BECKHOFF	Version	V0.03
	Date	9.03.2017
	Author	Rawlyk, Gehle, Achterberg

XTS Application Guide
(EtherCAT & Real Time configuration)

1 General Overview

It is suggested to use one core just for the fast 250µs XTS task (100% isolated). The interface between XTS TcSoftDrive and TcNc is well designed for use on separate cores. The computing requirements of the XTS system are dependent upon 2 factors. The time required to update the EtherCAT network(s) and the amount of execution time required to calculate the mover positions and mover control algorithms. The length of the track (per XTS infeed module) and how many infeed's are used determines how long the EtherCAT update will be, the number of movers will affect how long the calculation takes. Both the EtherCAT update and the XTS calculation must be completed within the 250us XTS cycle time.

In the case of the XTS System it is recommended to use a Beckhoff IPC with i7 quad core CPU (4th Generation) 2.4GHz and a 64Bit OS. Before using other Beckhoff IPC's please contact the Beckoff XTS support first. It is not recommended to run HMI applications on the XTS IPC and program on another system with remote access.

2 Considerations

- The XTS task must run with 250µs, do not change the cycle time of this task.
- Only tasks on the same core can use the same CU2508. Tasks on different cores must are not allowed to access the same CU2508! For example it is highly recommended against using a free port of the port multiplier CU2508, where XTS units are connected, for other IO devices.

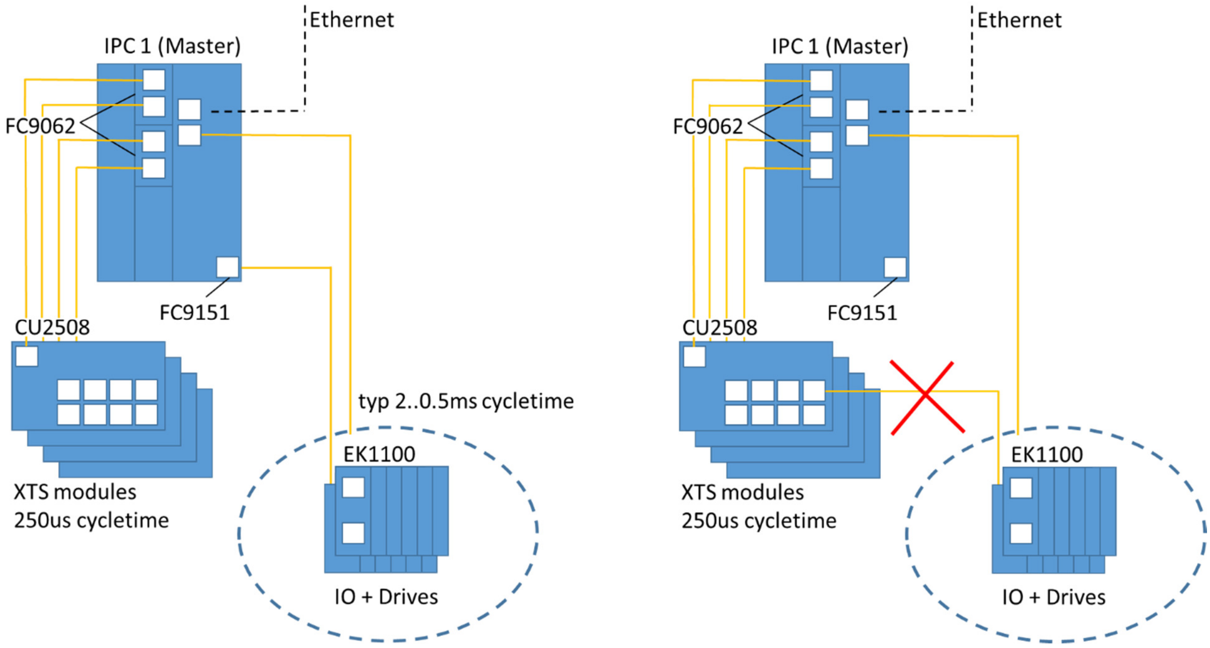


Figure 2-1: Suggested XTS IO connection

3 RealTime and Multicore Settings within TwinCAT

This picture shows the MultiCore settings for a XTS application. In the case of using a XTS System it is recommended to use 1 core just for windows and three cores as isolated real time cores. Through the hard separation of windows and real time functions, the best real time behavior could be achieved and windows/ graphic operations have least impact. But, anyhow it is not recommended to run an extended visualization on the XTS runtime IPC. Furthermore it is important to spread the real time calculation even over the remaining real time cores.

