A iCC 1994

1st international CAN Conference

in Mainz (Germany)

Sponsored by

Allen Bradley
National Semiconductor
Philips Semiconductors

Organized by

CAN in Automation (CiA)
international users and manufacturers group
Am Weichselgarten 26
D-91058 Erlangen
Phone +49-9131-69086-0
Fax +49-9131-69086-79
Email:headquarters@can-cia.de
URL: http://www.can-cia.de

A Servodrive Profile for CAN

John Doran, John Scannell, Michael Harrington

MOOG GmbH

Abstract

The trend towards distributed intelligence in automation applications is gathering pace. Many different bus systems are now emerging, aimed at the industrial automation market place. The Controller Area Network (CAN) offers several important features which can be exploited to give an optimum solution to many industrial applications.

This paper presents some of the work carried out with CAN-based servodrives over the past few years. With thousands of drives in the field, the advantages of CAN have been clearly demonstrated and the features of CAN have been exploited in full. The multi-master capability is used to implement electronic cam and gear functions over the bus while the broadcast facility is used to synchronise the sample periods of all drives down to the micro-second level. Within the ESPRIT project ASPIC[1], a device profile has been defined which details an open standard for addressing drives on a CAN network. This profile defines two channels of information over the bus, an operational channel for high speed, real-time data, and a service channel for parameter data. In designing the CAN Drive Profile, drive profiles for other bus systems were also taken into consideration.

Introduction

In the field of factory automation and production cells, the use of increasingly powerful digital technology has resulted in increased flexibility and productivity. Such technology allows distributed intelligence and control in sensor and actuator devices, which in turn has allowed increased functionality and lower overall system price.

The requirements of today's flexible automation systems have led to the development of the next generation of servo-controllers. These are based on "Intelligent Drives" communicating with a host controller over a high speed serial network - CAN. Functions available include speed and position loop closure, point-to-point motions and electronic gearing. In addition, all necessary parameters are downloaded over CAN eliminating the need for individual setup of each drive. Online status information can also be monitored.

However such a distributon of intelligence has highlighted the need for standardising the method of information exchange between devices, based upon a common communication medium.

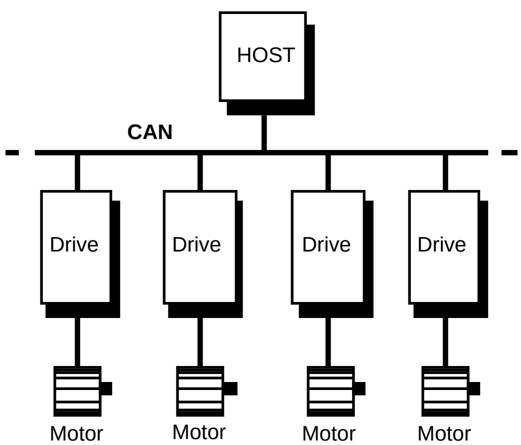


Figure 1. Block Diagram of CAN Servodrive System

Profiles

To fully exploit the benefits of distributed control systems, profiles must be defined for each device type in order to allow the user a common logical and physical interface. Profiles can be split into two parts. The first part is the Device Profile. A different Device Profile exists for each device type:- for example Drives have a different profile to that of Proximity Switches. The second part is the Communication Profile which defines the interface to the communication medium. For a given communication medium, the Communication Profile is the same for all devices. Profiles define the interface to similar type devices from different manufacturers, such that a given command to any such device will always provide the same response, independent of the manufacturer. The aim of this paper is to present a Device Profile for servodrives which will allow them to be connected and to communicate over the network in a standardised way. To achieve this, the functionality of the drive and the means by which it is accessed over the network must be defined.

The profile presented here is based upon the Controller Area Network (CAN)[2] as the communication medium. CAN is a high-speed serial bus for distributed control applications and was chosen for its high speed, high reliability and low cost. It is a two-wire, electrical standard operating at transfer rates up to 1Mbit/second. It is widely used in the automation and automotive industries and is supported by the major semi-conductor manufacturers.

