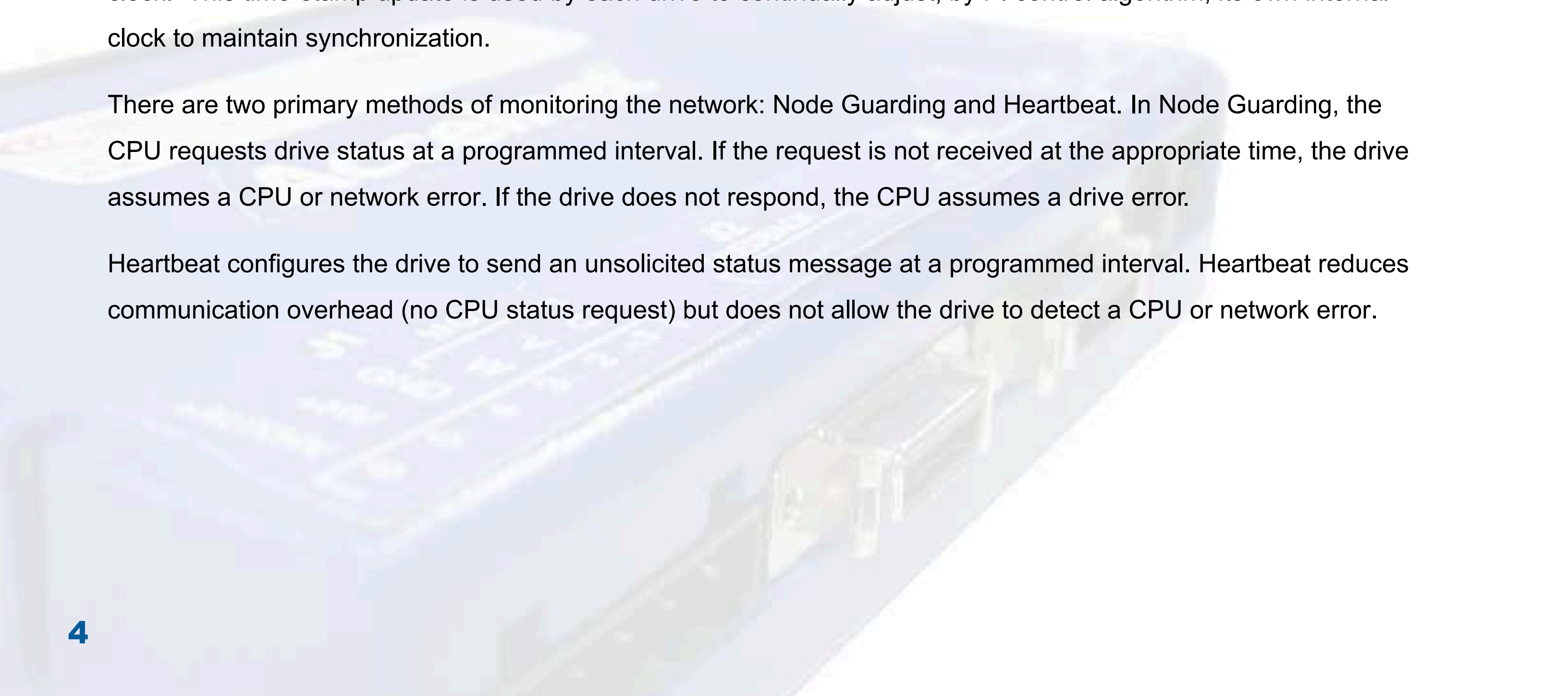
PDOs are an elegant solution for optimizing real-time transfer of data packets sized 8 bytes or less. A PDO can be mapped, ahead of time, to a specific drive object. All 8 bytes of the CAN data field are now available for data transfer. A PVT vector can be transmitted in one CAN frame with minimal communication overhead.

Network Synchronization and Management

Drive synchronization is a powerful feature of CANopen. At startup, one drive is configured to broadcast a periodic Sync Message, establishing the network timebase. CANopen additionally provides for internal clock synchronization. Periodically, the transmitter broadcasts the actual time of the last Sync Message as measured by that drive's internal clock. This time-stamp update is used by each drive to continually adjust, by PI control algorithm, its own internal clock to maintain synchronization.

There are two primary methods of monitoring the network: Node Guarding and Heartbeat. In Node Guarding, the CPU requests drive status at a programmed interval. If the request is not received at the appropriate time, the drive assumes a CPU or network error. If the drive does not respond, the CPU assumes a drive error.

