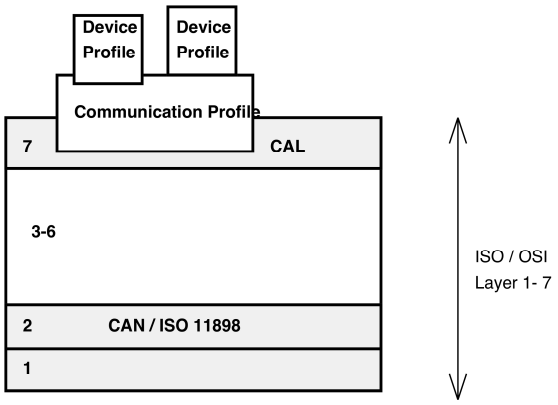
CAL based device profiles

Martin Jaeggi
Selectron System GmbH
Schupfer Straße 1
90482 Nürnberg, Germany

Introduction

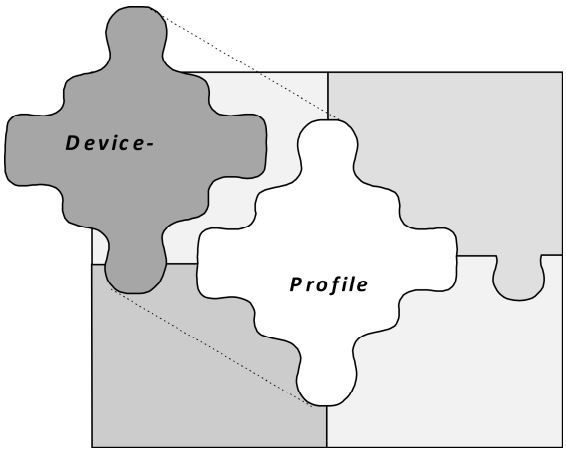
CAN is a Bus defined according to ISO 11898. It has been successfully tried and tested in millions of applications and is gaining popularity. For simple applications it is sufficient to allocate the identifiers (11 bit message recognition) to the different communication partners, and the setup is ready for data exchange. This simplicity is one of the important advantages provided by CAN and certainly the justification for it being so widespread. However this communication, based on layer 2 does not always provide the operating ease which is expected from a network today. CAL puts an easy-to-operate and efficient layer 7 at your disposal, permitting safe and simple data exchange between the communication partners.



If we install devices from different manufacturers, CAL does not guarantee that the appliances are interchangeable as the contents of a CAL data message are not defined. With a CAL superposed communication and device profile, we define further communication characteristics and thereby reach uniform data communication between all the appliances connected as well as the required inter-operability.

What is a profile ?

In communication technology (field bus systems), profiles are used to define generally valid communication characteristics of certain types of appliances (I/O-modules, drives, gateways). Devices with identical profiles herewith feature a common set of characteristics and functions. With regard to the user, this means that the appliances are interchangeable and that they can be operated together within a network (compatibility, inter-operability). Profiles offer the advantage of generally defined operating possibilities but are restricted to certain applications and, in most cases, cannot be used to use special functions.



In certain cases it is not required that a device fulfils all the functions of a profile. Thus, the profile leaves a number of options open for the appliance manufacturer. Only the essential features of the device have to be respected in all cases. Apart from the mandatory and optional functions, the manufacturer is free to integrate specific functions. The user must, however, be aware of the fact that such functions are normally not supported by other manufacturers, which results, in incompatibility.

Profile setup

The following description deals with the CAL based communication systems CANopen and the corresponding device profile for I/O modules. As the communication profile and the device profiles very much depend on each other, the following description does not distinguish clearly between the two things. For the user, this distinction, with regard to the use of I/O modules, is not very important. It is essential, however, to know the accurate, complete functional features of the device.

Based on an I/O module, we would like to show what a profile looks like. We will start with the

description of a minimal solution and will then integrate more and more functions in the device, until a final version is obtained which fulfils all the possibilities of the profile.

What is an I/O module ?