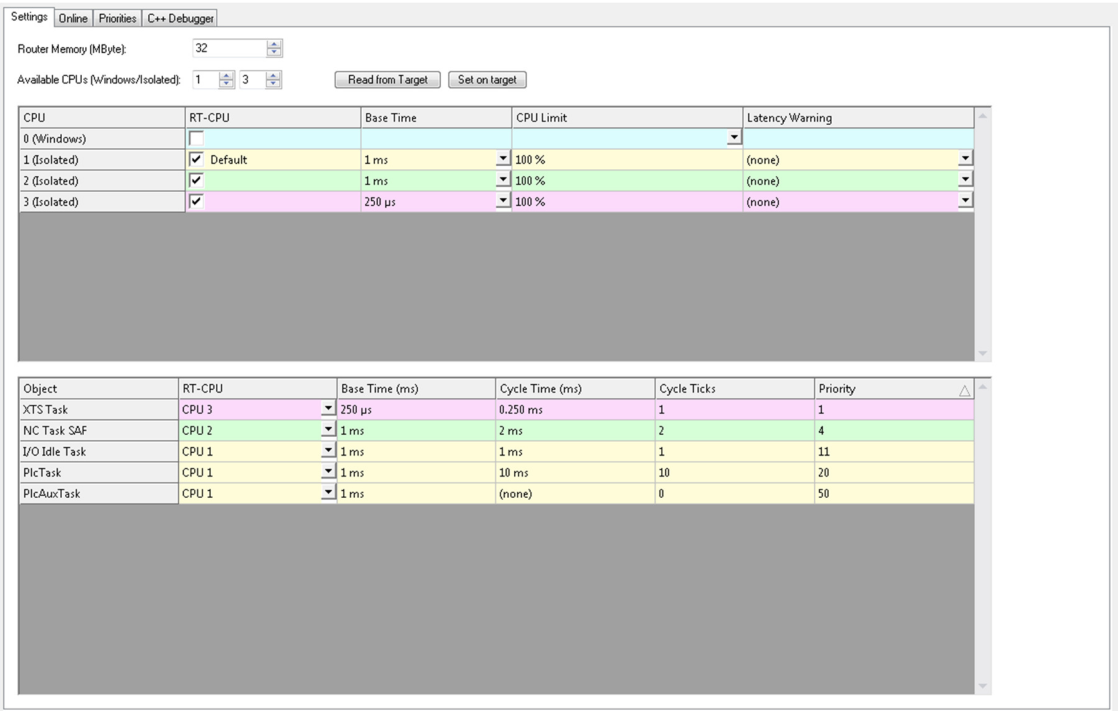
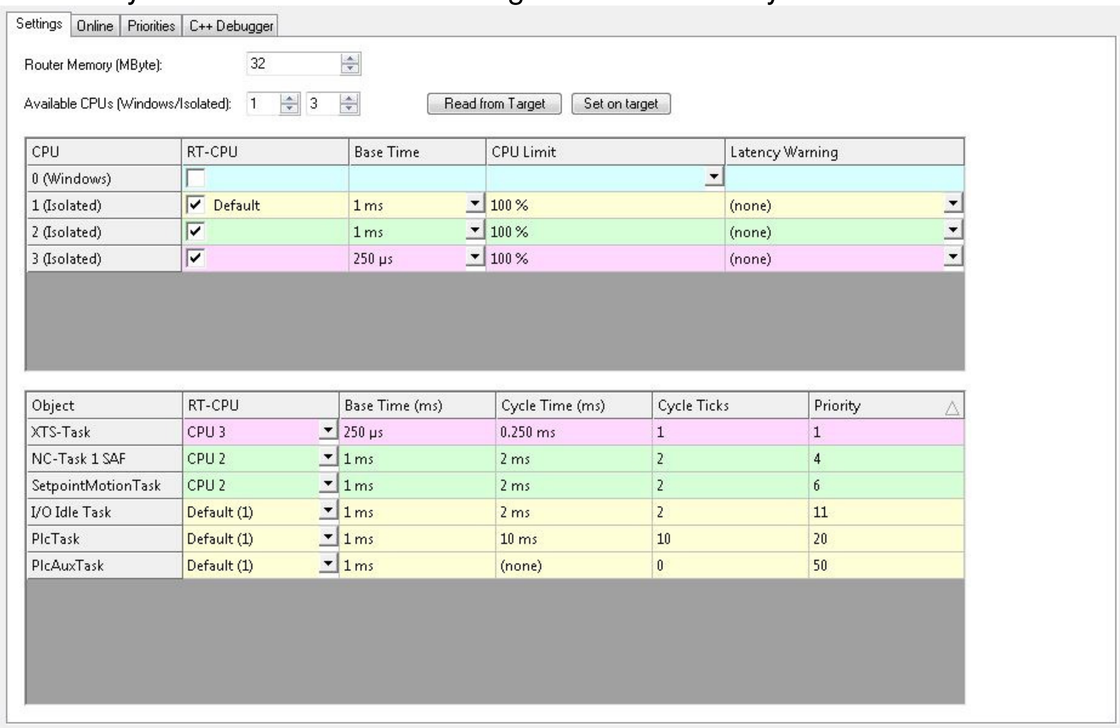


Figure 3-1: Real-time settings without external set-point generation

The picture above shows the recommended RT-Settings for an application with one XTS-System and the C6930-0050. For reaching the best performance it is possible to use one separate core for each task. The XTS is running on the third core with a base tick time of 0.25ms, 1 cycle tick and a resulting cycle time of 0.25ms. The NC-task is using the second core with the standard configured cycle time of 2ms. If it is not necessary to use the NC with a cycle time below 1ms, the base tick time should stay at 1ms to reduce the CPU load. If no cyclic setpoint values for the NC are calculated in a PLC-Task the Plc, PlcAux and I/O Idle Task should be set on the first core with the required cycle time.

