iCC 1997

4th international CAN Conference

in Berlin (Germany)

Sponsored by

Motorola Semiconductor
NEC Electronics (Europe)
Siemens 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

Jürgen Klüser, Vector Informatik GmbH

Project Planning and Device Configuration in CANopen Networks

In project planning for larger distributed systems there are a number of tasks that are time consuming, unclear and susceptible to errors. For CANopen systems and other networked facilities, these tasks include:

- **Analysis and structuring of all automation tasks, development of functional groups**
- **Selection of the required network devices**
- **Definition of the process data**
- **Definition of the communication parameters**
- **Establishing device parameters / application objects**
- **Programming and integration of control units (PLC, IPC, ..)**
- **Configuration download, start-up**
- **System analysis**
- **Documentation**
- **Customer service**

This paper gives an overview of these important project tasks and details some selected topics. One specific part is the discussion of requirements to CANopen from the view of programmable devices such as PLC or IPC. Regarding this, the paper presents a brief overview about parts of the work of the SIG Programmable Devices.

Analysis and Structuring of the Automation Task

First let's have a look into the typical method of setting-up networked facilities. Usually the customer is a mechanical engineering company with deep knowledge about the production process itself. But very often there is only a vague idea of the technical details concerning the network. This common problem leads to incomplete or even completely missing specifications for the network system integrator.

To date, the widespread practice is to start the integration without a clear methodical analysis and structuring of the entire task. Devices are commonly integrated one by one using a trial-and-error method. Adding the next device often leads to going back to the previous step and redoing it. This method is very time consuming and unclear. The current trend in automation technology of moving toward distributed intelligence even complicates the situation and requires even more care in system integration.

To satisfy these system level requirements, software tools for project planning are very helpful. Such a tool has to support the structuring of the project into functional groups. Graphical visualisation of the topological situation helps to avoid losing track of things. Pure Master/Slave systems have to be supported as well as Peer-to-Peer architectures. A textual description of the complete project task, each functional group, and of each device guides through all steps of the planning

Selection of the required Network Devices

In the next step the devices required in the distributed system are selected based on their system function in the facility. Although this selection is influenced from long-standing co-operation with hardware manufacturers or marketing aspects, the capabilities of the required devices have to be well defined with consideration to their tasks. On the one side this concerns the device application - where each type of device such as frequency converters, drives, I/O devices or MMIs has its own special characteristics - and on the other side the network communication capabilities. Assume that a device shall broadcast signals to a set of several other devices. Since this is not part of the mandatory pre-defined communication set, features such as PDO re-mapping and dummy mapping, and a sufficient number of PDOs are required.

This is particularly important for Multi-Master architectures since the communication interrelations are quite complex and thus make high demands to the communication abilities of the devices.

The profile concept of CANopen eases the selection and exchangeability of devices. All devices are grouped in families with the same basic behaviour. For example the input line states of an input device are accessible in an exact, defined way completely independent from the device manufacturer. Sometimes the system integrator has to distinguish between features that are mandatory and features that are optional. For example, the drive profile defines operation modes such as Homing Mode or Interpolated Position Mode and defines clearly the behaviour of these modes - but the profile does not force manufacturers to implement all modes.

CANopen defines an Electronic Data Sheet "EDS". This standardised file format contains a description of general device features, version management and the complete object dictionary structure. Concerning the device selection this enables software tools to show a list of all available devices together with their specific attributes. Project Planning Tools, Configuration Tools, Analysis Tools and other tools can access this information in a well-defined manufacturer independent way. In this manner the CANopen end-users, system integrators and developers are free in the selection of software tools without dependencies to the devices used in the network.

Definition of the Process Data

For access to device parameters and process data CANopen defines an addressing scheme which uses an Index (and Sub-Index) into the object dictionary of each device. Additionally each object has a name that is pre-defined in the profile. Assume a system having dozens of input modules. They all would have an object with the reader-unfriendly designation "Index 6000H, Sub-Index 01H, Read_8_Inputs_1H-8H". The usage of all these input objects with same index and name - only distinguishable by the device's Node-ID - has the danger of confusion and mistaken connections.

One solution to this problem is the use of unique, easy-to-understand names such as "EmergencyStop" or "MoldPressure". Both sides of a process data connection can use the same names for their mapped objects. This concept can run through the complete system including the naming of the appropriate variables in the programming system of the control units, for example an IEC1131 system.

It is a high demand to all software tools - especially project planning tools and configurators - to support this symbolic naming throughout the project. The storage of assignments from CANopen object addresses to symbolic names can be done in the so-called Device Configuration Files "DCF". This standardised file format stores all attributes of a concretely configured device in addition to the information from the EDS. The DCF files can be used for standardised data transfer between any kind of software tools - including the symbolic names.

