CANopen tables

ifm electronic

1 Annex

Contents

1664

Additionally to the indications in the data sheets you find summary tables in the annex.

1.1 CANopen tables

Contents

9941

The following tables will inform you about important values and settings of the CANopen interfaces.

1.1.1 IDs (addresses) in CANopen

3952

In CANopen there are different types of addresses (IDs):

- COB ID
  The **C**ommunication **Ob**ject **Id**entifier addresses the message (= the communication object) in the list of devices. A communication object consists of one or more CAN messages with a specific functionality, e.g.
  - PDO (**P**rocess **D**ata **O**bject = message object with process data),
  - SDO (**S**ervice **D**ata **O**bject.= message object with service data),
  - emergency (message object with emergency data),
  - time (message object with time data) or
  - error control (message object with error messages).
- CAN ID
  The **CAN Id**entifier defines CAN messages in the complete network. The CAN ID is the main part of the arbitration field of a CAN data frame. The CAN ID value determines implicity the priority for the bus arbitration.
- Download ID
  The download ID indicates the node ID for service communication via SDO for the program download and for debugging.
- Node ID
  The **Node Id**entifier is a unique descriptor for CANopen devices in the CAN network. The Node ID is also part of some pre-defined connectionsets (→ function code (→ page 4)).

Comparison of download-ID vs. COB-ID:

Controller program download		CANopen	
Download-ID	**COB-ID SDO**	**Node ID**	**COB-ID SDO**
1…127	TX: 580_{16} + download ID	1…127	TX: 580_{16} + node ID
	RX: 600_{16} + download ID		RX: 600_{16} + node ID

TX = slave sends to master
RX = slave receives from master

1.1.2 Structure of CANopen messages

Contents

9971

A CANopen message consists of the COB ID and up to 8-byte data:

COB ID			DLC	Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7		Byte 8	
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Details are given in the following chapters.

NOTE: Please note the reversed byte order!

Examples:

Value [hex]	Data type	Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7		Byte 8	
12	BYTE	1	2	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1234	WORD	3	4	1	2	–	–	–	–	–	–	–	–	–	–	–	–
12345678	DWORD	7	8	5	6	3	4	1	2	–	–	–	–	–	–	–	–

Structure of the COB ID

9972

The first part of a message is the COB ID. Structure of the 11-bit COB ID:

Nibble 0				**Nibble 1**				**Nibble 2**			
11	10	9	8	7	6	5	4	3	2	1	0
--	**3**	**2**	**1**	**0**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--	**function code**				**node ID**						

The COB ID consists of the function code (→ page 4) and the node ID.

Example:
Communication object = TPDO1 (TX)
Node number of the device = 20_{16} = 32_{10}

Calculation:
Function code for the communication object TPDO1 = 3_{16}
Significance of the function code in the 11-bit COB ID = 3_{16} x 80_{16} = 180_{16}
Add the node number (20_{16}) ⇒ the COB ID is: $1A0_{16}$

1				**A**				**0**			
3	2	1	0	3	2	1	0	3	2	1	0
0	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
--	3_{16} = 3_{10}				20_{16} = 32_{10}						

Function code / Predefined Connectionset

9966

In the "CANopen Predefined Connectionset" some function codes are predefined.

When using the predefined connectionset you can operate a CANopen network of up to 127 participants without the risk of a double assignment of COB IDs.

Broadcast or multicast messages:

Communication object	Function code [hex]	COB ID [hex]	Related parameter objects [hex]
NMT	0	000	
SYNC	1	080	1005, 1006, 1007, 1028
TIME	2	100	1012, 1013

Point-to-point messages:

Communication object	Function code [hex]	COB ID [hex]	Related parameter objects [hex]
EMERGENCY	1	080 + node ID	1014, 1015
TPDO1 (TX)	3	180 + node ID	1800
RPDO1 (RX)	4	20016 + node ID	1400
TPDO2 (TX)	5	280 + node ID	1801
RPDO2 (RX)	6	30016 + node ID	1401
TPDO3 (TX)	7	380 + node ID	1802
RPDO3 (RX)	8	400 + node ID	1402
TPDO4 (TX)	9	480 + node ID	1803
RPDO4 (RX)	A	500 + node ID	1403
Default SSDO (TX)	B	58016 + node ID	1200
Default CSDO (RX)	C	60016 + node ID	1280
NMT Error Control	E	70016 + node ID	1016, 1017

TX = slave sends to master
RX = slave receives from master

SSDO = server SDO
CSDO = client SDO

SDO command bytes

9968

Structure of an SDO message:

COB ID	DLC	Command	Index		Sub-index	Data *)			
XXX	8	byte	byte 0	byte 1	byte	byte 0	byte 1	byte 2	byte 3

*) depending on the data to be transmitted

NOTE: Please note the reversed byte order!

