



Change log

KPA Master Development Kit 2.4.50419.0, 2022-02-17

Contents

1. MDK version 2.4.50419.0, 2022-02-17	1
1.1. INtime	1
2. MDK version 2.4.50416.0, released 2022-02-14	2
2.1. Common	2
3. MDK version 2.4.50411.0, released 2022-01-28	3
3.1. Common	3
3.2. Xenomai	3
3.3. RTX64	3
4. MDK version 2.4.48714.0, 2021-12-02	4
4.1. Common	4
5. MDK version 2.4.48713.0, 2021-11-22	5
5.1. Common	5
5.2. FreeRTOS	5
6. MDK version 2.4.47806.0, 2021-09-24	7
6.1. Linux	7
7. MDK version 2.4.46804.0, 2021-08-17	8
7.1. Common	8
7.2. Linux	8
7.3. Xenomai	8
7.4. RTX	9
7.5. INtime	9
8. MDK version 2.4.45419.0, 2021-09-24	10
8.1. Common	10
9. MDK version 2.4.45405.0, 2021-06-25	11
9.1. Common	11
10. MDK version 2.4.44108.0, 2021-06-14	12
10.1. Common	12
10.2. RTX64	12
10.3. Rt-Preempt	12
10.4. Windows	12
10.5. Xenomai	13
11. MDK for Rt-Preempt 2.4.44103.0, 2021-05-28	14
11.1. Features	14
11.2. Fixed bugs	15
12. MDK for RTX64 2.4.44101.0, 2021-05-14	16
12.1. Features	16
12.2. Fixed bugs	17
13. MDK for Rt-Preempt 2.4.29549.0, 2020-12-14	18
13.1. Features	18
13.2. Fixed bugs	18
14. MDK for RTX64 2.4.37800.0, 2020-11-27	19
14.1. Features	19
14.2. Fixed bugs	22
15. MDK version 2.4.29528.0, 2020-10-29	23
15.1. Common	23

16. MDK version 2.4.29520.0, 2020-09-25	24
16.1. Common	24
16.2. Windows	27
17. MDK for INtime 2.4.26504.0, 2020-06-18	28
17.1. Features	28
17.2. Fixed bugs	28
18. MDK for Windows 2.4.25003.0, 2020-05-25	29
18.1. Features	29
18.2. Fixed bugs	29
19. MDK version 2.3.43629.0, 2020-04-20	30
19.1. Rt-Preempt	30
20. MDK version 2.4.15804.0, 2019-12-27	31
21. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.41907.0, 2019-10-18	32
21.1. Features	32
21.2. Fixed bugs	32
22. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.39209.0, 2019-07-31	33
22.1. Features	33
22.2. Fixed bugs	33
23. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.38100.0, 2019-06-28	34
23.1. Features	34
24. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.19400.0, 2018-12-17	35
25. MRT for RTX64 2.2.10600.0, 2018-01-31	36
26. Imprint and disclaimer	37
26.1. Trademarks	37
26.2. Disclaimer	37
26.3. Quality Management	37
26.4. Contacts	37
26.5. Mailing address	37
26.6. Copyright	37

1. MDK version 2.4.50419.0, 2022-02-17

1.1. INtime

1.1.1. Fixed bugs

- Minor bugs fix.

2. MDK version 2.4.50416.0, released 2022-02-14

2.1. Common

2.1.1. Features

2.1.1.1. A new call `EcatRPCStartClientWithTimeout()` has been added to `rpcclient` library to redefine communication timeout at rpc client startup.

2.1.1.2. XML parser improvement

From now on, in addition to the values 1/0, Master can recognize the True/False values of boolean variables when parsing an XML file.

3. MDK version 2.4.50411.0, released 2022-01-28

3.1. Common

3.1.1. Features

new copyrights in 1.6

3.1.1.1. New EcatRPCGetErrorMessage RPC call

Added a new EcatRPCGetErrorMessage RPC call to get error message by its code.

3.1.1.2. 07_ProcessImageDeliveryMultipleCycles sample

A new sample 07_ProcessImageDeliveryMultipleCycles has been added to demonstrate how to handle multiple tasks by multiple PI clients.

3.1.1.3. Sample of Synchronization mode 1 and 2

A new sample 41_MasterStartupSyncMode1and2 has been added to demonstrate Master 2 work in Synchronization 1 and 2 modes. It inherits behavior of the similar sample in Master 1.6.

3.1.2. Fixed bugs

- Wrong timing while working with several cyclic tasks.
- Incorrect definitions of the modules ECAT_MASTER_AC_ID, ECAT_PT_ID, ECAT_TASKQUEUE_ID.
- Insufficient description of Configuration Pool in API Help.

3.2. Xenomai

3.2.1. Fixed bugs

- Master chashes after running in Operational state for a while.