The next picture shows the task settings, if cyclic set point values are requested in the NC, for example for a self-calculated motion profile. In this case it is necessary that the NC and calculating PLC-Task runs synchronous.



CPU	RT-CPU	Base Time	CPU Limit	Latency Warning
0 (Windows)	☐			
1 (Isolated)	☑ Default	1 ms	100 %	(none)
2 (Isolated)	☑	1 ms	100 %	(none)
3 (Isolated)	☑	250 µs	100 %	(none)

Object	RT-CPU	Base Time (ms)	Cycle Time (ms)	Cycle Ticks	Priority
XTS-Task	CPU 3	250 µs	0.250 ms	1	1
NC-Task 1 SAF	CPU 2	1 ms	2 ms	2	4
SetpointMotionTask	CPU 2	1 ms	2 ms	2	6
I/O Idle Task	Default (1)	1 ms	2 ms	2	11
PlcTask	Default (1)	1 ms	10 ms	10	20
PlcAuxTask	Default (1)	1 ms	(none)	0	50

Figure 3-2: Real-time settings with external set-point generation

In this case it is possible to run the calculation in a separate, faster PLC task on the same core as the NC Task. With these settings it is ensured, that every NC-cycle new calculated set point values are available.
If all RT-CPUs are isolated cores and there is still some jitter seen in the online view on the task timing one improvement possibility could be to set the following key in the Registry. Open regedit.exe and create under:
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Beckkhoff\TwinCAT3\System
the DWORD variable RtProfile and set it to 2. For a 32 Bit OS without Wow6432Node.

4 IO configuration

If several XTS infeed modules are used (to build a XTS path) requiring multiple EtherCAT networks it is necessary to connect them to a CU2508. If more than 4 XTS infeed modules are used, it is necessary to use a separate CU2508 because of the transfer time of the EtherCAT Data it is possible to use only 4 XTS infeed modules per CU2508.