The Communication Profile was developed in the ESPRIT project ASPIC[3,4] and is based on the CAN Application Layer (CAL)[5]. This profile addresses the needs of communication in a distributed realtime environment, as well as providing the means for open interconnect of devices in many different industrial applications.

Servodrive Functionality

A Servodrive controls the torque, velocity and position of an electric motor and may have all the functionality of a complete single-axis motion controller. It is assumed that velocity and optionally position feedback signals are available in the drive. All the drive functions are set-up by the host controller via CAN using parameters. The definition of these parameters and their representation is defined in the Drive Profile. Up to 64 Intelligent Drives can be run on a single bus. All the drives are synchronised via CAN on a sample-time basis to ensure absolute co-ordination of motions. Each drive can carry out different types of motions including point to point, jog, interpolation and electronic gearing/camming. A Servodrive may be modeled as shown in Figure 2.

The Bus Interface provides the link between the CAN network and the Drive Supervisor. The interface looks after the reception and transmission of messages on CAN. The interface also synchronises the internal timing of the Drives to the Sync message on the bus. In this way, the sample period clocks of all Drives are exactly synchronised.

Drive Supervisor

The Drive Supervisor controls the operation of the drive. It responds to commands and parameters sent over the bus. It starts motions, sets and reads parameter values, monitors system performance and provides status information such as motor position, faults, digital inputs and reference generator status to the host over CAN. The Drive Supervisor also contains a state machine which controls start-up of the drive and the various operating modes.

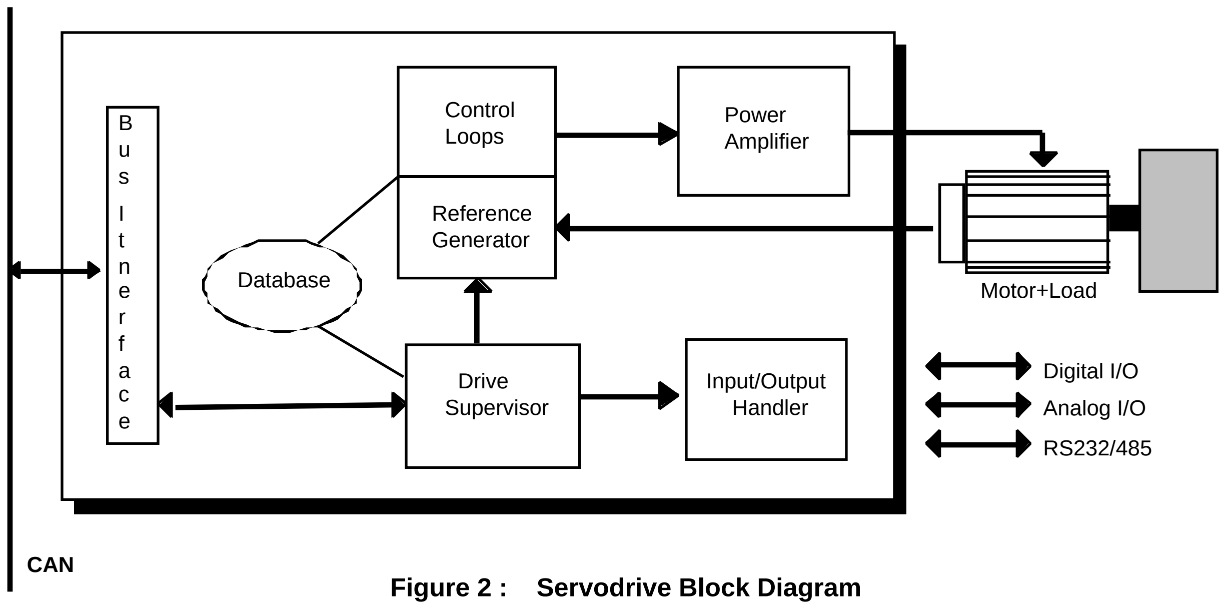


Figure 2 : Servodrive Block Diagram

Database

The database contains all the parameters for the Servodrive. These are defined in the Object Dictionary. This includes the standard Object Dictionary defined in the Drive Profile and any additional parameters which are manufacturer specific. The Object Dictionary also defines which parameters are mandatory for a particular operating mode. All drive parameters are accessible for reading and/or setting over CAN. All parameters are defined in the following way:

Type	Parameters may have a Basic Data Type such as Integer_16 or may have a structure composed of basic types e.g. Operational Mapping Structure.
Range	Gives the allowable range of the parameter.
Scaling	Defines the units of the parameter.
Default Value	Value of parameter which should be used if not written by the Host.
Mapping	Defines whether the parameter may be mapped to the Operational Channel or not.
M/O	Defines whether the parameter is Mandatory or Optional. Some parameters are only required for certain modes of operation.
Meaning	A short description of the parameter and its function.

Input/Output Handler

To enable the CAN drive to function as an intelligent sub-system, a number of I/O functions are required. These include for example relay outputs, limit switch inputs and an RS232 port for direct access to the drive parameters. Integrating this I/O in the drive avoids blocking up the bus bandwidth unnecessarily with low level sensor information and allows the safety functions to operate independant of the CAN network. It also means that the essential drive functions can be tested without having a CAN connection which speeds up fault isolation and debugging during commissioning.

Control Loops

Loop Closure is the basic functionality found in any servodrive. Traditionally this was limited to a straight-forward PID velocity loop having a ±10V reference command. Replacement of the ±10V command with a digital interface gives the ability to transmit many different parameters including 32 bit reference commands. This allows the loop closure functionality to be increased substantially to include closing the position loop and adding velocity / acceleration feedforward.

Reference Generator

The reference generator is used to generate the real-time command references for the control loops. Motion commands, together with the necessary parameters, are sent over the bus. To guarantee synchronous motion of multiple axes, motion commands sent to different drives within one command window all start at exactly the same instant. Depending on the application, the Host may select one of several basic modes of operation.

In **Profile Generation** mode, the Host only needs to specify the parameters of the motion segment required. The realtime generation of the position or velocity setpoints is then performed on the drive with continuous feedback of status such as actual position and following error to the Host. Both trapezoidal and S-curve profiles can be generated. The load on the Host is thereby considerably reduced allowing either a lower cost Host hardware or improved higher level functionality such as process visualisation. This mode is suitable for all applications which do not require fast real-time interpolation between multiple axes such as found in a CNC controller. Synchronisation between a master axis and one or more slave axes is possible using the Gear/Cam function. The standard motion commands are summarised in Table 1. Manufacturers may add additional commands as required.

In **Position** mode, position set-points are calculated in the Host controller. This allows the path resulting from the motions of multiple axes to be interpolated on the host and thus controlled precisely. Set-points are transmitted every communication cycle over the CAN bus. To achieve good stiffness and smooth performance, these set-points may be interpolated on the drive to allow the position loop to updated at a higher rate than the bus communication cycle. Position mode involves a higher loading on the Host than Profile Generation mode but is required in applications where path control is necessary.

In **Velocity** mode, velocity set-points for the drive are calculated on the Host and transmitted every communication cycle over the CAN bus. These are used as the command reference for the velocity control loop. To achieve smoother peformance the set-points may be interpolated on the drive. This mode is used when the drive must run continuously at a set velocity.

In **Torque** mode, the Host sends Torque set-points to the drive and these are used by the drive to generate current command references thereby controlling the torque. Torque mode may operate in either open or closed loop. To operate in closed loop, an external torque feedback signal is required.