3.3. RTX64

3.3.1. Fixed bugs

- Exception while running 23_AC_MasterRestart sample.

4. MDK version 2.4.48714.0, 2021-12-02

4.1. Common

4.1.1. Features

4.1.1.1. Master Diagnostics sample

A new sample application 29_Statistics has been added to demonstrate how to handle diagnostic registers and Master internal statistics.

5. MDK version 2.4.48713.0, 2021-11-22

5.1. Common

5.1.1. Features

5.1.1.1. Sample of missed PI updates detection

A new sample application 28_MissedPIUpdateDetection has been added to show how to detect missed PI updates in a custom application.

5.1.1.2. Sample of Process Image mapping

A new sample application 27_FullPIClient has been added to show how to map whole Process Image defined in ENI by one PI Client.

5.1.1.3. 33_BenchMultiThread sample improvement

33_BenchMultiThread sample application has been modified to support working with enabled the hardware timed send feature.

5.1.2. Fixed bugs

- Insufficient description of Configuration Pool in API Help.
- Error ENI Loading in 05_ManualCycleHandling sample.
- 07_ProcessImageDelivery sample continues to work after PI variable mapping failed.

5.2. FreeRTOS

5.2.1. Features

5.2.1.1. MIP package update

The **ip_cores** folder has been removed from MIP package, because this is a part of HW_SAMPLE package.

5.2.1.2. MIP samples improvement

The **hw_platform** project of Xilinx SDK has been added to the MIP package to run MIP samples via Xilinx SDK.

5.2.1.3. .bif files update

The boot folder for all CPU packages has been updated:

- .bif files were updated and renamed;
- readme was updated;
- the way of creating boot.bin for R5 CPU was simplified;
- documentation was updated and improved.

5.2.1.4. macro.h improvement

macro.h file has been updated:

- removed unused INI parameters;
- added descriptions of some INI parameters;
- added define `KPA_MASTER_TASK_PRIORITY` to simplify setting of user task priority;
- removed unused code.

5.2.1.5. BSP improvement

Now **configUSE_TASK_FPU_SUPPORT** is set to 2 by default, to avoid corrupting of FPU registers while context restoration after IRQ.

5.2.1.6. Hardware Scheduler support

From now on the Hardware Scheduler feature is supported by Masters which are implemented on the FPGA platform.

Benefits of this feature:

- increase of Master performance by decreasing EtherCAT frame preparation time;
- allows to achieve smaller cycle period;
- up to 100% bus utilization.

To enable the feature, the INI parameter **master.hwscheduler.enable** is used and the hardware timed send feature should be enabled too.

5.2.1.7. Xilinx SDK projects update

Xilinx SDK project files for all Master samples have been added to regular MIP delivery. Debug and Release build configurations have been set to build and run samples from the package with using reference BSP and HW_PLATFORM.

5.2.2. Fixed bugs

- Incorrect implementation of the `SLEEP()` macros in `macro.h`.

6. MDK version 2.4.47806.0, 2021-09-24

6.1. Linux

6.1.1. Features

6.1.1.1. Салют-ЭЛ240М1 support

From now on Салют-ЭЛ240М1 is supported.

7. MDK version 2.4.46804.0, 2021-08-17

7.1. Common

7.1.1. Features

7.1.1.1. PI mapping error trace event

A new trace event has been added to show the intersection process image offset in case of mapping error caused by write access protection.

7.1.2. Fixed bugs

- External Distributed Clock does not work.

7.2. Linux

7.2.1. Features

7.2.1.1. Driver Integration Package for BeagleBone Black Xenomai

Driver Integration Package for BeagleBone Black Xenomai has been added and it consists of kernel module and KPA network driver. The timed send emulation feature is supported by the driver among others.

7.2.1.2. Network driver as dedicate dynamic library

From now on KPA network driver is provided as dedicate dynamic library (netdrvrip.so) within Driver Intergration Package (DIP) for target board.

7.2.1.3. Ubuntu 16.04 support

From now on MRT binaries are available for Ubuntu 16.04.

7.3. Xenomai

7.3.1. Features

7.3.1.1. Driver Integration Package for BeagleBone Black Xenomai

Driver Integration Package for BeagleBone Black Xenomai has been added and it consists of kernel module and KPA network driver. The timed send emulation feature is supported by the driver among others.

7.3.1.2. Network driver as dedicate dynamic library

From now on KPA network driver is provided as dedicate dynamic library (netdrvrip.so) within Driver Intergration Package (DIP) for target board.

7.3.2. Fixed bugs

- Default value of memory pool size in INI file is less than 32Mb.

7.4. RTX

7.4.1. Fixed bugs

- 16cpp_04_MasterStartupSyncMode1and2 sample can't go to the Operational state with enabled "separate IO update".

