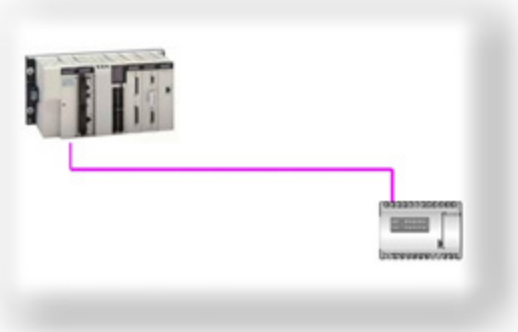
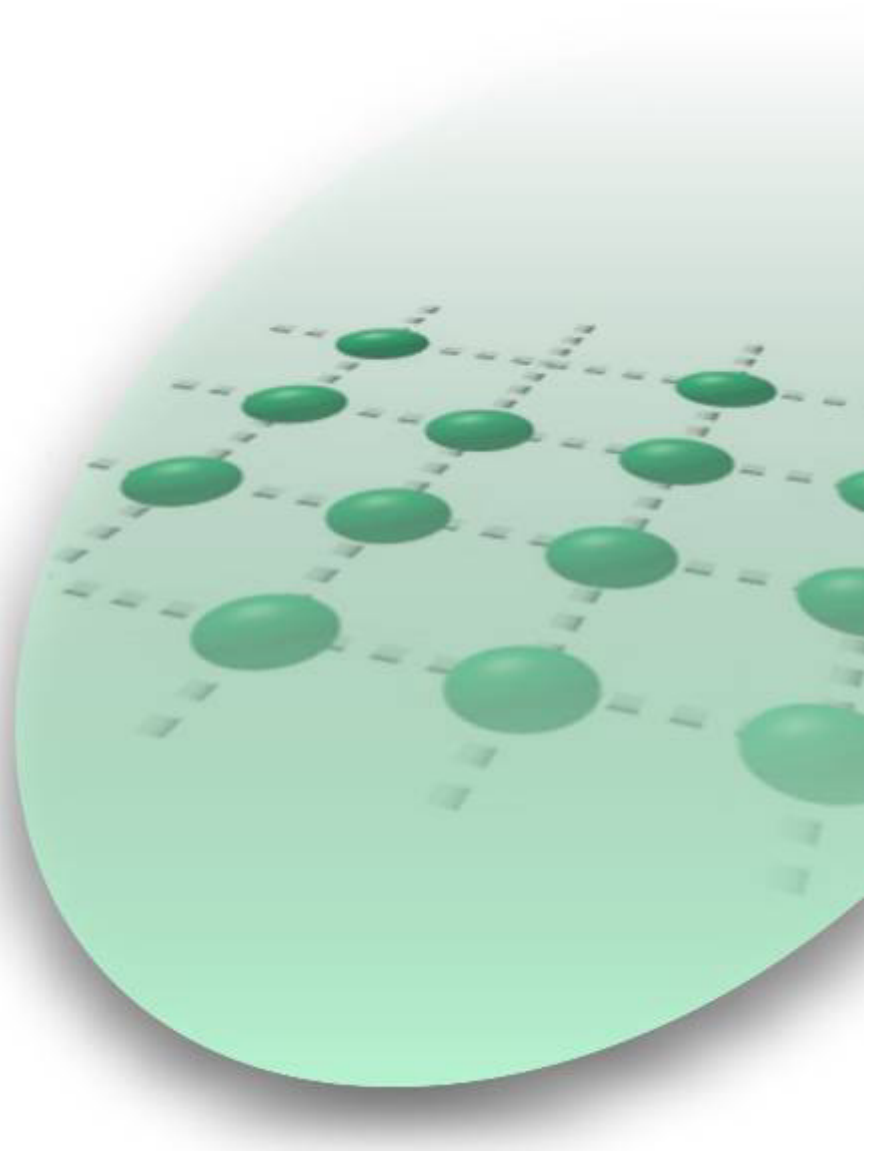
CANopen Networkparameters
Influencing factors and Optimisation
System Manual Document
with Optimisation Examples for Premium and Micro





33003588.00

Merlin Gerin
Square D
Telemecanique

Schneider Electric
Building a New Electric World

Contents

Introduction

The System Manual Document is intended to be a “Quickstart Manual for a System”.
It is **not** intended to replace any specific product documentation. Instead, it offers further information to the product documentation, for installing, configuring and starting up the system.