Point to Point	Absolute or relative move. Trapezoidal or S-shaped velocity. Position and velocity variable on-the-fly.
Move Time	Trapezoidal or S-shaped velocity. Profile calculated to complete motion within the specified time.
Jog	Trapezoidal or S-shaped velocity. Velocity and acceleration variable on-the-fly. Used when the drive runs continuously with a given velocity profile.
Quick Stop	Linear velocity ramp to stop with programmed deceleration. Used for emergency conditions.
Homing	Seuence of motions to find home position based on digital input and resolver zero.
Triggered Motion	Trigger start of Point to Point and Jog motions based on transition of digital input.
Electronic Gearing	Broadcast of position from Master Axis over bus. Gearing ratio set in slave drives. Gearing ratio variable each communication cycle.
Electronic Camming	Extension of electronic gearing with non-linear table in slave. Cam amplitude and offset variable each communication cycle.
Function Generator	Square-wave generator with programmable amplitude, offset and frequency. Used for tuning the drive.

Table 1. Motion Commands for Profile Generation Mode

CAN Communication Protocol

CAN's real time distributed peer-peer and broadcast/multicast abilities makes it a very suitable choice as an industrial fieldbus for servo-drive applications. The following outlines the Communication Profile concept, in terms of bus message traffic and type of data transferred.

Channel Concept

Data transfer between the Host and Intelligent Drive may be classified into two groups :

- **Operational Data:** This is high-priority operational data which must be sent regularly, e.g., reference position. Transmission of this data is done via the ***Operational Data Channel.***

- **Service Data:** This is low-priority data which is sent at irregular intervals, e.g., parameterisation data, diagnostic data. Transmission of this data is done via the ***Service Channel.***

Real Time Behaviour

The real-time behaviour of the bus-traffic is modelled with respect to two types of operation:-

- **Synchronous:** The host transmits the SYNC message on a cyclical basis. All Synchronous Messages (e.g. COMMAND and ACTUAL) are transmitted within a certain time window with respect to the cyclical transmission of the SYNC. This guarantees the real-time update of all process-related dynamic data. Operational Data Channel messages are typically synchronous.

- **Asynchronous:** Message transmission is on demand, rather than with respect to the SYNC. Service Channel messages are typically asynchronous. Event type data on the Operational Channel may also be asynchronous.

Seven message types are used for communication and these are defined as CAN objects and assigned CAN Identifiers. These messages are described in Table 4. Depending on the real time requirements, these messages are sent using either the Operational Channel or the Service Channel. Both the SYNC and the MASTER_REFERENCE messages are broadcast to all drives and there is only one instance of each of these. The other messages relate to a specific drive in the network and therefore have as many instances as there are drives on the network. The messages are prioritised as listed in Table 4, with the EMERGENCY messages having the highest priority.

Message	**Transmission Direction**	**Real Time Behaviour**	**Data Bytes**
EMERGENCY	Drive _ Host	Asynchronous (Operational)	2
SYNC	Host _ All	Synchronous (Operational)	1
MASTER_REFERENCE	Drive _ All	Synchronous (Operational)	6
COMMAND	Host _ Drive	Synchronous (Operational)	1 - 8
ACTUAL	Drive _ Host	Synchronous (Operational)	1 - 8
HOST_REQUEST	Host _ Drive	Asynchronous (Service)	8
DRIVE_RESPONSE	Drive _ Host	Asynchronous (Service)	8

Table 2 : Intelligent Drive CAN Messages

Typical traffic on the bus is shown in figure 3. The function of the individual messages is as fpllows:

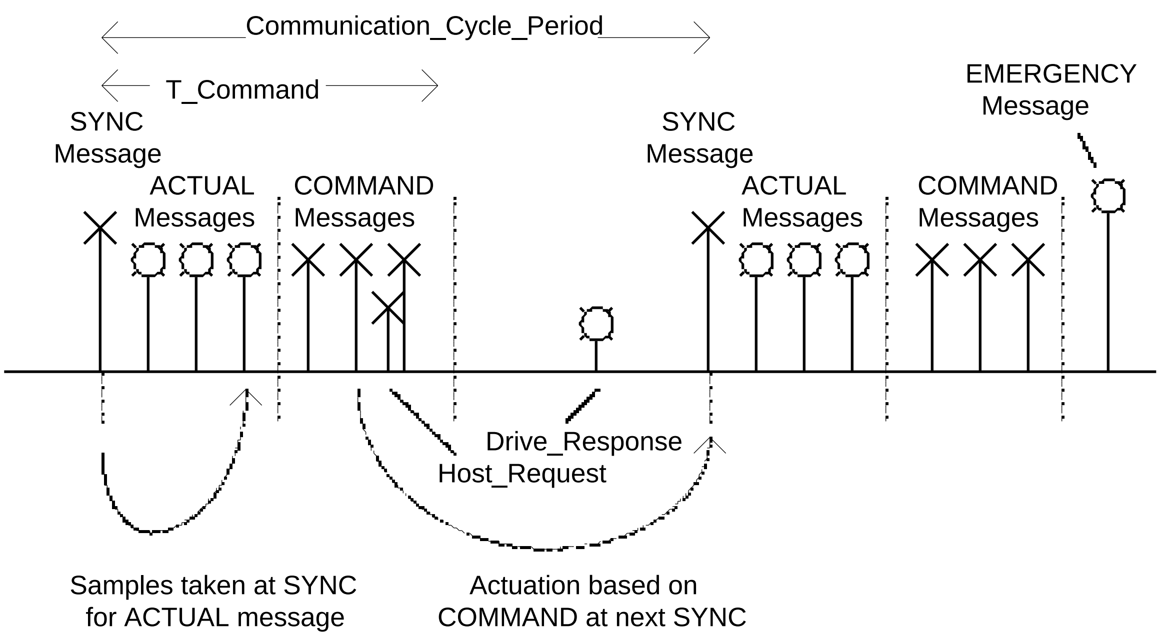


Figure 3. Typical Bus Traffic

EMERGENCY
The Emergency message is transmitted asynchronously by the drive whenever a fault condition occurs. The Emergency Message contains the STATUS_WORD parameter. The STATUS_WORD may also be transmitted cyclically in the ACTUAL message.

SYNC
In order to guarantee simultaneous update of command reference values for control loops in different drives, the SYNC message is transmitted by the host at a regular interval. The period between the

SYNC messages is specified by the Communication_Cycle_Period parameter. The drives may use the SYNC message to synchronise their own timing with that of the master.

MASTER_REFERENCE
Transmitted by a drive, which is designated to be a gearing master by the host. The MASTER_REFERENCE is used in drives which are designated by the host to be gearing/ caming slaves. When a Drive is configured to transmit the MASTER_REFERENCE message, it transmits this message periodically, at a frequency which is an integral multiple of the SYNC frequency of transmission, see figure 4. The transmission of the MASTER_REFERENCE message may be at a higher rate than the SYNC in order to provide a real time synchronised update of a process related variable (a master drive's shaft position). The Rate of transmission is defined by the Master_Reference_Rate parameter. Note that the MASTER_REFERENCE message has a higher transmission priority than the COMMAND or ACTUAL messages but lower than the SYNC. The transmitting drive should arrange to transmit its MASTER_REFERENCE at a time which does not cause CAN arbitration with the SYNC. The transmission of the MASTER_REFERENCE message may result in CAN arbitration because of simultaneous transmission of the COMMANDs and ACTUALs.

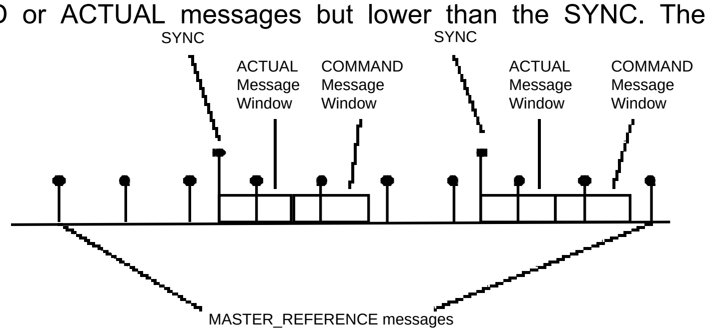


Figure 4 : Transmission of the MASTER_REFERENCE message

COMMAND
Used to transfer operational command data from the host to the drive. e.g., Enable drive, Move to endpoint xxx, etc. This contains up to eight bytes of data with the first byte defining the command and the remaining bytes containing appropriate data for that command. The COMMAND message is transmitted cyclically in a defined time window after the SYNC message. The length of this time window is specified by the T_Command parameter.