7.5. INtime

7.5.1. Fixed bugs

- 16cpp_04_MasterStartupSyncMode1and2 sample can't go to the Operational state with enabled "separate IO update".

8. MDK version 2.4.45419.0, 2021-09-24

8.1. Common

8.1.1. Fixed bugs

- Master does not work as DC reference clock.

9. MDK version 2.4.45405.0, 2021-06-25

9.1. Common

9.1.1. Features

9.1.1.1. Support of C++ Samples from MIP 1.6

From now on all Samples for C++ language of MIP 1.6 are available and work in MIP 2.x.

9.1.2. Fixed bugs

- Master error 0x803A occurs at the end of C sample 31_BenchSingleThreadParallelBuslo.

10. MDK version 2.4.44108.0, 2021-06-14

10.1. Common

10.1.1. Features

10.1.1.1. Timeouts handling improvement

A new timer LocalTimeMonotonic has been added to avoid wrong handling of frame receive timeouts.

The corresponding variable nsLocalTimeMonotonic was added to Process Image.

10.1.2. Fixed bugs

- Exception appears on the second attempt to access CoE with CompleteAccess service.
- RPC Server hangs at Master state switching while SoE Reading OD.

10.2. RTX64

10.2.1. Features

10.2.2. RTX64 4.0.3 support

From now on Master supports RTX64 version 4.0.3.

10.2.3. Default Treads Priority scheme

From now on default values for thread priorities are implemented in Master INI file and they follow a default priority scheme .

For details, refer to MRT manual.

10.3. Rt-Preempt

10.3.1. Fixed bugs

- libecatmrpc.so name is used in .h files instead of libecatmsvr.so.

10.4. Windows

10.4.1. Features

10.4.1.1. Dedicated Tx thread for each channel

From now on in a network driver each channel for frame sending is handled in a dedicated Tx thread.

10.4.2. Fixed bugs

- Master with cable redundancy hangs when cable is unplugged from PC.
- Wrong DL status is shown in slave's InfoData variable (<Slave>.InfoData.DL Status).

10.5. Xenomai

10.5.1. Features

10.5.2. 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.

11. MDK for Rt-Preempt 2.4.44103.0, 2021-05-28

11.1. Features

11.1.1. Slave PDO indexes API

Getting information on slave's PDO indexes via API. Added the following functions.

To get slave's Tx/Rx PDO indexes:

```
EcatSlaveGetPDOIndexes(hMaster, wSlaveFixedAddr, dwFlags, pdwSize, pwIndexes)
```

Where

IN hMaster - master handle

IN wSlaveFixedAddr- slave physical address

IN dwFlags - 1 - RxPdo, 2- TxPdo

IN/OUT pdwSize - buffer size (items) / actual items count

OUT pwIndexes - array to store indexes of the slave

To get information on slave's PDO indexes:

```
EcatSlaveGetPDOInfo(hMaster, wSlaveFixedAddr, wPdoIndex, pPDOInfo, pdwNameSize, ptsNameBuff)
```

Where

IN hMaster - master handle

IN wSlaveFixedAddr- slave physical address

IN wPdoIndex - PDO index

OUT pPDOInfo - pointer to store slave PDO Information

IN/OUT pdwNameSiz - buffer size in characters / actual name size excluding '\0'

OUT ptsNameBuff- array to store indexes of the slave

To get information on PDO entries:

```
EcatSlaveGetPDOEntries(hMaster, wSlaveFixedAddr, wPdoIndex, pdwSize, phVariables)
```

Where

IN hMaster - master handle

IN wSlaveFixedAddr- slave physical address

IN wPdoIndex - PDO index

IN/OUT pdwSize - buffer size (items) / actual items count

OUT phVariables - array to store variable handles of the slave

To get an example of these functions usage, refer to 17_SlaveValues sample which was updated with using new API functions and printing of PDO Information .

11.1.2. IPv6 over EoE via INI parameter

From now on it is possible to configure the use of IPv6 over EoE with the help of INI parameter. The parameter **ethdev.ipv6.master** was added. By default, its value is 0 that means the use of IPv6 is disabled. To enable it, set 1 to this parameter.

11.1.3. RPC synchronization queue length INI parameter

To configure the length of RPC server queue, INI parameter **rpc.syncqueue_length** has been added.

11.1.4. MIP 1.6 compatibility

MRT 2.4 is compatible with MIP 1.6. From now on it is possible to install new version of MRT 2 with MIP 1.6. It allows using an application for MDK 1.6 with extended functionality of MRT 2.4.

11.2. Fixed bugs

- Failed to scan a big bus (with more then 500 slaves).

12. MDK for RTX64 2.4.44101.0, 2021-05-14

12.1. Features

12.1.1. Virtual COM improvements

Added Virtual COM diagnostic messages. To get their description and remedies, refer to the Troubleshooting section in MRT Manual.pdf.