The functional description or the specification for a specific user application is **not** part of this manual. Nevertheless, the document outlines some typical applications where the system might be implemented.

Versioning

The examples are based on the following hard- and software versions:

Software:	Sycon V2.9
Hardware:	-

Bus Capacity in CANopen Networks

Controlling data transmission in CANopen Networks

As devices on a CANopen network can transmit their allocated data at any time they wish without waiting for a data request, the data transmission on a CANopen bus characteristically consists of phases of intensive telegram traffic followed by large and small pauses inbetween. Possible collisions are recognised and solved before errors occur, in that according to defined rules (arbitration), one sender always prevails and other senders must wait for the appropriate moment when the bus is free to make another attempt. Thus, all devices can send data without a collision ocurring.

The advantage of this method is obvious; for normal telegram traffic no master is required to request information from a specific device. Since all devices have a right to send data at any time and no collisions can occur, this saves time. However, during times of intensive data traffic, certain devices may never win the arbitration and are never allowed to send their data. This causes a traffic jam that can only be solved by reducing the telegram traffic on the bus.

The method is comparable to normal vehicular traffic on the streets. Vehicle owners use their vehicles as and when they please. Collisions at junctions are avoided using layed down priority rules. If the traffic increases, those vehicles wishing to move from side roads on to the main road are slowed down or halted and you end up with a traffic jam.

Bus load

Bus load is a measure of the telegram traffic. It is defined as the quotient from the actual time requirement for all telegrams sent, divided by the unit of time used – normally seconds. A bus load of 50% means for example, that only half the number of telegrams has been sent that the system would normally allow. A problem free bus load should normally be, on average, less than 70%.

Bus load optimsation

The bus load must be considered when configuring CANopen networks. It depends on the bus load whether a device can send its data or whether certain telegrams are contually displaced and cause a traffic jam.

The bus load can be optimised by carefully choosing the right network parameters. The following chapter describes the influence these parameters can have on the bus load and give recommendations as to the most sensible values.

Runtime of a CAN Telegram

To estimate the bus load, the runtime for CAN telegrams is required. Depending on the bit rate and PDU length, the following maximum values (in milliseconds) can be deduced:

Bit rate	User data (PDU-Length)								
[kBit/s]	0	1	2	3	4	5	6	7	8
50	1.09	1.28	1.47	1.66	1.86	2.05	2.24	2.34	2.62
125	0.44	0.51	0.59	0.67	0.74	0.82	0.90	0.79	1.05
250	0.22	0.26	0.29	0.33	0.37	0.41	0.45	0.49	0.52
500	0.11	0.13	0.15	0.17	0.19	0.21	0.22	0.24	0.26
1000	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13

Factors Influencing the Bus Load

General

The bus load of a CANopen network depends on the following factors:

- Bitrate
- Number of PDOs per unit time
- Amount of user data in a telegram
- Number of remote requests
- Number of SDOs at runtime
- Sync Interval
- Type of error protocol (Node Guarding vs. Heartbeat)
- Quality of transmission / the physical bus

Bitrate

Introduction

The bit rate (transmission rate) has a direct influence on the bus load. Doubling the bit rate of a telegram reduces the run time for a telegram by 50% and so reduces the bus load (indirectly proportional) by 50% also. The user, however, cannot randomly select a bit rate as this is limited by the type of cable used and its length.

Cable Types

Schneider Electric supplies the following cable types:

- TSXCANCAxxx (CE certified, standard cable for the european market)
- TSXCANCBxxx (UL certified, standard cable for the U.S. market)
- TSXCANCDxxx (Cable for heavy mechanical loads, e.g. in drag chains)

Signal	Colour	Diameter
CAN_H, CAN_L	white	0.25 mm² (AWG24)
CAN_GND, V+	black, red	0.34 mm² (AWG22)

max. Cable Length

The following table shows the dependancy of the bit rate on the cable length for the cables listed above:

Bit rate [kBit/s]	1000	800	500	250	125	50	20	10
max. Length	20 m	25 m	100m	250m	500m	1000m	2500m	5000m

