# Object Dictionary (OD) and Electronic Data Sheet (EDS)

One of the most important properties of CANopen is the provision of a standardized device description in the form of the so-called "object dictionary".   
   
This is based on a "table" (**device object dictionary**), which has the same structure for all types of devices. With this it is possible to access all important data, parameters and functions of a device using a kind of logical addressing system (index, subindex) from the "outside", i.e. via the CAN bus. Access to the OD is provided by means of the SDO protocol.   
   
In addition to the standardized description of the communication properties of devices under DS 301, CANopen defines so-called "device profiles" for typical devices of automation technology. These specify the most important parameters, data and functions per device type (e.g. input/output modules, drives, encoders, etc). In this way the use of CANopen devices becomes more independent of device manufacturers, as all CANopen-compatible devices are interchangeable, at least in terms of the standardized basic functionality.   
   
An entry in the object dictionary (OD object) is addressed via a (hexadecimal) index and can have up to 255 sub-entries (subindex). Sub-entries are normally used to combine values of the same type, such as with an array, or to access connected values, such as with a data record. An OD object therefore has a 24-bit address, made up of the 16-bit index and 8-bit subindex.   
   
The object dictionary is sub-divided into areas of 4096 entries each, which are standardized. The range between 1000 and 1FFF is called communication area/communication profile, and is specified in detail in DS-301. The area from 2000 to 5FFF contains the manufacturer-specific device objects. The area between 6000 and 9FFF is standardized in the so-called device profiles, of which more later. The area between A000 and AFFF contains the so-called network variables (NWV); these dynamic entries (dynamic channels), which are generated if required, e.g. in an PLC-program. All other areas are still unused.   
   
Behind every OD object there is a value, which can be read or written with the aid of a SDO transfer. With this a device can be configured and controlled, as an object dictionary entry can also directly represent a property or a function of the device (e.g. start-up of a drive). With read access to an object dictionary entry, the device returns the value of the entry. The data type and meaning of the value must be known to the enquirer. For this reason there is the **"electronic data sheet"** (EDS), which describes each object dictionary entry with address (main-/subindex), parameter name, data type, access type and default value.   
   
An EDS is an ASCII file in WIN.INI format, which is used by CANopen configuration tools to allow the user to configure devices. The data types used in CANopen are the usual bytes, words and double words, with and without signs in each case. In addition there are ASCII and Unicode strings (visible\_string, unicode\_string). There is a one-bit Boolean data type, as well as four-byte and eight-byte floating point types in accordance with IEEE 754-1985. Time\_of\_day is used to display of time. Time\_of\_day is a structure that counts the milliseconds since midnight in 28 bits and the days since 01 January 1984 in sixteen bits. The other four bits in between up to the byte limit are reserved. Finally there is the byte stream of undefined length, which is referred to as a domain in CANopen.   
   
The communication area of the object dictionary contains all parameters that concern the communication behavior.   
   
Although the object dictionary is mandatory for CANopen devices, only a few entries themselves are mandatory; these are the entries [1000sub00], [1001sub00], [1018sub00] and [1018sub01].   
   
Here are the objects of the communication area in table form:   
 

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Main index** | **Object** | **Name** | **Data type** | **Access type** | | 1000 | Variable | device type | DWORD | ro | | 1001 | Variable | error register | BYTE | ro | | 1002 | Variable | manufacturer status register | DWORD | ro | | 1003 | Array | pre-defined error field | DWORD | rw | | 1005 | Variable | COBID of Sync object | DWORD | rw | | 1006 | Variable | communication cycle period | DWORD | rw | | 1007 | Variable | synchronous window length | DWORD | rw | | 1008 | Variable | device name | string | const | | 1009 | Variable | hardware version | string | const | | 100A | Variable | software version | string | const | | 100C | Variable | guard time | WORD | rw | | 100D | Variable | life time factor | BYTE | rw | | 1010 | Array | store parameters | DWORD | rw | | 1011 | Array | restore default parameters | DWORD | rw | | 1012 | Variable | COBID of TimeStamp object | DWORD | rw | | 1013 | Variable | high resolution time stamp | DWORD | rw | | 1014 | Variable | COBID of emergency object | DWORD | rw | | 1015 | Variable | inhibit time for emergency object | WORD | rw | | 1016 | Array | consumer heartbeat time | DWORD | rw | | 1017 | Variable | producer heartbeat time | WORD | rw | | 1018 | Record | identity object | DWORD | ro | | 1200 ... 127F | Record | 1st ... 128th server SDO | SDO parameter | rw | | 1280 ... 12FF | Record | 1st ... 128th client SDO | SDO parameter | rw | | 1400 ... 15FF | Record | 1st ... 512th receive PDO | PDO parameter | rw | | 1600 ... 17FF | Array | 1st ... 512th receive PDO mapping | PDO mapping | rw | | 1800 ... 19FF | Record | 1st ... 512th transmit PDO | PDO parameter | rw | | 1A00 ... 1BFF | Array | 1st ...512th transmit PDO mapping | PDO mapping | rw | |

   
As can be seen, almost all CANopen communication objects are found in the object dictionary, namely node monitoring in [100C], [100D] and [1017], signaling of emergency situations in [1014], [1015], [1001] and [1003], processing of the time stamp in [1012], processing of the synchronization message in [1005], [1006] and [1008], device access via SDO in [1200]ff, and finally process data exchange via PDOs in [1400] to [1BFF].   
   
With object dictionary entries that are data records or areas, there is a special feature on subindex 0, as this defines the subindex of the last valid element as a BYTE value - in most cases this value is synonymous with the number of elements. The elements themselves are provided in subindex1 to subindex <value of subindex 0>. Subindex 0 is particularly useful in the case of arrays of variable size, for example with PDO mapping.