ACTUAL
Used to transfer operational status data from the drive to the host e.g., Fault information, position information, etc. This contains up to eight bytes of data. The ACTUAL message is transmitted cyclically in a defined time window just after the SYNC message.

HOST_REQUEST
The Host Request is the client's request for a read/write parameter access, as part of the Service Channel. This message uses the CiA's Multiplexed Domain protocol. The host supplies the index and optional sub-index of the parameter to be read or written. Parameter meaning and format is detailed in the Drive Object Dictionary.

DRIVE_RESPONSE
The Drive Response is the server's response to a read/write parameter access, as part of the Service Channel.

Initialisation of the Servodrive

The standard CAL initialisation procedure is used to bring the drive into the OPERATIONAL state. When the drive is initially powered on it is in the DISCONNECTED state. The NMT services of CAL are used to CONNECT the drive and create a REMOTE NODE in the host. After this the drive enters the PREPARING state where the DBT master assigns CAN identifiers to each of the CAN messages in the service and operational channels. If no DBT service is provided, the default CAN identifiers defined in the Drive Profile shoud be used. Once the CAN identifiers are assigned, the drive is in the PREPARED state. From this state, it may be switched at any time to the OPERATIONAL state by the NMT master. At this stage, the Service Channel is fully operational. The CAN Identifiers of the messages in the Operational Channel have been assigned but the contents of the COMMAND and ACTUAL messages have not been set up.

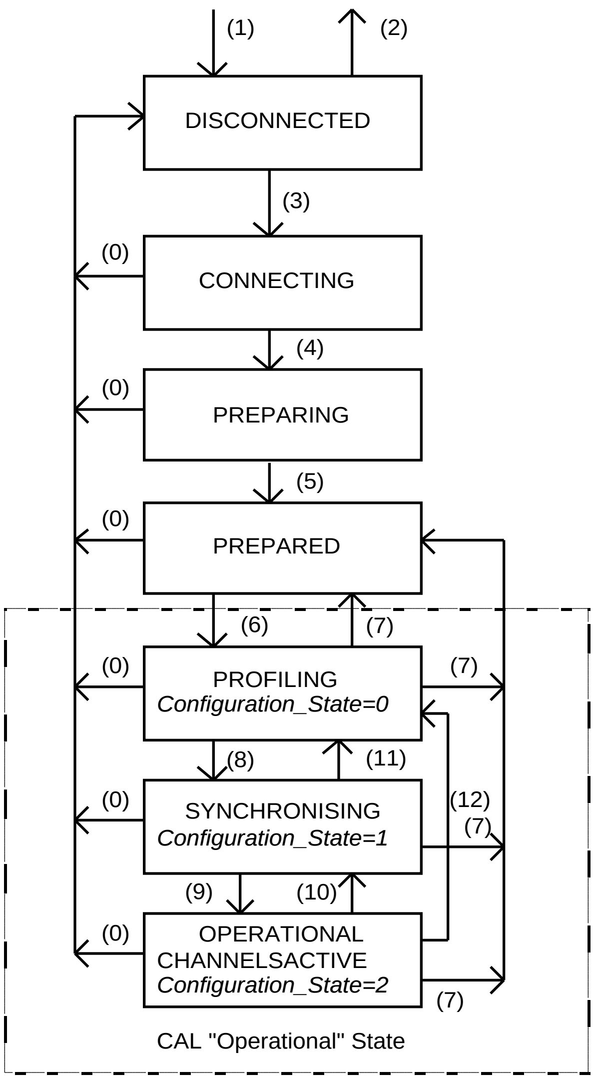
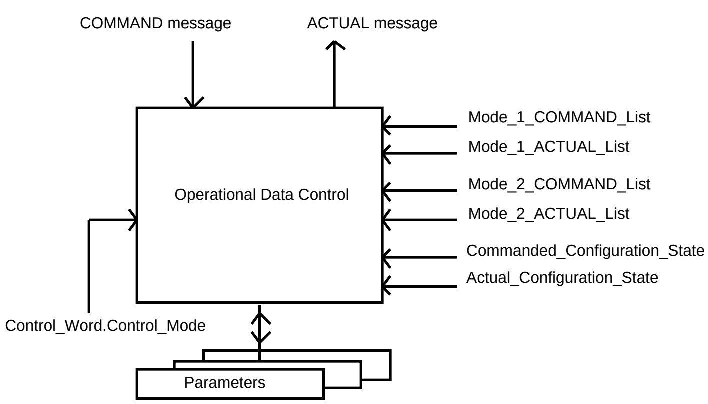