The CANopen manual often refers to 40 meters as the maximum length cable for 1 Mbit transmision rate. This value, however, does not take into account the electrical isolation of the bus interface to the device, a standard with Schneider Electric products. Taking the electrical isolation into account, the maximum calculated cable length is 4 meters. Field tests however have shown that 20 meters is a practical length provided the cable has no branches or other negative effects.

max. Branch Length

CANopen networks allow branching. These are connected via a branch connector (TAP) which is inserted into the main bus. The TAPS must be spread at minimum distances along the bus as shown below:

Bit rate [kBit/s]	1000	800	500	250	125	50	20	10
Single Branch	0.3 m	3 m	5 m	5 m	5 m	60 m	150 m	300 m
Sum per TAP	0.6 m	6 m	10 m	10 m	10 m	120 m	300 m	600 m
min. TAP Spacing*		3.6 m	6 m	6 m	6 m	72 m	180 m	360 m
total	1.5 m	15 m	30 m	60 m	120m	300m	750m	1500m

* can be calculated for each TAP. TAP Spacing = 60% of the total of all branches in the TAP

Number of PDOs per Unit of Time

The number of PDOs per unit of time depends on:

- The total number of PDOs configured for the bus.
- The frequence of each PDO (Transmission Type)

The number of PDOs configured for the bus is the sum of all TPDOs and RPDOs for all the connected devices. Every one of these PDOs has a send trigger which is normally defined by the transmission type, which also defines the transmission frequency.

Both factors must be considered together and are the main influencing factor on the bus load. Reasonable values here play a large role in bus optimisation.
An analog value (PDO with 2 bytes) which is transmitted every 5 milliseconds at a bit rate of 250kbits/sec generates a bus load of 5.8 % (200 ∗ 0.29 / 10). A more sensible analog value (used for e.g. for temperature or tank gauges) with a 100msec transmission interval only causes a 0.29% (10 ∗ 0.29 / 10) increase in the bus load.

User Data Length

The CAN telegram consists of up to 8 bytes of user data (PDU=Process Data Unit) with an overhead of 44 bits. The effects on the bus load are shown in the runtime table in the section „Runtime of a CAN telegram“, above.

The table shows that the bus load incurred by a PDO with 8 bytes of user data is about twice the amount of a PDO with only 2 bytes of user data. The CANopen advantage of having the PDOs data orientated instead of device orientated can clearly be seen here in the positive effect it has on the bus load.

Data orientated means that instead of transmitting all the data of a device in a single telegram all at once (as with Profibus), the complete device data can be split over several PDOs or even reduced to a sub set of data in a single PDO. These PDOs can have different transmission types and frequencies.

Choice of Transmission Type

General

The transmission type (TT) defines when a CANopen device is allowed to send its data. The transmission type can be synchronous, after a data change or only on remote request. User-specified transmission types are also allowed but are, however, not considered here.

- With sychronous transmission, a CANopen device can only transmit after receiving one (or more) Sync telegrams.
- In the case of data change, the device sends its data when a single bit in the data field of a PDO changes. This transmission type has extra parameters; the inhibit time and the event time. These parameters define the time window in which the PDO may be sent. The inhibit time fixes the minimum time between transmissions and the event time the maximum.
- Remote request means that a device only transmits its data when requested.

The remote request should be avoided where possible as it automatically increases bus load. Also, the remote request has no standard format. Some manufacturers set the data length in the request frame to zero instead of inserting the amount of data to be returned by the device. The remote request can also lead to collisions if the request goes out to 2 devices at the same time. In this case the COB ID is identical and the RTR bit is set. The collision takes place after the arbitration field in the telegram frame and cannot be recognised. For these reasons the remote request will not be considered further.

Effect of the Event Timer and Inhibit Time on TPDO Transmission

Inhibit Time [ms]	Event Time[ms]	Data Change [ms] (one change)	Transmission of TPDO after [ms]
0	0	0	-
0	0	50	50
0	100	0	alle 100
0	100	50	50, 150, 250, then every 100 ms
0	100	150	100, 150, 250, then every 100 ms
100	0	0	-
100	0	50	100
100	0	100	100
100	0	150, 170	150, 250
200	≤200	regular	every 200 ms
200	300	150	200, 400, 600, 900, then every 300 ms
200	300	400	300, 500, 700, 900, then every 300 ms

Example: Synchronous vs. Asynchronous Transmission

Assume we have a CANopen network with 250 kbit/sec bit rate with 2 devices, each with a PDO with 8 user data bytes to transmit, but each has a different transmission type. Device A transmits synchronously, device B, asychronously on data change.