Definition of the Communication Parameters

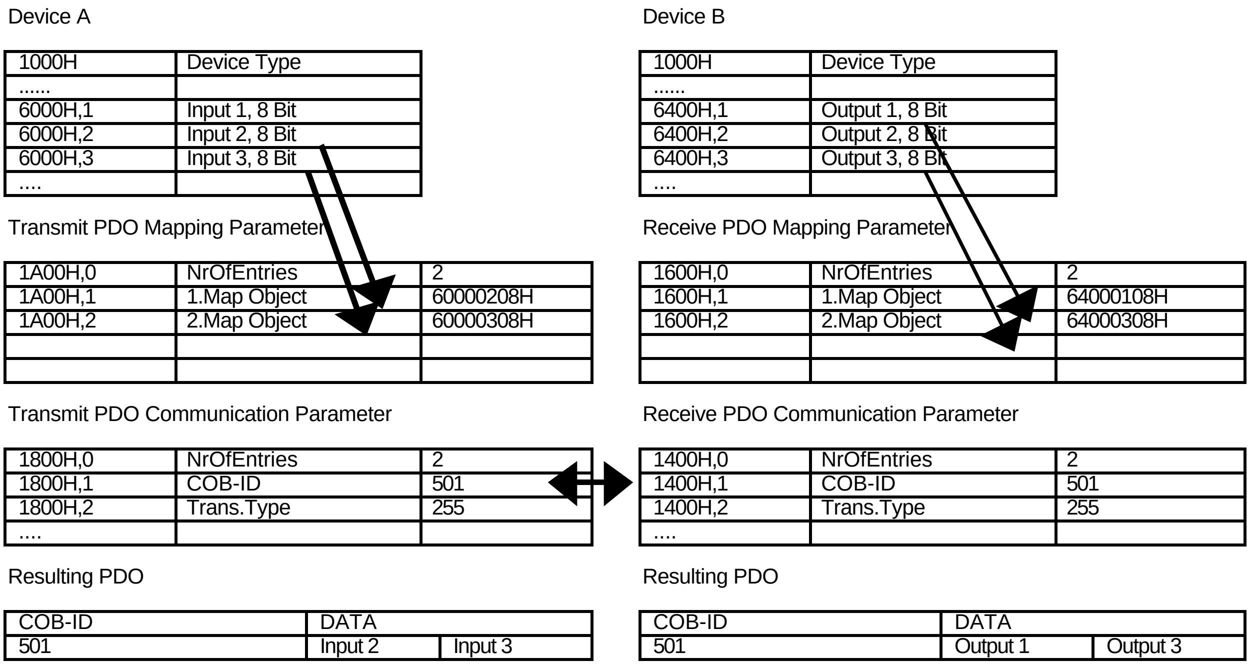
Communication parameters can be divided into global communication parameters and connection

Global communication parameters are settings such as baudrate, SYNC object and communication cycle period. It has to be ensured, that all clients of a SYNC have the same settings of communication cycle period and synchronous window length. Setting parameters for many devices is potentially error prone and unnecessarily time consuming, but can be easily done by configuration tools.

The configuration of connection related communication parameters consists of PDO Linking, PDO Mapping and the settings of the appropriate transmission types. CANopen and its device profiles define default values for these attributes. Especially the so-called Predefined Connection Set allows the set-up of pure Master-Slave-Systems without the need of configuring each device's PDOs. All slaves work with their default settings. The application programmer can access all PDOs by the knowledge of the COB-ID distribution rules and the profile dependent process data mappings. This often used method avoids the need (and costs) for extra configuration software, but has the disadvantage of mixing application code and network specific configuration stuff. Furthermore the application programmer has to have deep knowledge about CANopen internals.

The Predefined Connection Set can neither be used for Multi-Master architectures nor for Master-Slave-Systems with direct connections between slaves. A common term for these systems is Decentralised Intelligence. For these systems the PDO Linking and Mapping must be performed. This mechanism is demonstrated in a simple example:

Transmit Input 2, Input 3 of device A to Output 1 and Output 3 of device B.



Device A

1000H	Device Type
......	
6000H,1	Input 1, 8 Bit
6000H,2	Input 2, 8 Bit
6000H,3	Input 3, 8 Bit
....	

Device B

1000H	Device Type
......	
6400H,1	Output 1, 8 Bit
6400H,2	Output 2, 8 Bit
6400H,3	Output 3, 8 Bit
....	

Transmit PDO Mapping Parameter

1A00H,0	NrOfEntries	2
1A00H,1	1.Map Object	60000208H
1A00H,2	2.Map Object	60000308H

Receive PDO Mapping Parameter

1600H,0	NrOfEntries	2
1600H,1	1.Map Object	64000108H
1600H,2	2.Map Object	64000308H

Transmit PDO Communication Parameter

1800H,0	NrOfEntries	2
1800H,1	COB-ID	501
1800H,2	Trans.Type	255
....		