The definition of the I/O device profile today includes digital and analogue I/O's. These devices have their own intelligence (micro controller) so that more complex services of a protocol may be processed (contrary to SLIO-solutions). However, the user has no access to an internal programme and will thus configure the module by using the predefined functions. Furthermore, the profile does as e.g. not define which voltage ranges are possible for an analogous module.

For encoders, RS-232-C-converters, keyboards and displays, a first suggestion exists. However the corresponding profiles have, up to the present time, not been reviewed by the CiA.

The minimal version

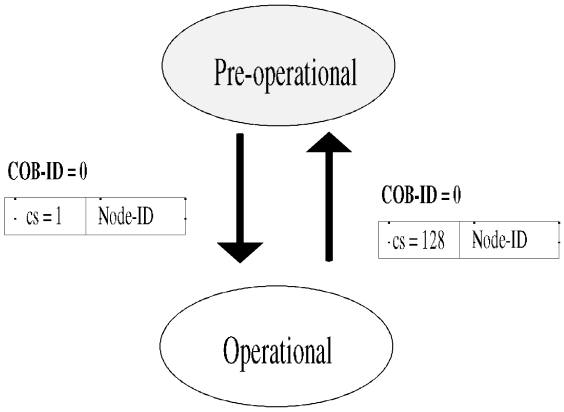
Let us consider a simple I/O module with 16 digital inputs and 8 digital outputs. For a minimal version, we will have to define the following CAN-identifiers (IDs) for the following tasks:

- Starting/stopping the data communication
- Transmitting input data
- Receiving output data

The communication profile prescribes that for event-controlled data identifiers in the priority classes 3, 4 or 5 have to be allocated. For us this means that both IDs for inputs and outputs have to be within the range of 661 and 1320.

0	Emergency messages
1	Synchronous messages
2	Synchronous Data's
3	Asynchronous Data's
4	(Asynchronous Data's)
5	(Asynchronous Data's)
6	Service messages
7	(Service messages)

The attribution of the IDs to data may be adjusted firmly, for instance ID 601 for the inputs and ID 602 for the outputs. In order to avoid a conflict between two appliances with the same IDs CANopen defines default IDs which depend on the node adress (CAL node ID). This can be modified e.g. by a DIP-switch.



To start, res. stop the communication, two CAL NMT services are used.

Thus, for complete implementation only 3 identifiers have to be installed in the node.

Questioning data

By default, our I/O module will only transmit the input data, in case one of the inputs has changed (event-controlled data transmission).

For most standard applications, this function is absolutely sufficient. However, what will happen if an operating terminal is switched on after the bus has been running for a certain time ? The terminal would receive the current data only after the input has changed and, up to that moment, would not display the correct values. The solution is that all process data are defined in the shape of the CMS-object "stored event", by the argument "immediate notify". What does this mean ? Data is transmitted immediately, as soon as the event takes place (the input has changed). The value is additionally "stored" and may be questioned again over a remote frame telegram.

After starting, our terminal may herewith collect all required data and need not wait until the modules announce themselves.

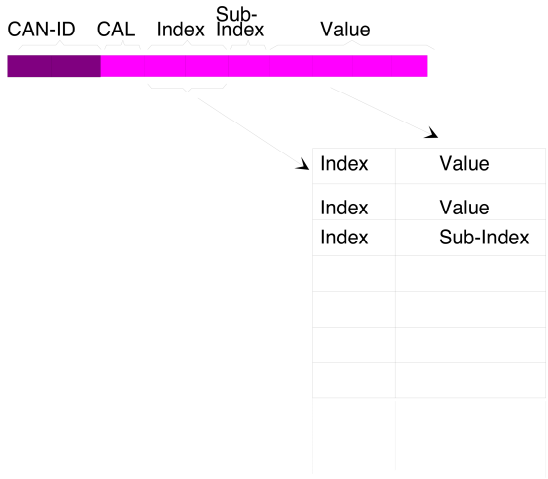
Dynamic ID distribution

600 different IDs are permitted for event-controlled data. When adjusting the ID over a switch, we have to limit ourselves to an inferior quantity or we shall need too much space for the adjustments (DIP-switch with 10 switches). In case we also want to allocate the IDs for inputs and outputs to different priority classes, a sensible adjustment over the hardware is not possible any longer.