To make a reasonable comparison, assume that both devices transmit with a 50 millisec. period (i.e. the Sync interval is 50 millisec. and in device B the data changes every 50millisecs.)

The cycle for device A consists of a Sync telegram and a PDO as response. The Sync is a CAN telegram with data length zero and takes 0.22ms. The time for a PDO with 8 bytes is 0.52ms. i.e. the cycle time for device A is 0.74 ms

The cycle time for device B only consists of the PDO and lasts 0.52ms.

It would appear that the synchronous transmission, in comparison to the asychronous, increases the bus load by 50%.

This comparison is not completely true for a larger network, however. The Sync telegram is sent to the whole network, not just a single device. If we change the setup so that each device sends 4 PDOs instead of one, we get the following results:

- Device A: 1 Sync + 4 PDOs -> Cycle time = 2.3 ms
- Device B: 4 PDOs -> cycle time = 2.08 ms

Synchronous device A transmits every 50ms, i.e. 20 times per second. Device A occupies the bus for 46ms per second which results in a bus load of 4.6%. Device B on the other hand, sending data only when it changes, only causes a bus load of 4.16%

This shows that with an increasing number of PDOs the difference becomes negligeable. The advantage of the synchronous transmission is the spreading of the user data whereas the asynchronous transmission only occupies the bus when data has actually changed.

Number of SDOs at runtime

An SDO cycle consists of a client SDO and the reply, the server SDO. Both SDOs are normally 8 bytes long which means an SDO cycle at 250kbit/sec requires a total of 1.06 msec of bus time.

As far as the bus load is concerned, the „SDO cycle at runtime“ is what concerns us. This means the SDOs are in a phase in which at least one CANopen device can transmit, the so-called „operational“ condition. It is at this point that data transmission can jam up.

SDO cycles at run time are caused when the application reads or changes CANopen parameters from the devices (parameter upload/download). They are also caused when devices power down during bus operation and have to be re-configured.

However, SDO cycles do not block PDO traffic as they are, due to their higher indentifying numbers, only operational when PDOs are inactive. This does mean however, that too much PDO traffic can block out the SDO traffic. This is why it is advisable, as stated above, to aim for an average bus load of under 70%.

Sync Interval

The sync interval defines the time interval between the transmission of Sync telegrams.

The effect of the Sync on the bus load can be demonstrated with the following example:

Sync interval 50msec, Bit rate 250 kbit (Sycon default setting). The sync telegram (data length =0) has a run time of 0.22msec. It is sent 20 times per second, i.e. it uses 20x0.22=4.4 msec of the CANopen bandwidth. This is effectively a bus load of 0.44%

However, to view the Sync in its isolated form is unrealistic as it is the trigger for all synchronous PDOs so that each Sync is followed by a series of PDOs which generate the real bus load in this case.

Type of Error Protocol

CANopen has two methods for supervising devices: Node Guarding und Heartbeat.

Node Guarding

With Node Guarding, the controlling device (NMT Master) sends a request to all devices it is supervising. The device must reply to this request. The request and response telegrams for this supervision are CANopen telegrams with a single data byte. The interval between two request telegrams is called the Guard Time and is provided by the device manufacturer in the EDS. A standard default value is 200 msec.

Node guarding with this default value at 250kbit/sec bit rate would take 2x0.26=0.52 msec pro device. For a Canopen bus with 10 devices this would add 2.6% to the bus load.

Heartbeat

With the Heartbeat method, all the controlled devices (Heartbeat Producer), including the controlling device, send a cyclic (hence heartbeat) message to the supervising Heartbeat Consumer. Heartbeat messages are CAN telegrams with one data byte. The interval between heartbeat messages is defined by the "Heartbeat Producer Time“, a value provided by the device manufacturer in the EDS and usually set to a default of 200 msecs.

Using the same example as for node guarding above, for 10 devices and the controlling device, we require 11 heartbeat messages, i.e. 11x0.26 msecs. With a 200msec default, we have 5 cycles per second which results in 5x2.86=14.3 msecs or a 1.43% increase in bus load.