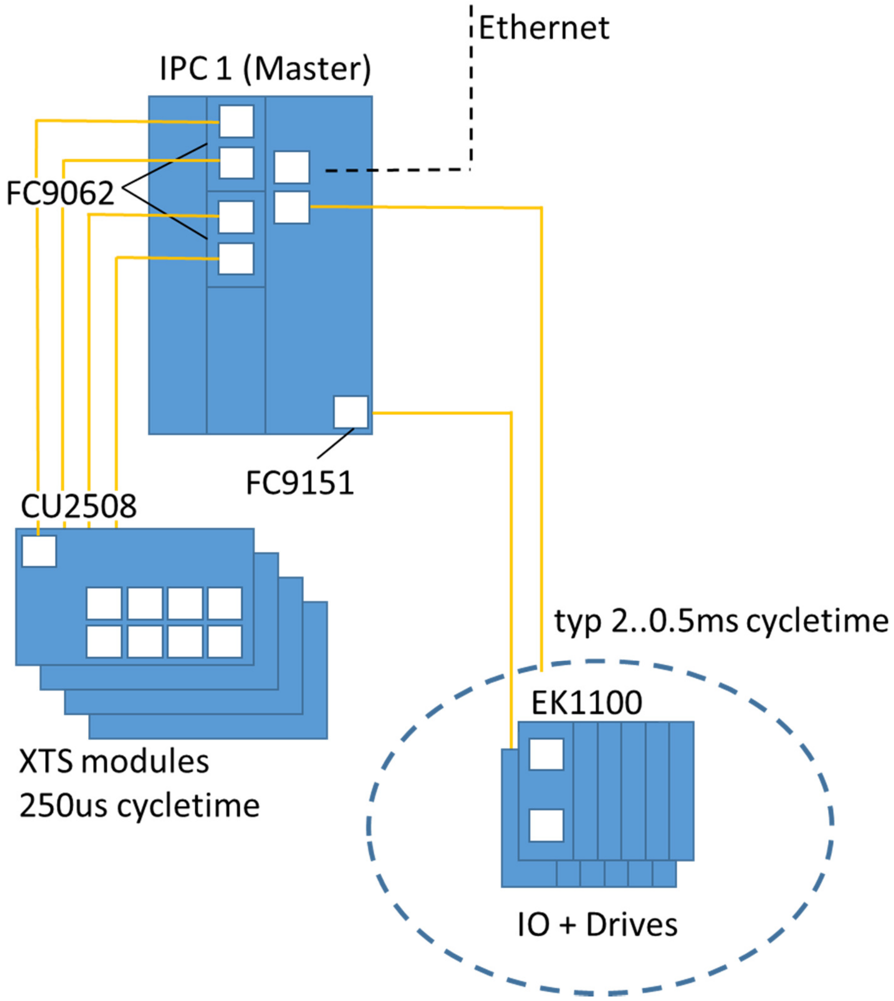


Figure 4-1: XTS CU2508 configuration

The picture above shows a sample configuration with a C6930-0050 with two FC9062 and a FC9151 (or FC9071) option card for the extension to 7 Ethernet adapters. The first on-board Ethernet adapter should only be used for functions with a lower required performance, like a company Ethernet Network or remote access. The additional Ethernet Ports on the FC9062 cards could be used for up to 4 CU2508 from which every CU could be connected to maximum 4 XTS infeed

Modules. The remaining two Ethernet adapters could be used for the other IO or drives which belongs to the application.

To configure the shown set-up from figure 4-1 “XTS CU2508 configuration” a few things must be done, to get a stable running EtherCAT and XTS System.
The port multipliers must be synced among each other and configured as the reference clock, the DC-Master reference clock has to be a Hardware from an IO-task which has the lowest cycle time and the output shift times must be adjusted.

4.1 Synchronization of the CU2508

For the Synchronization of each CU2508 it is necessary to connect the master CU2508 (the first one) to each other CU2508 with an Ethernet connection at the free Ethernet ports 5-8. As shown in the following screenshots the ports at the master CU has to be configured as “Sync Master” and the connected port at the slave has to be configured as “Sync Slave” at the advanced settings of the “RT-Ethernet Adapter” ports.

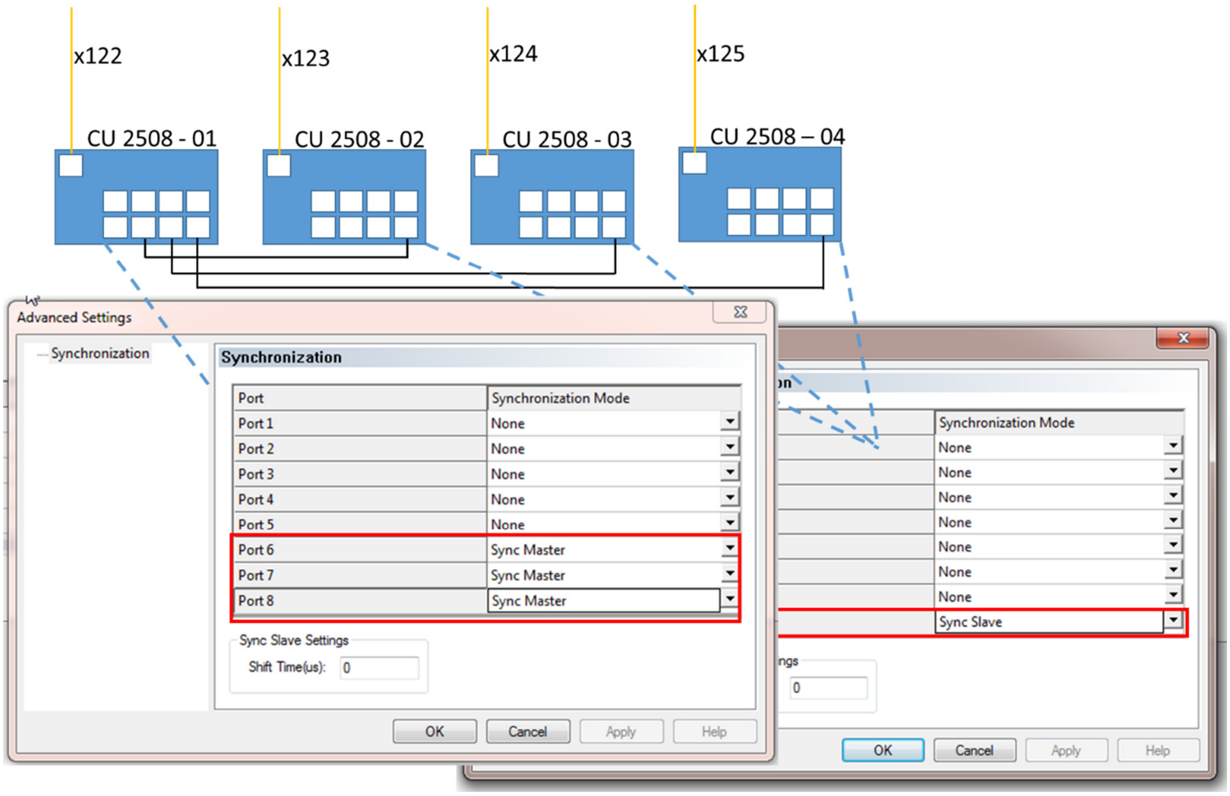


Figure 4.1-1: Synchronization of the CU2508 network

4.2 Reference Clock Configuration

Set the port multiplier as potential reference clock (see the following picture) for all of the XTS lines.