Figure 5. Bus Interface State Diagram

The OPERATIONAL state is divided into three sub-states. The first of these is called the PROFILING state during which the Host uses the Service Channel to parameterise the drive and to configure the Operational Channel. The COMMAND and ACTUAL messages may be specified using the parameters Mode_1_COMAND_List and Mode_1_ACTUAL_List respectively. These contain a list of up to eight drive parameters which will be transmitted on the operational channel. The default parameters in the COMMAND message are Control_Word, Reference_Velocity and Reference_Position. The default parameters in the ACTUAL message are Status_Word, Actual_Velocity and Actual_Position. As an option, a second Operational Channel mapping may be defined using Mode_2_COMMAND and ACTUAL List parameters. This allows high speed switching between two alternate modes during drive operation, e.g. changing from Position Mode to Torque Mode.



The mapping of parameters to the Operational Channel is controlled by the Operational Data Contol function. This function takes the input parameters Mode_1 and Mode_2 COMMAND_Lists and ACTUAL_Lists, and maps appropriate parameters to the COMMAND and ACTUAL messages, corresponding to the Control Mode (1 or 2).

Figure 6. Control of Operational Data Channel Mapping

After the parameters have been set up the drive is placed in the SYNCHRONISING state. The Host require synchronisation but for them it is equivalent to the PROFILING state. When the drives are synchronised the host may switch them to the OPERATIONAL CHANNELS ACTIVE state and start transmitting COMMAND messages and receiving ACTUAL messages. During all of these sub-states, the Service Channel may be still be utilised.

Drive Power Generation Control and Fault Reaction

Once the drive is fully operational, the Drive Supervisor is responsible for reaction to fault conditions of the drive, control of the Power Generation Section, control of sequences of operation of other functional blocks and the generation of a status report to the host. The Drive Supervisor Block is accessed principally through the Control Word and the Status Word.

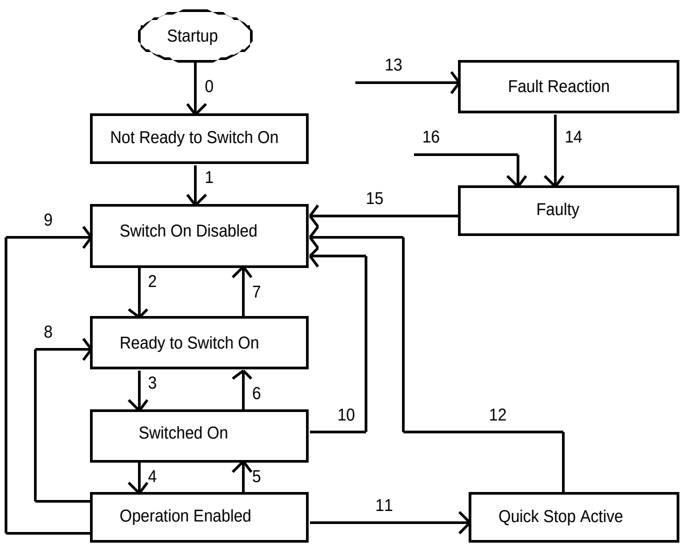


Figure 7. Drive Supervisor State Diagram

The state diagram in figure 7 describes the state machine operation of the Drive Supervisor with respect to control of the power electronics, as a result of user commands and internal drive faults.