An SDO COB ID consists of:

CANopen	
Node ID	**COB ID SDO**
1…127	TX: 580_{16} + node ID
	RX: 600_{16} + node ID

TX = slave sends to master
RX = slave receives from master

DLC (data length code) indicates the number of the data bytes (for SDO: DLC = 8).

SDO command bytes:

Command hex | dec		Message	Data length	Description
21	33	request	more than 4 bytes	send data to slave
22	34	request	1…4 bytes	send data to slave
23	35	request	4 bytes	send data to slave
27	39	request	3 bytes	send data to slave
2B	43	request	2 bytes	send data to slave
2F	47	request	1 byte	send data to slave
40	64	request	---	request data from slave
42	66	response	1…4 bytes	send data from slave to master
43	67	response	4 bytes	send data from slave to master
47	71	response	3 bytes	send data from slave to master
4B	75	response	2 bytes	send data from slave to master
4F	79	response	1 byte	send data from slave to master
60	96	response	---	data transfer ok: send confirmation of receipt from slave to master
80	128	response	4 bytes	data transfer failed send abort message from slave to master → chapter SDO abort code (→ page 6)

SDO abort code

9970

NOTE: The SDO abort code is NOT part of the emergency message!

Abord code [hex]	Description
0503 0000	toggle bit not alternated
0504 0000	SDO protocol timed out
0504 0001	client/server command specifier not valid or unknown
0504 0002	invalid block size (block mode only)
0504 0003	invalid sequence number (block mode only)
0504 0004	CRC error (block mode only)
0504 0005	out of memory
0601 0000	unsupported access to an object
0601 0001	attempt to read a write only object
0601 0002	attempt to write a read only object
0602 0000	object does not exist in the object dictionary
0604 0041	object cannot be mapped to the PDO
0604 0042	the number and length of the objects to be mapped would exceed PDO length
0604 0043	general parameter incompatibility reason
0604 0047	general internal incompatibility in the device
0606 0000	access failed due to an hardware error
0607 0010	data type does not match, length of service parameter does not match
0607 0012	data type does not match, length of service parameter too high
0607 0013	data type does not match, length of service parameter too low
0609 0011	sub-index does not exist
0609 0030	value range of parameter exceeded (only for write access)
0609 0031	value of parameter written too high
0609 0032	value of parameter written too low
0609 0036	maximum value is less than minimum value
0800 0000	general error
0800 0020	data cannot be transferred or stored to the application
0800 0021	data cannot be transferred or stored to the application because of local control
0800 0022	data cannot be transferred or stored to the application because of the present device state
0800 0023	object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error)

1.1.3 Bootup messsage

9961

After booting the CAN participate sends the boot-up message once:

hex	dec
700_{16} + node ID	$1\ 792_{10}$ + node ID

The participant is now capable of communicating in the CAN network.

Structure:
The node ID of the participant is $7D_{16} = 125_{10}$.
The boot-up message is: $77D_{16} = 1\ 917_{10}$

1.1.4 Network management (NMT)

Contents

9974

Network management commands

9962

With the following network management commands the user can influence the operating mode of individual or all CAN participants. Structure:

Byte 1	Byte 2	Byte 2
COB ID	command	node ID

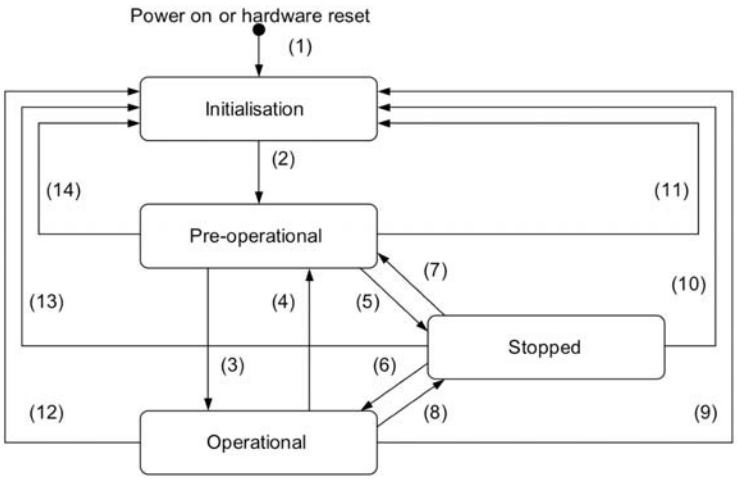
Node ID = 00 ⇒ command valid for all nodes in the network at the same time

COB ID	NMT command		Description	
00	$01_{16} = 01_{10}$	node ID	start_remode_node	start CAN participate
00	$02_{16} = 02_{10}$	node ID	stop_remode_node	stop CAN participate
00	$80_{16} = 128_{10}$	node ID	enter_pre-operational	switch to pre-operational
00	$81_{16} = 129_{10}$	node ID	reset node	reset CAN participate
00	$82_{16} = 130_{10}$	node ID	reset communication	reset CAN communication

NMT state

9963

The status byte informs about the state of the CAN participant.



Graphics: state transitions under CANopen

Permitted transitions:

(1) State is automatically reached at power on

(2) Internal initialisation completed – node automatically goes to PRE-OPERATIONAL

(3) NMT service "Start Remote Node Indication"

(4) + (7) NMT service "Enter PRE-OPERATIONAL Indication"

(5) + (8) NMT service "Stop Remote Node Indication"

(6) NMT service "Start Remote Node Indication"

(9)...(11) NMT service "Reset Node Indication"

(12)...(14) NMT service "Reset Communication Indication"

NMT state for CANopen master

9964

State hex	dec	Description
00	0	not defined
01	1	Master waits for a boot-up message of the node. OR: Master waits for the expiry of the given guard time.
02	2	- Master waits for 300 ms. - Master requests the object 1000_{16}. - Then the state is set to 3.
03	3	The master configures its slaves. To do so, all SDOs generated by the configurator are transmitted to the slaves one after the other: - The Master sends to the slave a SDO read request (index 1000_{16}). - The generated SDOs are compressed into a SDO array. - The slave knows it's first SDO and the number of it's SDOs.
05	5	After transmission of all SDOs to the slaves the master goes to state 5 and remains in this state. State 5 is the normal operating state for the master.

To read the node state out of the FB:

Used function block	Node state is found here
CANx_MASTER_STATUS CANx_SLAVE_STATUS	output NODE_STATE
CANOPEN_GETSTATE	output NODESTATE

NMT state for CANopen slave

9965

State hex	dec	Description
FF	-1	The slave is reset by the NMT message "Reset Node" and automatically goes to state 1.
00	0	not defined
01	1	state = waiting for BOOTUP After max. 2 s or immediately on reception of its boot up message the slave goes to state 2.
02	2	state = BOOTUP After a delay of 0.5 s the slave automatically goes to state 3.
03	3	state = PREPARED The slave is configured in state 3. The slave remains in state 3 as long as it has received all SDOs generated by the configurator. It is not important whether during the slave configuration the response to SDO transfers is abort (error) or whether the response to all SDO transfers is no error. Only the response as such received by the slave is important – not its contents. If in the configurator the option "Reset node" has been activated, a new reset of the node is carried out after transmitting the object 1011_{16} sub-index 1 which then contains the value "load". The slave is then polled again with the upload of the object 1000_{16}. Slaves with a problem during the configuration phase remain in state 3 or directly go to an error state (state > 5) after the configuration phase.
04	4	state = PRE-OPERATIONAL A node always goes to state 4 except for the following cases: • it is an "optional" slave and it was detected as non available on the bus (polling for object 1000_{16}) OR: • the slave is present but reacted to the polling for object 1000_{16} with a type in the lower 16 bits other than expected by the configurator.
05	5	state = OPERATIONAL State 5 is the normal operating state of the slave: [Normal Operation]. If the master was configured to [Automatic startup], the slave starts in state 4 (i.e. a "start node" NMT message is generated) and the slave goes automatically to state 5. If the flag GLOBAL_START was set, the master waits until all slaves are in state 4. All slaves are then started with the NMT command [Start All Nodes].
61	97	A node goes to state 97 if it is optional (optional device in the CAN configuration) and has not reacted to the SDO polling for object 1000_{16}. If the slave is connected to the network and detected at a later point in time, it is automatically started. To do so, you must have selected the option [Automatic startup] in the CAN parameters of the master.
62	98	A node goes to state 98 if the device type (object 1000_{16}) does not correspond to the configured type.
63	99	In case of a nodeguarding timeout the slave is set to state 99. As soon as the slave reacts again to nodeguard requests and the option [Automatic startup] is activated, it is automatically started by the master. Depending on the status contained in the response to the nodeguard requests, the node is newly configured or only started. To start the slave manually it is sufficient to use the method [NodeStart].