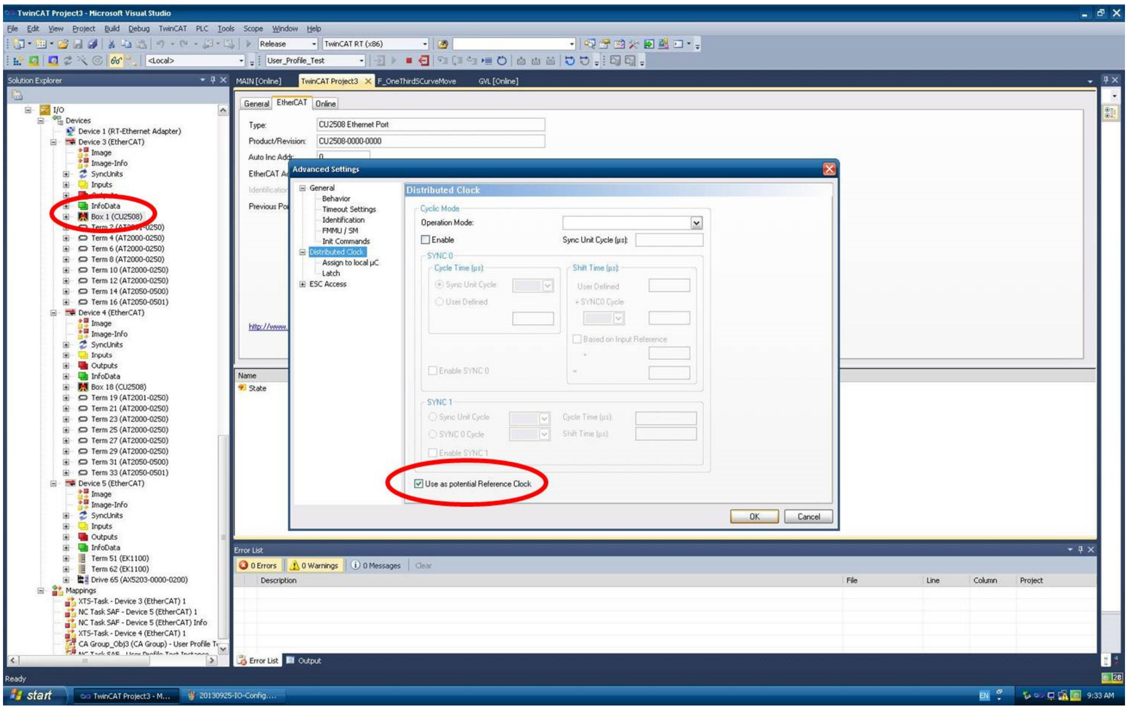


Figure 4.2-1: Reference clock configuration

4.3 Set the first XTS EtherCAT Master to the DC-Reference Clock

In a default configuration the TwinCAT System Manager sets the first DC-able Device as independent DC Master Clock, which will be the master Clock for all DC-able devices in the EtherCAT network. If the DC-master clock device is triggered with a slower cycle time than the XTS cycle time of 0.25ms the DC-clock is not updated every XTS-cycle in all DC slaves and could jitter much more, as it is updated in every XTS cycle.
To set the first EtherCAT XTS Master as reference clock it is necessary to uncheck the “Automatic DC mode Selection” and choose the option “Independent DC Time (Master Mode)”. In the same way the last DC-master must be told to use the DC-time of the XTS-EtherCAT Master, because there could be only one DC Master clock in one EtherCAT configuration.