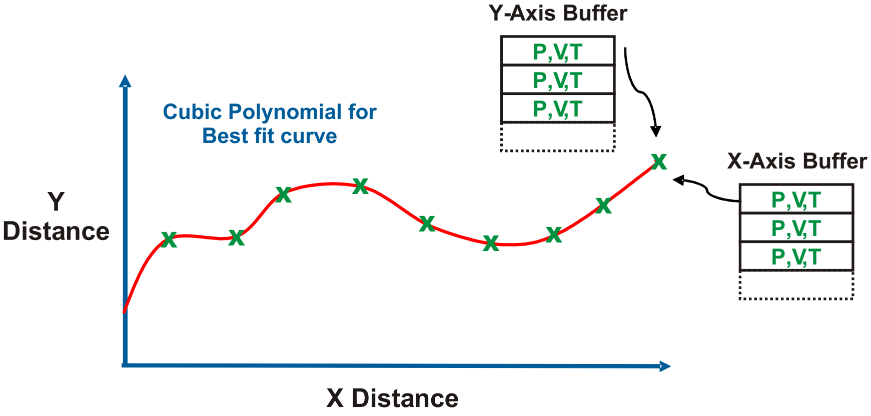
Heartbeat configures the drive to send an unsolicited status message at a programmed interval. Heartbeat reduces communication overhead (no CPU status request) but does not allow the drive to detect a CPU or network error.



Profile and PVT Mode

Profile Mode is typically used for point-to-point motion. The CPU sends the move distance, speed, acceleration and deceleration rate to the drive via SDOs then commands the move to be executed. Multi-axis coordination can be achieved by preloading moves and commanding simultaneous execution via broadcast PDO.

For applications where the coordinated path through space is critical, PVT Mode is preferred. The CPU generates a stream of points defining the position and velocity of each axis at specific times. Points are sent every 10 msecs (typical) and buffered in each drive. The drives perform a cubic polynomial interpolation to generate the best-fit curve, synchronously updating commanded position every 333 µsecs.



CANopen, Ethernet, Firewire and DeviceNET

CANopen makes highly efficient use of network bandwidth. Real-time data is transmitted with minimal overhead. Collisions are non-destructive for high priority messages. The ability of the drives to close the position loop and perform interpolation minimizes network traffic. 16 axes can be updated with a PVT point every 4 msecs, enabling the highest performance contouring.

The CAN Data Link Layer has a maximum speed of 1 M bit/sec on a 40 meter cable (derate for longer cables). A common misconception is that a faster bit rate makes a better bus. There are many trade-offs in bus performance. CAN is optimized to communicate the required data in the required time at the lowest cost.

CANopen is an international standard for motion control. Ethernet and Firewire are finding use in some applications but there is currently no standard implementation. Ethernet is non-deterministic. It can be used for drives operating in Profile Mode but is not appropriate for PVT Mode. Firewire typically networks drives operating in Current Mode. The CPU now closes the position and velocity loops and may incorporate specialized control algorithms.

DeviceNET is sometimes confused with CANopen. DeviceNET uses the CAN Data Link Layer but adds a proprietary Application Layer (developed by Rockwell). DeviceNET is less efficient for real-time communication and does not incorporate PVT Mode and clock synchronization.

Copley Motion Libraries and COM Objects

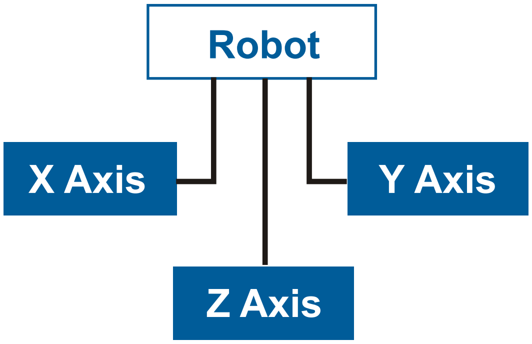
Copley Motion Libraries (CML) and COM Objects (CMO) make system commissioning fast and simple. The development of low-level code to control a CANopen network of intelligent drives is eliminated. No need to worry about communication card interfacing, mapping PDOs, setting up Sync Messages and Node Guarding or packing data into SDOs. All network housekeeping is taken care of automatically by a few simple commands linked into your application program.

Copley supports two levels of development environments. CML links into a C++ application program. CMO provides COM objects that can be used by Visual Basic, .NET, Labview or any program supporting the COM object interface.

CML delivers optimal flexibility and performance. The application program can run on a PC under Windows and Linux or on an embedded microprocessor with any real-time multi-tasking operating system. CMO provides optimal ease of use for applications where the inherent latency of the Windows operating system is not a factor. The chart below summarizes CML and CMO tradeoffs.