Conclusion: The bus load for larger CANopen configurations using node guarding is twice as much as when using the heartbeat method.

Quality of Transmission / Bus physics

Finally, the quality of transmission also has an effect on the bus load. Bad transmission can cause CAN to generate error frames in the acknowledgement slot of the telegram which leads to a repetition of the telegram transmission and increases the bus load.

To avoid this you should:

- Only use standard CANopen cable
- Never exceed allowed cable or branch lengths
- Ensure the bus is terminated with the proper resistors (120 Ω)
- Earth the cable screening properly (large surface area)
- Never exceed 64 devices per bus segment

The double assignment of COB IDs can also have a negative effect on transmission quality and bus load. If two or more devices are transmitting with the same COB ID, collisions cannot be avoided and error frames are generated.

Recommendations for Network Parameter Optimisation

A few tips to begin with:

Baudrate

As long as cable lengths and devices allow it, start at a bit rate of 250 kBit/s or adjust as required. Configure the devices and optimise the bus parameters as listed below. Once this is completed you can increase the bit rate if more bandwidth is required (Never exceed the maximum transmission speed for the lengths of cable involved)

Sync Intervall

The Sync telegram itself has a negligeable effect on the bus load (see above), however the PDOs it generates as a response do.

A default value of 100msec is often used the Sync interval. If you want to update your process data more often, reduce this value. On the other hand if you want to reduce the bus load, increase this value. As the bus load is indirectly proportional to the Sync interval, the value should not be too small. As a rule, it should not normally be less than a third of the PLC cycle time. This guarantees that the PLC has up to date data in every cycle.

Error Protocol

As it generates less bus load, use the heartbeat protocol where possible. Use node guarding only for devices with no heartbeat. By increasing the heartbeat producer time (default 200msec.) you can reduce, even if only minimally, the bus load.

With node guarding, increasing the Guard Time has a more positive effect on the bus load.

In both cases the rule is, the bigger the value, the lower the bus load. However the bigger the value the longer it takes to recognise a missing device.

Transmission Types

As the **RPDO**s only define when devices set received data as valid, their transmission types (TT) are irrelevant. Here we will only consider the transmission types of **TPDO**s.

The RPDOs still use bandwidth however. Normally the PLC generates the trigger for the TPDOs. For the Premium and Micro PLCs this is the SEND trigger. See the explanation in the section „CPP100,CPP110“ below.

As they define when a device is allowed to send its PDOs, the transmission types of the TPDOs have a considerable effect on the bus load.

With each TPDO you must decide whether to make it a synchronous(0 ... 240) or asynchronous(255) transmission type. **Do not use the Remote Request**. The EDS files usually provide default values for the PDOs. Use either these values or take them as guidelines.

Advantage of synchronous TT: Data transmission has an even spread.
Advantage of asynchronous TT: Data transmission immediately on data change

Some common rules for choosing the Transmission Types:

- **Digital Data:** TT255 (Send on change), Inhibit Time= 0 (switched off)), Event Timer= 100 ms. As digital data is not continuously changing, the inhibit time can be switched off. Advantage: Data changes are sent immediately. To reduce the bus load, increase the value for the event timer.
- **Analog Data:** TT255 (Send on change), Inhibit Time = 100 [= 10 ms], Event Timer= 100 ms. As analog values continually change, the transmission frequence must be slowed down using the Inhibit Time. To reduce the bus load increase the values for the Inhibit Time and the Event Timer. The application should indicate what values are acceptable and what are not.
- **Equidistant Data:** TT1, Sync Intervall = 100 ms. Use this method so that several PDOs can be sent equidistant. The Sync telegram serves as a Trigger. The sync interval can be reduced/increased as required. The application will show what value is acceptable. To decrease bus load, PDOs can be sent after each N th. Sync has been received, especially if data transmission is slow (e.g. due to temperature)

  Example: Consider the following scenario:

  5 PDOs for revolutions, 12 PDOs for fill levels, 20 PDOs for temperatures. Sync-Interval 25 ms. revolution-PDOs: TT=1, fill level-PDOs: TT=4, Temperature-PDOs: TT=40. This would mean that the revolutions are transmitted every 25 ms, the fill levels every 100 ms and the temperatures second.

  In total $5*40 + 12*10 + 20*1 = 340$ PDOs/sec plus 40 Syncs/sec.