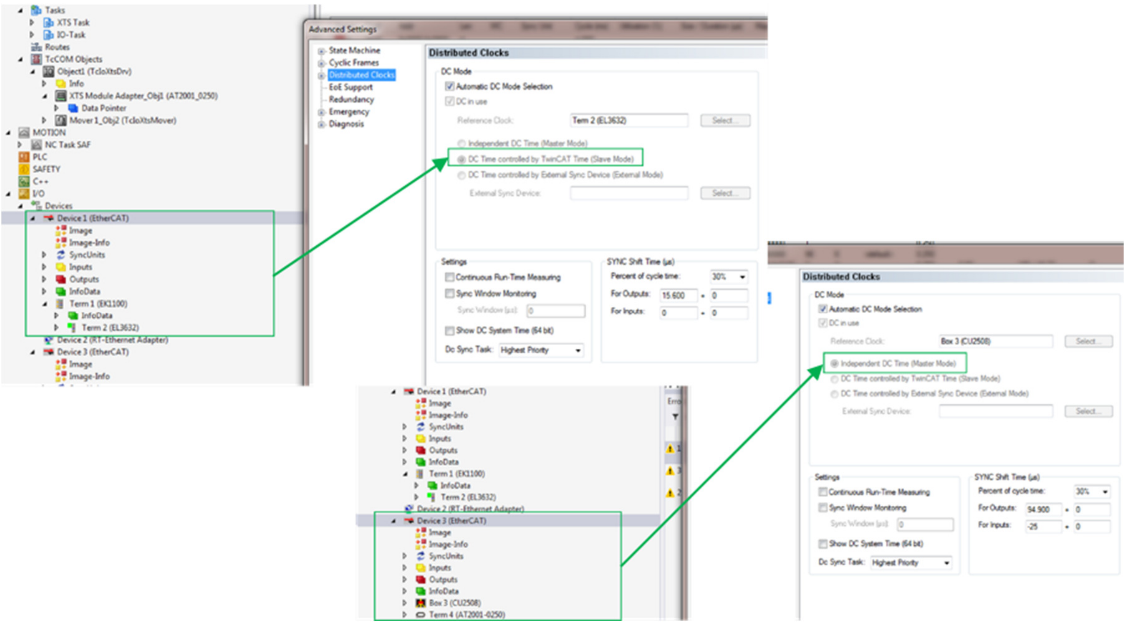


Figure 4.3-1: Set XTS EtherCAT to the Master DC-reference clock

4.4 Output Shift time adjustment

It is necessary to synchronize the Sync output shift time to the same and latest time.

Example:

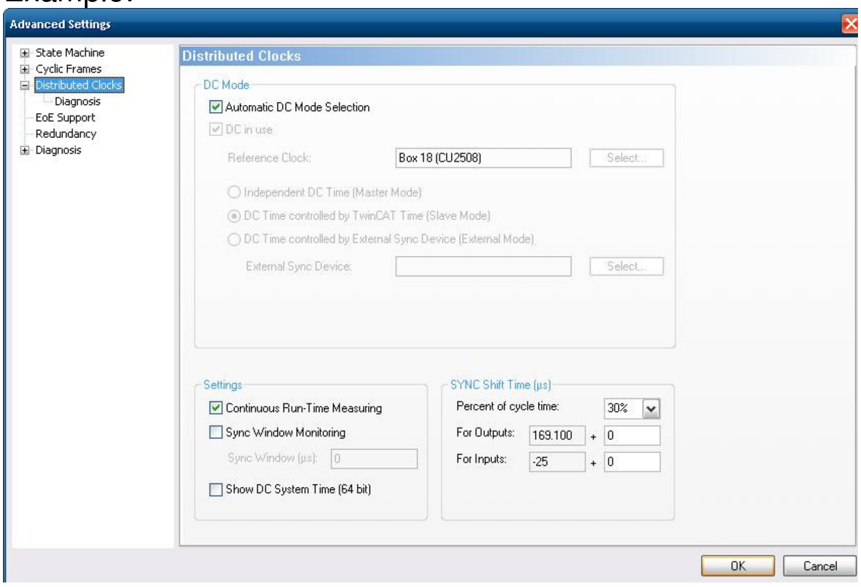


Figure 4.4-1: XTS EtherCAT Master with the latest Output time

DC settings in EtherCAT Device 4. The TwinCAT System Manger pre-calculated an output shift time of 169.1µs.

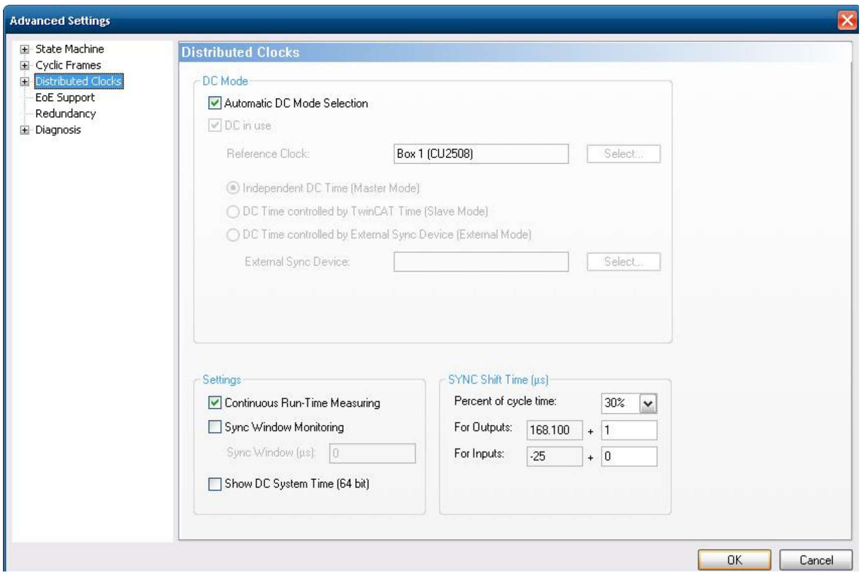


Figure 4.4-2: XTS EtherCAT Master Adjustment to the latest Output time

DC settings in EtherCAT Device 3. The System Manger pre-calculated an output shift time of 168.1µs. To bring both devices on exactly the same time an additional output shift time of 1µs is set on Device3 thereby giving both devices an output shift time of 169.1us.