	Application	Hardware	Operating System	Advantages	Tradeoffs
CML	C++	PC Embedded microprocessor	Windows Linux Any real-time, multi-tasking OS	Performance Flexibility	C++ Expertise
CMO	Visual Basic .NET Labview	PC	Windows	Ease of Use	Windows latency Program execution speed

Coordinated Motion Made Easy



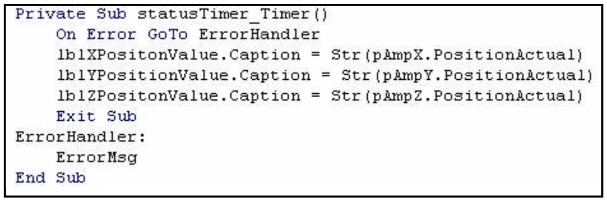
Coordinated motion is made easy with CML and CMO. A coordinated set can be created by grouping axes together. As an example, the coordinated set “Robot” can be commanded to move to x,y,z coordinates. **PVT generation for all axes is performed automatically.**

For most coordinated motion applications, CMO and Visual Basic will perform well despite the latency of Windows. The ability of the drive to perform cubic polynomial interpolation and buffer PVT points significantly reduces the real-time demand on Windows.

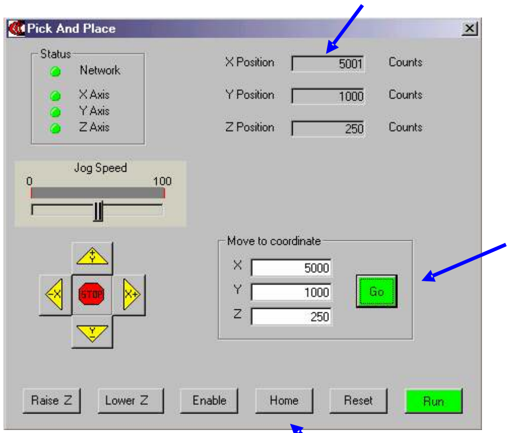
Visual Basic Example

An example of a simple pick-and-place XYZ robot GUI is shown below. The simplicity of the underlying Visual Basic code to aquire status and perform coordinated motion is highlighted. Commands (e.g. Home, CoordMoveAbs) are available from convenient pull-down menus.

Focus on your application. CANopen and PVT are taken care of.

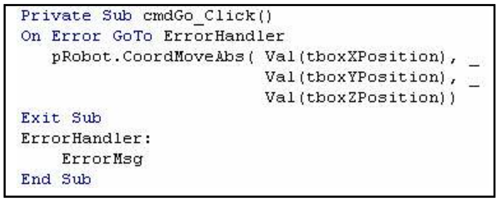


```
Private Sub statusTimer_Timer()
    On Error GoTo ErrorHandler
    lblXPositonValue.Caption = Str(pAmpX.PositionActual)
    lblYPositionValue.Caption = Str(pAmpY.PositionActual)
    lblZPositonValue.Caption = Str(pAmpZ.PositionActual)
    Exit Sub
ErrorHandler:
    ErrorMsg
End Sub
```

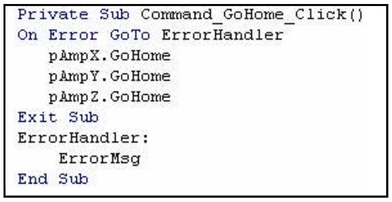


X Position 5001 Counts
Y Position 1000 Counts
Z Position 250 Counts

X 5000
Y 1000
Z 250



```
Private Sub cmdGo_Click()
On Error GoTo ErrorHandler
   pRobot.CoordMoveAbs( Val(tboxXPosition), _
                        Val(tboxYPosition), _
                        Val(tboxZPosition))
Exit Sub
ErrorHandler:
    ErrorMsg
End Sub
```



```
Private Sub Command_GoHome_Click()
On Error GoTo ErrorHandler
   pAmpX.GoHome
   pAmpY.GoHome
   pAmpZ.GoHome
Exit Sub
ErrorHandler:
    ErrorMsg
End Sub
```

Drives, Communication Cards and I/O

Copley Controls provides a comprehensive range of servo and stepper drives up to 5kW :

Xenus	Brushless/brush servodrive, 115/230 VAC operation
Accelnet	Brushless/brush servodrive, 20-180 VDC operation
Stepnet	Stepper drive, 20-75 VDC operation

CAN communication cards are available from numerous vendors including: Kvaser, Ixxat, Vector and National Instruments Corporation. General purpose I/O is available globally. WAGO is a leading supplier in North America.

Copley CANopen drives and software are fully tested with third party communication cards and I/O to deliver a complete solution. A successful transition to distributed control is guaranteed.

Copley Controls Corp.

Copley Motion Systems LLC

World Headquarters, USA

20 Dan Road
Canton, MA 02021
Tel: (+1) 781 828 8090
Fax: (+1) 781 828 1750
Sales Fax: (+1) 781 828 6547
E-mail: usasales@copleymotion.com

European Headquarters

Luckyn Lane
Pipps Hill
Basildon
Essex, UK
SS14 3BW
Tel: (+44) 1268 287 070
Fax: (+44) 1268 293 344
E-mail: sales@copleymotion.com

Regional Offices
West Coast, USA

California
Tel: (+1) 510 881 2064

Published - Feb 2003