Receive PDO Communication Parameter

1400H,0	NrOfEntries	2
1400H,1	COB-ID	501
1400H,2	Trans.Type	255
....		

Resulting PDO

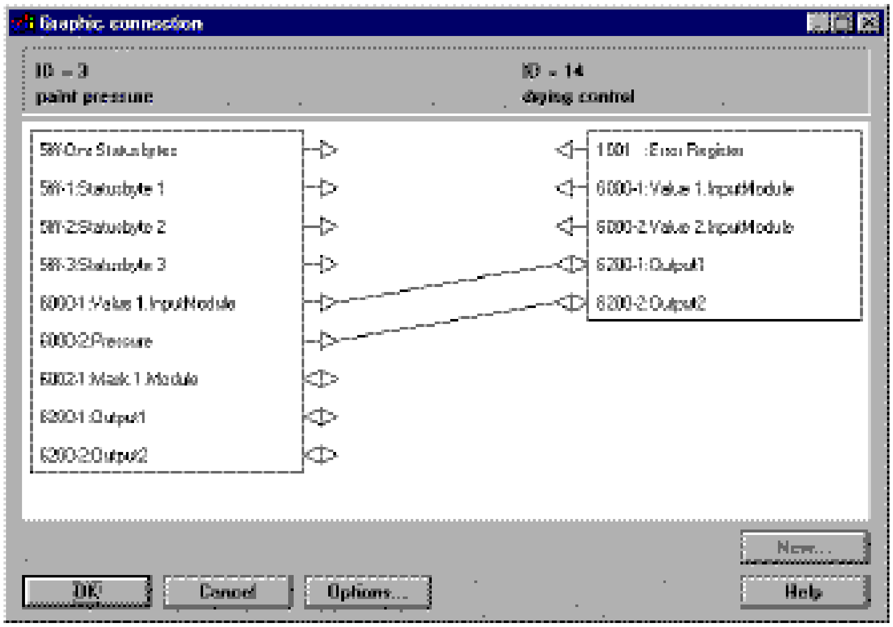
COB-ID	DATA	
501	Input 2	Input 3

Resulting PDO

COB-ID	DATA	
501	Output 1	Output 3

This example shows, that the system integrator requires deep knowledge about the details of the CANopen communication object tables. The complexity will grow significantly with increasing number of devices and the usage of the broadcast mechanism. Additionally the system integrator will be confronted with the problem of devices without re-mapping ability.

A sophisticated project planning tool is able to take over the task of generating the appropriate PDO Linking and PDO Mapping tables. The user only needs to define the communication channels by simple drag-and-drop with the mouse. A display of such a connection is shown in the following picture:



Additional attributes such as the Transmission Type can be handled as local parameters (options) of the connection. This avoids the setting by mistake of different transmission types in both end points of the connection.

Establishing Device Parameters / Application Objects

Many devices provide configurability of a number of application parameters, such as threshold values for analogue I/Os or ramp parameters for drives. With the Service Data Object (SDO) CANopen defines a mechanism to access all those parameters via the object dictionary. The term "Device Configuration" normally means this part of the network configuration process. There exists a series of Configurations Tools of different manufacturers for this task.

Programming and Integration of Control Units (PLC, IPC, ..)

In a network programmable nodes can be characterised as a process with input variables and output variables. The set of variables will be arguments of the program and hence will be only known in a final state when the program has been written. The arguments must be handled as variables located in the object dictionary.

The marking of such parameters depends on the programming system (e.g. IEC1131) and can not be standardised by CiA. But it can be assumed that there is a set of network variables with the logic attribute EXTERN.

Compiling/Linking (or interpreting) a program including EXTERN variables requires relocation information. Within CANopen devices this information is the Index (and Sub-Index) of the variable. Most of the programming systems know the mechanism of a resource definition. This can be used to assign the CANopen attributes (Index, Sub-Index, R/W, Assignment of CANopen data type to local data type etc.) to the corresponding symbolic names (variable name in the program). The resource definition may be created with a simple text editor or with much more comfort by a configuration tool. On systems with a disk-based file system a direct exchange via the DCF format is even possible.

Defining EXTERN variables requires a rule for distributing the indices. The SIG Programmable De-process picture of the device. This allows PLC/IPC manufacturers an easy-to-implement linear address calculation.

Besides the information of the EDS, the DCF stores the dynamic allocated objects. With this information it is possible for a configuration tool to retrieve the objects of a programmed device. In the other direction it is possible to define variables with a project planning tool and afterwards use that information by the linker/relocator/resource definition editor of the programming system. This makes it possible to implement programming systems with nearly no knowledge of CANopen.