When a fault is detected, the Drive Supervisor sets the fault bit in the Status Word. The host can then get details of the fault by reading the Fault_Code parameter. This 16 bit code is defined by the Fault_Code table in the Drive Profile.

Drive faults are classified as fatal or non-fatal. When a non-fatal fault occurs, the drive can still actuate the motor in a controlled fashion. When a fatal fault occurs the drive can no longer control the motor, so an immediate switch off of the drive is necessary.

System Integration

Standardisation of Device Profiles and the higher communication layers for CAN, together with the growing number of interface modules for various control platforms, will make it easy to integrate Servodrives and other devices to a host controller. Software integration is acheived using high level function calls, for example to generate moves, change gear ratio or read the drive status.

PC, VME and PLC Host Controllers

For these standard controller platforms there are various CAN interface cards already available. In terms of bus traffic and timing, servodrive applications are quite demanding. It is therefore advisable to use an interface card with its own CPU so that the application protocol can be implemented separately and remains independent of the load on the Host Controller.

Machine Specific Host Controller

For machines produced in larger quantities, it is often more economical to design a machine specific host controller. In this case, Intelligent Drives can be linked into the system by adding a CAN interface to the host and by writing suitable driver functions on the host which directly access the parameters defined in the Drive Profile. Integration on various host controllers has been sucessfully carried out in several cases with minimum effort.

Advantages of Networked Servodrives

Digital Servodrives connected via a digital serial network offer many advantages over the classical connection with analog reference and discrete command/status signals. As the use of networks increases drives will not only be able to communicate with each other as in Master/Slave mode but will also be able to communicate directly with other devices such as sensors and on/off actuators to realise true distributed intelligent control. Some of the direct advantages of networked drives are:

- Less Wiring - Lower Installation Costs
- Less Interconnections - Higher Reliability
- Status and Diagnostic Information
- Intelligent Drive offloads Host Controller
- Single MMI - All Drives Parameterised over CAN
- Drift-free, Repeatable Performance
- No Analog References - No Noise Problems
- Flexible - Reliable - Maintainable

Application Exapmles of CAN Servodrives

For several years, CAN Servodrives have been applied to a Scara Robot application. All communication between the Servodrives and the Robot Controller is via CAN. The Robot Controller calculates cyclic command positions for all axes and transmts these to the Servodrives where the positions are fine interpolated. The synchronisation of all axes is achieved by the real-time behaviour of CAN.

In a printing application, CAN Servodrives are used to position the print rollers for different print colours. The Servodrives are slaved to a Master axis which pulls the printed medium through the machine using using the CAN Master/Slave mode. This maintains precise synchronisation of all axes which ensures no overlapping of colour patterns. Not only does this replace many expensive gear boxes and drive shafts but it also dramatically reduces the set-up time of the machine. New patterns can now be reprogrammed at the press of a button.

A further application uses CAN Servodrives on a Cartesian Robot for unloading injection moulded parts. Removing the need to close high speed position loops in the robot controller allows considerable reduction in the size and cost of the Robot Control hardware.

Further Work

Within the CiA, the Special Interest Group (SIG) on Robotics and Drives is working to define a Device Profile for Servodrives. This will be based on the profile described above as well as experience gained from other fieldbus drive profiles.

[R]eferences

1 ESPRIT Project 7302 ASPIC
"Automation and control Systems for Production units using an Installation bus Concept".

[2] ISO/DIS 11898:1992, Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for high speed communication.

[3] ASPIC Consortium: Specification of Installation Bus System on Production Units. 1994. Available from STA, Alteburgerstr. 150, D-72762 Reutlingen.

[4] Rostan, M. "CAN Real-time Communication Profile". Proceedings of 1st International CAN Conference, 1994.

[5] CiA Standard "CAN Application Layer", CiA/DS 201-207, 1994.