

Master Development Kit 2

for Windows
Change Log

Version 2.4.29520.0
Date 2020-10-01

Table of Contents

1 MDK for Windows 2.4.29520.0, released 2020-10-01.....	3
2 MDK for Windows 2.4.25003.0, released 2020-05-25.....	6
3 MDK for Windows 2.3.43629.0, released 2020-04-03.....	7

1 MDK for Windows 2.4.29520.0, released 2020-10-01

1.1 Features

1.1.1 EoE support for Windows based OS

From now on it is possible to communicate between a slave and the master which is running on Windows based host, using Ethernet over EtherCAT (EoE) protocol.

The name of EoE adapter is set in the INI parameter **eo_adapter** (default value is "opentaprun").

1.1.2 Timed send emulation with channel redundancy

A special forwarded segment and a separate queue for forwarded segments have been added to support timed send with channel redundancy.

1.1.3 Custom logic for reading explicit identification value

From now on a custom logic is used to read explicit identification value from the device.

It reduces a Master start up time and the time of detecting topology changes.

1.1.4 Operations retry at frame losing

Added a mechanism to repeat the operation (command send, EEPROM read, memory read etc.) when frame is lost. To set number of tries, a new parameter **scheduler_retries** was added to INI file, default value is 3.

1.1.5 Default transferring timeout calculation

Implemented a mechanism of default transferring timeout (**t**) calculation:

$$t = \text{MAX}(\text{MasterCycle} * 2, t_{\min})$$

where $t_{\min} = 10 \text{ ms}$

Therefore, $t = 2 * \text{MasterCycle}$ but not less than 10ms.

Such mechanism allows to implement fast system reaction and minimize frame lost due to delay in the system.

1.1.6 Default task queue calculation

Implemented a mechanism of setting a default size of task queue (taskqueue.size parameter) depending on memory pool size (MemoryPoolSize parameter):

Memory pool size, Mb	Task queue
... < 64	4096
64 ≤ ... ≤ 256	8192
256 ≤ ...	16384

1.1.7 Up to 256 simultaneous slaves memory requests

From now on the number of simultaneous requests for reading slave's memory is increased up to 256.

1.1.8 Printing out INI parameters

A possibility to print all Master initialization parameters has been added. INI parameter **Verbose** was added. When it is enabled (Verbose=1), all parameters specified in INI file are printed out at starting Master.

1.1.9 New error code

ECAT_E_FRAME_CMD_SIZE_LIMIT is new error code means "Process frame error: the command size is too big to be placed into the frame".

1.1.10 INI file extension

New initialisation parameters have been added to configure Master using its INI file.

ss.eeprom.maxrequests - maximum number of simultaneously serviced EEPROM requests

transition.maxcommandsperslave - maximum number of commands to be sent to slave at a time.

scheduler.oncepercycle - schedule one driver send request per cycle.

scheduler.queueing.timeout.cycles - number of master cycles for queuing timeout.

Updated default values of the following parameters:

```
DrvTxAffinity=1,  
DrvRxAffinity=1,  
threads.count=3,  
threads.0.executors=rt,  
threads.0.priority=15,  
threads.0.affinity=2,  
threads.1.executors=work,  
threads.1.priority=2,  
threads.2.executors=sync,sys,  
threads.2.priority=15,  
scheduler.oncepercycle=1,  
scheduler.allatonce=1,  
ecatrouter.fwd.postponed=1,  
topology.maxconcurrentcmds=100,  
transition.maxcommandsperslave=100,  
transition.slavetransitionattempts=0.
```

1.1.11 New variable to share data folder

Now to set shared data folder, the environment variable ECAT_MASTER_MAKE_DATA_FOLDER_SHARE is used instead of CFG_SHAREDATADIR.

1.1.12 Slaves configuration for Autoconfigurator in slavelib sub-folder

From now on slaves configuration (uESI files generated in KPA EtherCAT Studio 2) should be in the **slavelib** sub-folder of either default folder or custom folder set in KPA_AC_ESI_DIR_NAME, depending what is used.

1.1.13 New API call to get RPC properties

New API call EcatRPCGetCapabilities has been added. It allows to get RPC API version and transport buffer size.

```
MKPA_RPCCLIENT_EXPORT ECAT_RESULT EcatRPCGetCapabilities(  
    IN ECAT_WORD connectionID,  
    OUT ECAT_DWORD * pdwApiVersion,  
    OUT ECAT_DWORD * pdwCapabilities,  
    OUT ECAT_DWORD * pdwTransportBufferSize);
```

where pdwApiVersion - version of RPC API
pdwCapabilities - reserved for future use
pdwTransportBufferSize - maximum size of transport buffer (can be set in INI parameter)

1.1.14 Asynchronous mailbox call EcatRPCReadMemoryByPosIndex

From now on EcatRPCReadMemoryByPosIndex() is called asynchronously.