5 XTS Task Settings

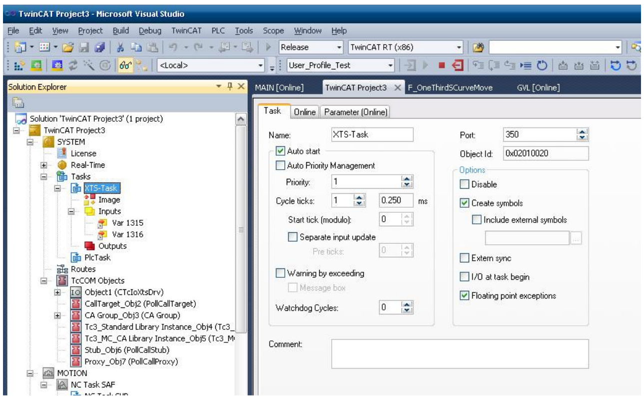
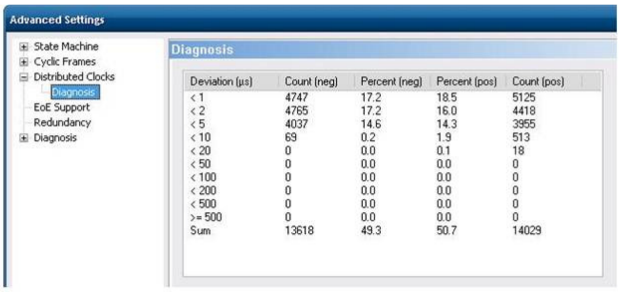


Figure 5-1: XTS Task settings

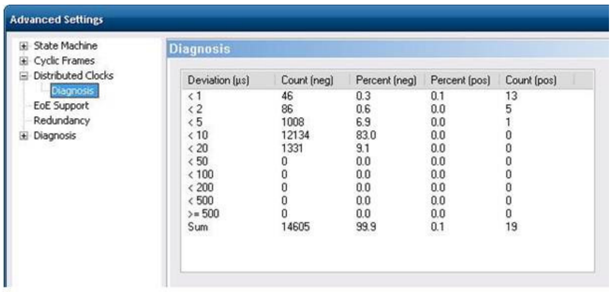
For the XTS Task port number, or if several XTS tasks and systems are running on one IPC, any number between 350 and 359 could be used.

6 EtherCAT Device Timing



Deviation (µs)	Count (neg)	Percent (neg)	Percent (pos)	Count (pos)
< 1	4747	17.2	18.5	5125
< 2	4765	17.2	16.0	4418
< 5	4037	14.6	14.3	3955
< 10	69	0.2	1.9	513
< 20	0	0.0	0.1	18
< 50	0	0.0	0.0	0
< 100	0	0.0	0.0	0
< 200	0	0.0	0.0	0
< 500	0	0.0	0.0	0
>= 500	0	0.0	0.0	0
Sum	13618	49.3	50.7	14029

Device 3



Deviation (µs)	Count (neg)	Percent (neg)	Percent (pos)	Count (pos)
< 1	46	0.3	0.1	13
< 2	86	0.6	0.0	5
< 5	1008	6.9	0.0	1
< 10	12134	83.0	0.0	0
< 20	1331	9.1	0.0	0
< 50	0	0.0	0.0	0
< 100	0	0.0	0.0	0
< 200	0	0.0	0.0	0
< 500	0	0.0	0.0	0
>= 500	0	0.0	0.0	0
Sum	14605	99.9	0.1	19

Device 4

Figure 6-1: EtherCAT device timing

This picture shows the DC diagnostic of the two XTS EtherCAT Master. Device 3 is the first and device 4 the second one. Device 4 shows higher derivation numbers. This is not a higher jitter, but rather this comes from the additional time from task begin till the data for the second device is proceeded by the IPC and the CU2508 and the second XTS frame is send out by the CU2508.

7 EtherCAT timing with CU2508 and XTS task timing

While each network on a CU2508 is updated simultaneously the data transmission from the PC to the CU2508 and inside the IPC is done sequentially. Additional the physics in the Hardware takes also some time. This ends up in an overhead to what is seen in the System Manager and the EtherCAT master:

- 3µs per motor module hardware duration (sum for back and forth)
- 5µs for internal data transmission inside the CU2508
- Gigabit delay for communication between IPC and CU2508 (10% of the longest 100 Mbit time multiply with (n+1) number of infeed’s connected on the CU2508)

The total time required for the complete XTS EtherCAT transmission is the sum of the longest EtherCAT network update time plus the overhead required for the total EtherCAT communication and the CU2508.