  Assuming that each PDO contains 4 data bytes, with a bit rate of 250 kbit, we have a bus load of $(340*0.37 + 40*0.22)$ msec, i.e. 13.5%
  Note: Synchronous transmission guarantees the equidistance of the PDOs. There is no guarantee however that 2 devices transmit their PDOs with TT2 at the same time. Indeed, it is more likely that device A sends its PDO with every even Sync and device B sends its PDO with every odd Sync.

Network Parameters for Schneider Electric Products

The following tables list default values, recommendations and commentary on Schneider Electric products. The following abbreviations are used:

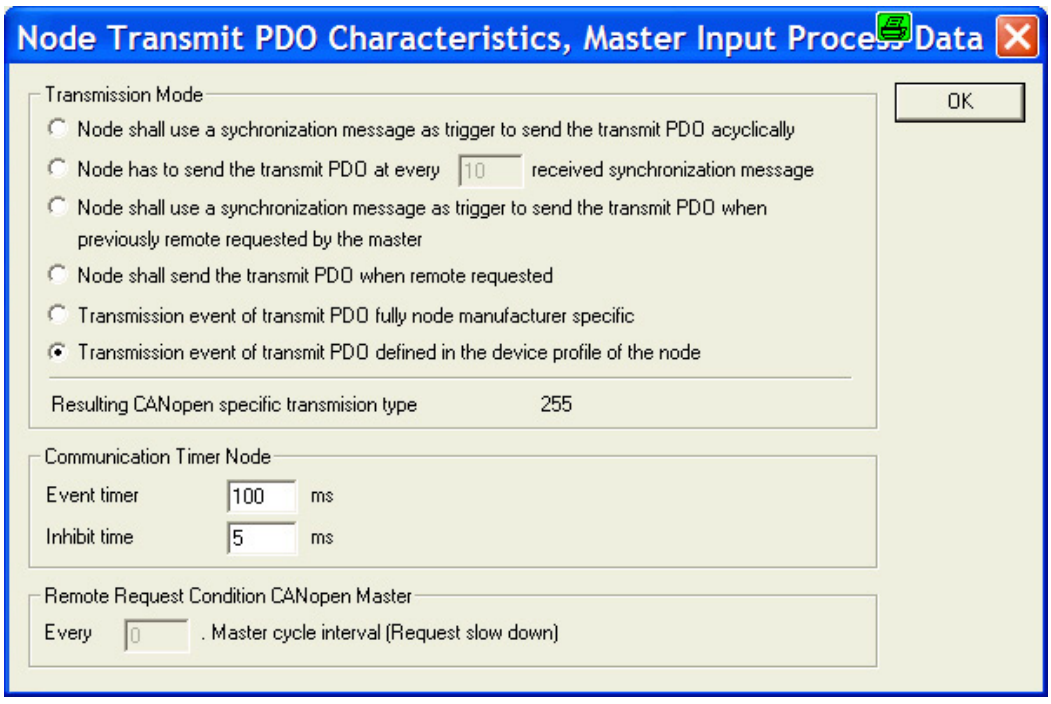
TT Transmission Type
IT Inhibit Time: min. time lapse between 2 PDOs
ET Event Timer: max. time lapse between 2 PDOs

Altivar 31

Warning: The time granularity for the IT value on the altivar is 10 times that normally given for standard CANopen devices (1ms instead 100 us). To offset this, the values for the configuration tools are reduced by a factor of 10.

PDO-No. / Type of Data	Default Value (EDS)	Comment
TPDO1	TT = 255, IT = 50, ET = 100	Due to the profile the TT value cannot be changed. If needed, to reduce the bus load, the value for Event Timer can be increased.
TPDO6	TT = 255, IT = 50, ET = 100	With this set up the altivar would want to send the PDO on every change of the actual value. However it must wait at least 5 msec between PDO transmission (see warning above). If the drive is stopped (i.e. actual value remains unchanged), the transmission period is 100 ms. The bus load can be reduced by increasing the values for IT and ET. This results in less updates of the actual value. If the actual data needs to be equidistant you must choose a synchronous TT.