Nodeguard messages are transmitted to the slave ...
- if the slave is in state 4 or higher AND
- if nodeguarding was configured.

To read the node state out of the FB:

Used function block	Node state is found here
CANx_MASTER_STATUS CANx_SLAVE_STATUS	output NODE_STATE
CANOPEN_GETSTATE	output NODESTATE

CANopen status of the node

1973

Node status according to CANopen (with these values the status is also coded by the node in the corresponding messages).

Status hex	dec	CANopen status	Description
00	0	BOOTUP	Node received the boot-up message.
04	4	PREPARED	Node is configured via SDOs.
05	5	OPERATIONAL	Node participates in the normal exchange of data.
7F	127	PRE-OPERATIONAL	Node sends no data, but can be configred by the master.

If nodeguarding active: the most significant status bit toggles between the messages.

Read the node status from the function block:

Function block used	Node status is found here
CANx_MASTER_STATUS CANx_SLAVE_STATUS	Structure element LAST_STATE from the array NODE_STATE_SLAVE
CANOPEN_GETSTATE	Output LASTNODESTATE

1.1.5 CANopen error code

Contents

9967

Emergency messages

9973

Device errors in the slave or problems in the CAN bus trigger emergency messages:

COB ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80_{16} + node ID		error code		object 1001_{16}	device-specific				

NOTE: Please note the reversed byte order!

Overview CANopen error codes

8545

Error Code (hex)	Meaning
00xx	Reset or no error
10xx	Generic error
20xx	Current
21xx	Current, device input side
22xx	Current inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains voltage
32xx	Voltage inside the device
33xx	Output voltage
40xx	Temperature
41xx	Ambient temperature
42xx	Device temperature
50xx	Device hardware
60xx	Device software
61xx	Internal software
62xx	User software
63xx	Data set
70xx	Additional modules
80xx	Monitoring
81xx	Communication
8110	CAN overrun-objects lost
8120	CAN in error passiv mode
8130	Life guard error or heartbeat error
8140	Recovered from bus off
8150	Transmit COB-ID collision
82xx	Protocol error
8210	PDO not procedded due to length error
8220	PDO length exceeded
90xx	External error
F0xx	Additional functions
FFxx	Device specific

Object 0x1001 (error register)

8547

This object reflects the general error state of a CANopen device. The device is to be considered as error free if the object 1001_{16} signals no error any more.

Bit	Meaning
0	generic error
1	current
2	voltage
3	temperature
4	communication error
5	device profile specific
6	reserved – always 0
7	manufacturer specific

For an error message more than one bit in the error register can be set at the same time.

Example: CR2033, message "wire break" at channel 2 (→ installation manual of the device):

COB-ID	DLC	Byte 0	Byte 1	Byte	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
80_{16} + node ID		**00**	**FF**	**81**	**10**	**00**	00	00	00

Error-Code = $FF00_{16}$

Error register = 81_{16} = $1000\ 0001_2$, thus it consists of the following errors:
- generic error
- manufacturer specific

Concerned channel = 0010_{16} = $0000\ 0000\ 0001\ 0000_2$ = wire break channel 2