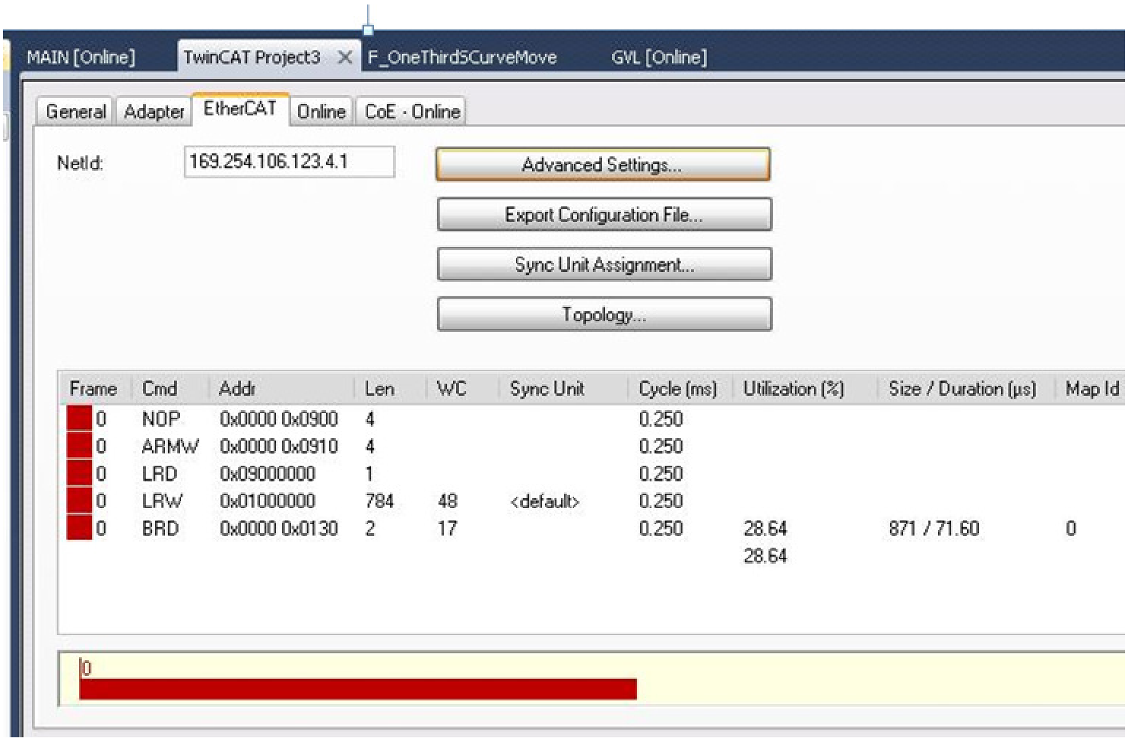


Figure 7-1: View on the EtherCAT timing for a 2m XTS system

Calculation example 1: 4m XTS calculated (IPC according chapter 1) with two infeed's connected to one CU2508 and 28 mover:

- 100 Mbit time 71.6 µs (visible in TwinCAT)
- 8 · 3 µs = 24 µs Hardware delay per infeed
- 5 µs internal delay for the CU2508
- (n+1) · 0.1 · 100 Mbit Time = 3 · 0.1 · 71.6µs = 21.5 µs for the Gigabit communication
  - ➔ 122.1 µs for communication

For the task timing, the IO data handling takes with the suggested IPC 3µs per meter XTS, 10µs task scheduler overhead and rough 2 µs per mover.

- 4 · 3 µs = 12 µs for the IO data handling
- 10 µs task overhead
- 28 mover · 2 µs = 56 µs for the mover calculation
  - ➔ 78 µs XTS task time

This ends up in this example and rough calculation to 200.1µs time for communication and task calculation and gives around 49.9 µs necessary reserve for jitter.

Calculation example 2: 6m XTS calculated (IPC according chapter 1) with four infeed's each 1.5m long connected to one CU2508 and 35 mover:

- 100 Mbit time 55.9 µs (visible in TwinCAT)
- 6 · 3 µs = 18 µs Hardware delay per infeed

- 5 µs internal delay for the CU2508
- (n+1) · 0.1 · 100 Mbit Time = 5 · 0.1 · 55.9µs = 28 µs for the Gigabit communication
  ➔ 107 µs for communication

For the task timing, the IO data handling takes with the suggested IPC 3µs per meter XTS, 10µs task scheduler overhead and rough 2 µs per mover.

- 6 · 3 µs = 18 µs for the IO data handling
- 10 µs task overhead
- 35 mover · 2 µs = 70 µs for the mover calculation
  ➔ 98 µs XTS task time

This ends up in this example and rough calculation to 205 µs time for communication and task calculation and gives around 45 µs necessary reserve for jitter.