The mechanism has been already proven in practice. Programming Systems and PLCs that follow the mechanism are available on the market. Vector Informatik GmbH has implemented it in its project planning tool ProCANopen.

Configuration Download and Start-up

The task of downloading the project data into the physical network can be sub-divided. One task is the testing start-up and interactive configuration with single devices, groups of devices or the complete network. This normally is done with a configuration tool. The other task is the configuration of devices that cannot store their parameters in non-volatile memory. This operation must be performed on every network boot-up.

The on-line configuration and boot-up of single devices with configuration tools is strait forward with the use of the Service Data Object (SDO) and some network management (NMT) services. This feature can be expected from every configuration tool. Since SDOs are point-to-point connections and devices seldom support more than one SDO, only one client can access a device at a time. This implies, that the designated network master must be turned off while configuring a device. In doing so, it is not possible to start-up a complete network including the master with a configuration tool. To solve this problem, the SIG Programmable Devices clarified the term master and defined a mechanism for dynamic establishment of SDOs. The new term CANopen Manager is introduced. It consists of NMT Master, SDO Manager, optional Configuration Manager and optional DBT Master. The SDO Manager provides synchronisation of SDO accesses by dynamic establishment. It is able to assign and withdraw the ownership of SDOs to several requesting devices such as Configuration Manager, configuration tools, analysis tools, MMIs and others. The paper "Industrial Control Systems with Distributed Intelligence based on CANopen" from Prof. G.Gruhler, B.Dreier and J.Wanner gives more details for this topic.

On boot-up of a CANopen network several devices may have the need for configuration since they cannot store their configuration parameters in non-volatile memory. In those pure Master-Slave-Systems, that are not configured with configuration tools, it is possible and adequate to perform the boot-up configuration by the application. In that case, protocol dependent information is mixed into application related code. For the purpose of a clear distinction between application and configuration and especially for decentralised applications it is highly desired to have the possibility of external configuration. This requires an automated management of downloading the configuration at boot-up. The SIG Programmable Devices defined the so-called Configuration Manager which is part of the CANopen Manager.

The Configuration Manager includes a set of objects that describe the configuration parameters to be downloaded to the devices at boot-up. These objects will be configured with the same configuration tools that are used for device configuration. For the possibility of easy-to-implement Configuration Manager with only little requirements to system resources the SIG Programmable Devices defined a basic set of objects that store only the absolutely necessary information in a so-called concise format. For more sophisticated Configuration Managers several features up to the storage of the overall project data are possible. This enables the user or customer service to read-out the complete information from the network.

The boot-up of such a system typically consists of the following steps:

1. Power on
2. Configuration Manager configures devices via default SDO
3. NMT Master starts Node Guarding by transmission of RTR
4. SYNC initialisation
5. NMT Master sets all nodes to operational state

For decreasing the boot-up time, the Configuration Manager checks, if the slaves have already loaded the required configuration. For example this may be true, if the slave is able to store its configuration in non-volatile memory. The check is done with the object Verify Configuration. This contains a configuration identifier that can be compared to the configuration identifier of the Managers database for each slave.

Documentation

A particular important component of project management is comprehensive documentation. It is mandatory for project planning tools to generate the complete set of documentation about the network information. It should be possible to combine this with application documentation such as the documentation output of an IEC1131 programming environment. Beside the print-out on paper, the information export to database systems and word processing systems is required.

System Analysis and Customer Service

For general CAN-based distributed systems CAN network system analysis is another important tool-related area especially when such tools share information with other tools. Because the usage of project planning and configuration tools creates complete descriptions of the network, and since CANopen standardises those file formats, it is possible to directly move that information to special analysis tools such as Vector's CANalyzer and CANoe.

The same is true for support of the customer service. Additionally project planning and configuration tools can read-out all device parameters and process objects; thus helping to find problems in communication or application related parameters. The usage of symbolic names as described above helps to get an overview of all system functions very quickly. Some more helpful features can relieve the users' workload. These are functions such as scanning the network or comparison of the actual project data to the originally planned parameters - to name only a few.

Conclusion

Project planning is a very complex task. CANopen and the related profiles satisfy all requirements for setting up a consistent communication network. The usage of CANopen project planning tools and configuration tools helps the project manager to ensure network consistency, to reduce complexity and to perform many tasks automatically. This leads to a much higher work efficiency, network security, less errors, and an easier to manage system.

References:

[1] M. Litschel, J. Klüser: Tool for Project Planning of CANopen Networks, Stuttgart 1996

[2] CAL specifications DS 20x, Erlangen, CAN in Automation.

[3] CANopen specifications DS/DSP 30x, 40x., Erlangen, CAN in Automation.

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