Configurationd ialog for the TT for TPDO6



Altivar 58

PDO-No. / Type of Data	Default Value (EDS)	Comment
TPDO1	TT = 255, no IT or ET	The Transmission Type cannot be modified. Devices with firmware version V1.0IE02 transmit in 10 ms intervals, firmware version V1.0IE03 has a minimum transmission interval of 50 ms and a maximmum of 1 sec (all fixed). The Altivar 58 offers no possibility to adjust the transmission frequency of the PDOs. The only advice here is that when the altivar is stopped it should be taken out of „run“ mode to avoid the transmission of the actual value even though it has not changed.

Altivar 71

PDO-No. / Type of Data	Default Value (EDS)	Comment
TPDO1	TT = 255, IT = 300, ET = 1000	ATV71 functions according to the new 402 profile.A PDO is only sent when the status word changes or the event time expires. In this case the inhibit time can be switched off so that a PDO is sent every time the status word changes instead of being sent every second.

Advantys STB, OTB

PDO-No. / Type of Data	Default Value (EDS)	Comment
TPDO1	TT 255, IT = 0	As PDO1 is mapped to digital data, it only needs to be changed if the application requires a synschronous transmission.
TPDO2	TT 255, IT = 0	As PDO2 is mapped to analog data, it requires a reasonable inhibit time (application dependant). If the application needs a synchronous tramsmission, change the TT.
TPDO3, ...		Depending on the data type, see TPDO1 or TPDO2 above.

Advantys FTB, FTM

PDO-No. / Type of Data	Default Value (EDS)	Comment
TPDO1	TT = 255, IT = 0, ET = 0	Define a reasonable value for the Event-Timer , e.g. 3 secs. As this only concerns digital data, only change the TT if the application requires it.

CPP100, CPP110

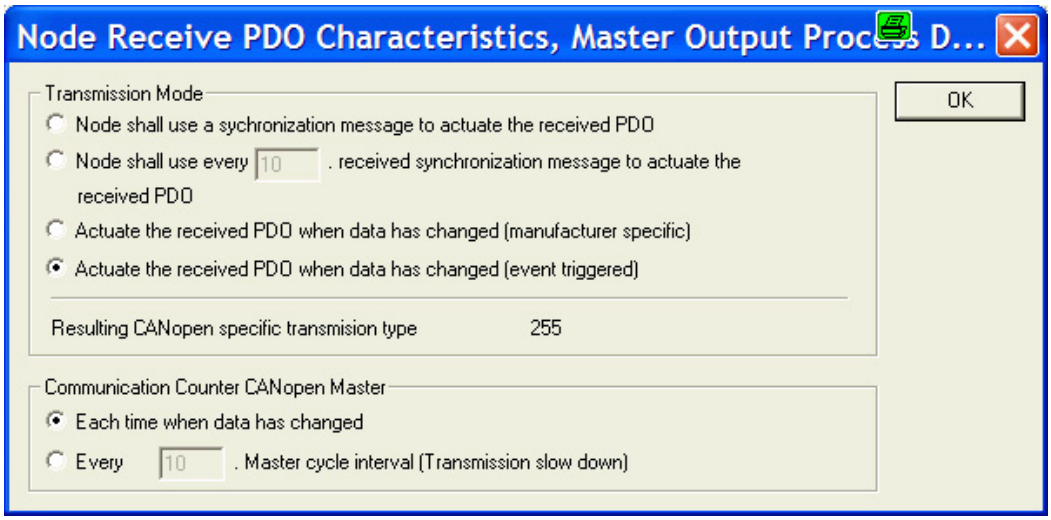
On the CPP100/CPP110, the send trigger for the RPDOs is already defined (Triggering Mode). This defines when the CPP110 sends the appropriate PDO. Choices are:

- Send PDO when process data changes
- Send PDO cyclically after each N th. Master cycle interval

Triggering Mode	Comment
Send on change of process data	The CPP1x0 only sends the PDO when the data changes. The interval between 2 PDOs is always greater than the cycle time of the PLC program.
Send cyclically	Der CPP1x0 sends the PDO cyclically, after every N th. firmware cycle. The firmware cycle time is not fixed and independent of the CANopen configuration. Here it is advisable to use an analyzer to tune the send interval.

Configuration dialog for the trigger modes

Triggering Mode „Send on change of process data“



Contact

Author	Phone	E-mail
Schneider Electric GmbH Customer & Market System & Architecture Architecture Definition Support	+49 6182 81 2555	cm.systems@de.schneider-electric.com