1.1.15 Performance improvement

System performance has been improved by implementing a static update of Process Image region objects.

It reduces CPU usage at switching slave state therefore make the switching faster.

1.1.16 Mailbox improvement

Improved work with Mailbox protocols:

- All mailbox operations are stopped at slave state changing.
- Mailbox request execution is stopped if slave is offline.
- CoE module re-factoring.

1.1.17 EEPROM operations improvement

During EEPROM operations slaves are accessed sequentially, one by one.

1.1.18 KPA Framework improvement

Dynamic allocation and re-allocation of items in EcatSyncQueueLoopData have been added.

1.1.19 Fault notifications improvement

Separate error codes have been added to handle memory pool fails during Master initialization and to notify about certain functionality unsupported by KPA network driver

1.1.20 Python application for RPC client

New Python application to work with RPC client.

1.1.21 Review of Windows PCAP driver usage

Improved safety, synchronization and performance of PCAP driver usage.

1.2 Fixed bugs

- Mailbox counter is reset on any Slave state transition.
- Exception happens on the attempt to close a network adapter while working in timed send emulation mode.
- Repeated FoE Download operation fails with error after an interrupted attempt.
- Multimaster functionality is not available when timed send emulation is enabled.
- ECAT_E_FRAME_SEND_TIMEOUT happens on big configuration in ECAT send scheduler
- Master crashes after requesting state if Distributed Clock is enabled.
- Cycle time is up to one second while transferring from Pre-Operational to Safe-Operational states.
- Master default folder is not available in the network.
- Sending of multiple frames takes a lot of time.
- Getting of adapters list causes master cycle delay.
- Task queue overflow happens at scanning big bus configuration.
- Resource Manager bug fix.
- Command size limit is not checked while parsing ENI file and building a frame.
- Wrong order of master modules de-initialization.
- Fast reaction on slave requested state change, auto-recovery cycle should be aborted when a new state has been required
- Maximum frame size: incorrect procedures of its calculation and usage for real-time context.
- Attach to Master configured with non-existing network card is successful.
- Last slave on the bus is missing or not recognized.
- Memory leak happens at reading slave memory.

2 MDK for Windows 2.4.25003.0, released 2020-05-25

2.1 Features

2.1.1 Auto configurator

A possibility to configure the bus on the fly has been added. The Online Configuration module (Auto Configurator) allows the user application to create a bus configuration by applying slave's configuration provided by its vendor (EtherCAT Slave Information - ESI) or an extended slave's configuration generated by KPA EtherCAT Studio (user's ESI - uESI).

For details, refer to Master API Help (Getting Started/ Autoconfigurator).

2.1.2 MIP Samples installation improvement

MIP samples were moved to the Program Data folder and from now on you can install samples for different versions of MS Visual Studio which installed on your PC. Also, it is possible to rebuild installed samples for a new version of MS Visual Studio by using the Change option of MIP installer.

For details, refer to MDK Readme (MIP installation).

2.1.3 New samples

New samples have been added:

26_MultiMaster - demonstrates how to work with several Master instances in the same process.

40_SimplePLC - demonstrates the usage of MIP framework from the user application.

2.2 Fixed bugs

- API Help item in the Start menu is displayed with a wrong name.
- Memory leak on Master detach with disabled the Resource Manager service.
- Framework initialization processes cycles (Init-OP cycle) if final initialization target is changed from "Ready" to "Request State"
- Slaves follow master state if transition.disablesslavefollowstate=1.

3 MDK for Windows 2.3.43629.0, released 2020-04-03

Initial release of KPA EtherCAT Master Development Kit 2.3 for Windows based on version 1.6.

It includes new features:

- Asynchronous control
- Simple Tasks execution
- Multiple EtherCAT Cyclic Tasks
- Modular design
- PI-driven control
- Events based model
- EtherCAT network driver: zero copy

koenig-pa GmbH
Im Talesgrund 9a
91207 Lauf a.d. Pegnitz
Germany
www.koenig-pa.de

Contact
email: sales@koenig-pa.de
tel.: +49 9128 725 614 (Feucht)
tel.: +49 9128 725 631 (Feucht)
fax.: +49 9128 725 407 (Feucht)

koenig-pa GmbH Quality Management System
certified according to DIN EN ISO 9001:2015.
Initial certification in 2008.
Copyright © koenig-pa GmbH, Germany
All rights reserved
EtherCAT® is registered trademark and patented technology,
licensed by Beckhoff Automation GmbH, Germany