For the dynamic allocation of IDs, we need a DBT master which allocates to the module the IDs of the required priority class when starting. Thus the allocation is extremely flexible and it allows the user to set high priority messages where it's needed.

The service channel.

Herewith our module is ready for standard applications. However, very soon we might want to be able to make certain adjustments and will need additional information from the module concerning internal data.



It would not make sense to allocate a special ID for each one of our requirements. With regard to CANopen we work, apart from the normal data exchange (this type of data is also called process-data-objects, PDOs), with a so-called service channel (service-data-object, SDO). To transmit service messages, we need only one pait of IDs per device. These IDs must have the priority class 6 (1321 - 1540). According to the CAL definition, the first byte is used for domain protocol. The second and the third byte of the SDO are used as additional index and the fourth one as a subindex. Via the index and the subindex, it is possible to address a huge chart (called object dictionary), from which we can read and write our values.

6000	Device Type	Unsigned 32
6001	Manufacturers Device Name	Vis. String
6002	'1.0' Hardware Version	Vis. String
6003	'1.0' Software Version	Vis. String
6004	Commanded Bus State	Unsigned 8
6005	Actual Bus State	Unsigned 8
6100	Receive POD001	POD Map.
6101	Receive POD002	POD Map.
6164	Transmit POD001	POD Map.
6165	Transmit POD002	POD Map.
6400	8 Input Lines	Unsigned 8
6401	8 Output Lines	Unsigned 8
6502	Interrupt Mask	Unsigned 8
6505	Input Filter	Unsigned 8
6506	Output mode on bus error	Unsigned 8
6507	Output state on bus error	Unsigned 8

The communication and appliance profile specifies precisely from where data may be taken and where data may be modified.

Certainentries in the dictionary are specified by the communication profile and others are defined by the different device profiles. There is always a distinction between mandatory, optional and manufacturer-specific registrations to keep the profiles as flexible as possible

Example of a Object Dictionary

The following chart refers to a typical I/O module. The registrations between 1000 and 1FFF are defined by the communication profile. They are common for all appliances with CANopen and contain the most important general data.

The registrations with the indices 6400 and 6401 contain input, res. output data of an I/O module. An appliance with a profile for I/O modules must always put data at disposal under this index. Herewith, one must make sure that with regard to any appliance, I/O data is in the same place, independent of the device manufacturer.

Example

The following example describes the input definition of the turn-on attenuation via the service channel. The respective registration figures under the index number 6505 and has the designation "input configuration". The sub-index permits more accurate configuration of the adjustments. Thus, more information about the sub-index setup is required.

Sub-index of the object 6505

Indicates on which inputs an additional filtering time constant must be switched .

Object description

If we intend to switch on the input attenuation of the first eight inputs, we have to overwrite the value of index 6505 / sub-index 2 with the value 225. An additional message is required for each group of eight inputs.

INDEX	6505h
Variable Name	Input Configuration
Object Code	8 (Array)
NR. of Elements	8
Data Type	Unsigned 8
Length	1

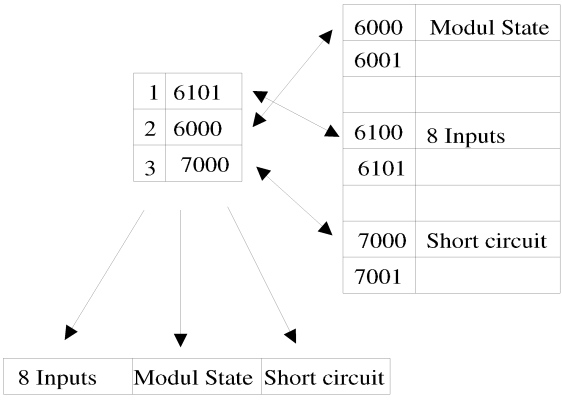
Description of the Sub-Index

Sub - Index	0
Description	NR of Blocks
Value range	Unsigned 8
Mandatory Range	0 ... 8

Sub - Index	1
Description	Filter for inputs 0 ... 7
Value range	Unsigned 8
Mandatory Range	0 ... 255

Data mapping

Up to this point our module has only transmitted, res. received the values of inputs and outputs over a PDO (data channel). For additional data we depend on the service channel. But it is also possible to allocate data other than the values of inputs to the PDO. As there are only 16 channels in our module up to now we may, for instance, define in the third and fourth data byte which of the 16 inputs has been modified since the last data message. This data allocation to a PDO according to requirements is named "mapping". In a so-called „mapping structure“ the user can define the cross-reference between the values in the object dictionary and the position of the value in the PDO message.



IO_PDO_001

As default, the PDOs have a standard setting (predefined default mapping), which is defined in the device profile and guaranties an identical behaviour off all devices.

Dynamic PDOs

If we extend the I/O module to 64 inputs, there is no PDO capacity for an additional message, thus we need an additional PDO. CANopen also allows this extension.

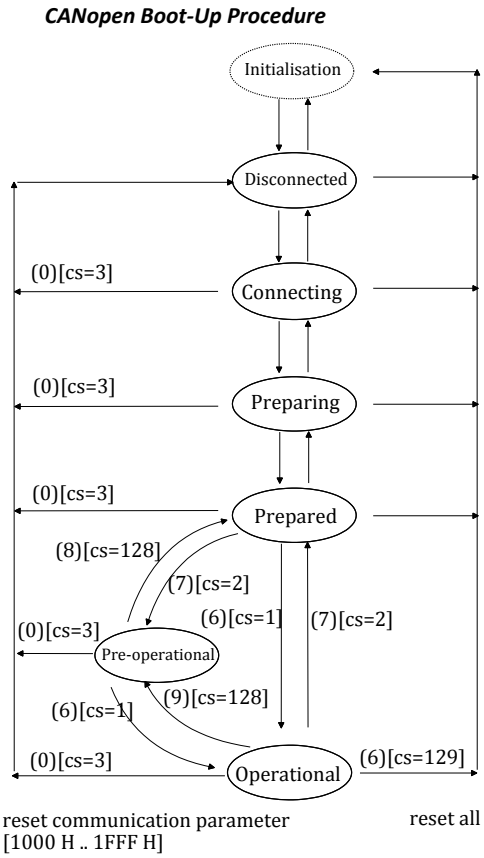
A module with a flexible number of PDOs will announce itself only with its service channel when performing full CAL bootup (if supported) and the DBT master will therefore allocate only one pair of IDs for this object.

The configuration master can ask for all data of the module over the service channel and herewith find out easily which possibilities the module offers. Based on the information received, the configuration master can then determine the required PDOs and allocate the ID numbers to the I/O module either via SDO or via DBT, if supported. After this "extended" starting phase, the I/O module configured in this manner will then be integrated into the network.

For the present example we will configure an 8 byte PDO for 64 inputs as well as a PDO with status information concerning short-circuit and supply voltage.

The boot-up

As we have seen, that certain adjustments and configurations may take place while starting the network. During this period the DBT, res. the configuration master prset all the relevant parameters of the appliances connected and guarantees safe and reliable communication..



To support CAL- and simple CANopen devices in a single network, CANopen supports the CAL boot up procedure with a single extension, the additional

state „pre-operational“. CAL-devices will proceed from the state „prepared“ directly to the state „operational“, simple devices with DBT support or devices which store there settings in a non volatile memory are directly in the state „pre-operational“ after power-on and can be switched then to the common state „operational“.

Advantages and disadvantages of CANopen

CANopen is the result of a European project named ASPIC. The crucial point of the study was the range of drive networks via the CAN-Bus. The revision made by the CiA has enlarged the field of applications.

However, applications including motors still have some priority.

The starting behaviour and configuration changes of nodes in the course of operation is a severe problem for all the field buses Practical work will show to what extent the CANopen concept will be successful.

For most of the appliance manufacturers CAL does not represent a satisfactory solution as the variability leaves too much leeway, thus no "plug-and-play" solution can be offered to the user. During the HMI '95 several companies have decided to make CANopen become reality

CANopen is being supported by the initiative of many medium sized companies. Therefore the customer will not depend on a dominant supplier but, on the other hand, there is no big, financially potential promoter.