12.1.2. IPv6 over EoE via INI parameter

From now on it is possible to configure the use of IPv6 over EoE with the help of INI parameter.

The parameter **ethdev.ipv6.master** was added. By default, its value is 0 that means the use of IPv6.

12.1.3. Slave PDO indexes API

Getting information on slave's PDO indexes via API. Added the following functions.

To get slave's Tx/Rx PDO indexes:

```
EcatSlaveGetPDOIndexes(hMaster, wSlaveFixedAddr, dwFlags, pdwSize,  
pwIndexes)
```

Where

IN hMaster - master handle

IN wSlaveFixedAddr- slave physical address

IN dwFlags - 1 - RxPdo, 2- TxPdo

IN/OUT pdwSize - buffer size (items) / actual items count

OUT pwIndexes - array to store indexes of the slave

To get information on slave's PDO indexes:

```
EcatSlaveGetPDOInfo(hMaster, wSlaveFixedAddr, wPdoIndex, pPDOInfo,  
pdwNameSize, ptsNameBuff)
```

Where

IN hMaster - master handle

IN wSlaveFixedAddr- slave physical address

IN wPdoIndex - PDO index

OUT pPDOInfo - pointer to store slave PDO Information

IN/OUT pdwNameSiz - buffer size in characters / actual name size excluding '\0'

OUT ptsNameBuff- array to store indexes of the slave

To get information on PDO entries:

```
EcatSlaveGetPDOEntries(hMaster, wSlaveFixedAddr, wPdoIndex, pdwSize,  
phVariables)
```

Where

IN hMaster - master handle

IN wSlaveFixedAddr- slave physical address

IN wPdoIndex - PDO index

IN/OUT pdwSize - buffer size (items) / actual items count

OUT phVariables - array to store variable handles of the slave

To get an example of these functions usage, refer to 17_SlaveValues sample which was updated with using new API functions and printing of PDO Information .

12.1.4. RPC synchronization queue length INI parameter

To configure the length of RPC server queue, INI parameter **rpc.syncqueue_length** has been added.

By default it is disabled. To enable it, set 1 to this parameter.

12.1.5. MRT Installer improvement

Implemented additional verification of NIC addresses set by the user for RTX and Windows sides.

The user will be notified when there is a conflict between set addresses.

For details on specifics of setting NIC addresses, refer to MDK.Readme.pdf.

12.2. Fixed bugs

- Master crashes while attaching to the bus on the load ENI stage.
- Wrong data delivery due to non-sequential mapping for pi delivery client.
- 19_MailboxMultipleRequests sample crashes with an exception.
- Invalid configuration is created if there is a hot-plugged segment with another nested hot-plug segment.
- Exception appears on Master stop if Cable Redundancy was configured.
- Cannot initialize licensing.
- KPARTXFilterDrv crashes during initialization if there is a frame exchange on the NIC.
- Cannot start an external control task.

13. MDK for Rt-Preempt 2.4.29549.0, 2020-12-14

13.1. Features

13.1.1. Timeouts handling improvement

A new timer LocalTimeMonotonic has been added to avoid wrong handling of frame receive timeouts.

The corresponding variable nsLocalTimeMonotonic was added to Process Image.

13.2. Fixed bugs

- Wrong DL status is shown in slave's InfoData variable (<Slave>.InfoData.DL Status).

14. MDK for RTX64 2.4.37800.0, 2020-11-27

14.1. Features

14.1.1. Explicit identification improvements

To speed-up slaves identification and reduce bus access methods, added a prohibition of explicit identification for devices which can be identified by their fixed address.

To reduce a Master start up time and the time of detecting topology changes, from now on a custom logic is used to read explicit identification value from the device.

14.1.2. Dynamic configuration in 05_ManualCycleHandling sample

From now on 05_ManualCycleHandling sample demonstrates how to dynamically allocate a configuration pool.

14.1.3. Mailbox requests limit depending on slaves number

Improved the way of setting mailbox requests limit.

From now on, if it is not set by the INI parameter **mailbox.maxrequestcount**, it is calculated depending on the number of slaves in the configuration which support mailbox communication.

14.1.4. 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

14.1.5. 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.

14.1.6. 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.

14.1.7. New samples

New samples have been added:

23_AC_SlaveConfigSelect - demonstrates how to use Autoconfigurator functionality.

24_DriveRotationCiA402 - demonstrates how to work with Drive Rotation CiA402.

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.

14.1.8. PI clients for reading / writing

Added PI clients with corresponding variables for reading / writing into CAN Interface driver.

14.1.9. Remote Autoconfigurator

From now on it is possible to access to Autoconfigurator remotely. Added:

- RPC API for Autoconfigurator;
- ecatmkpa-python improvements to scan and configure master.

14.1.10. Timed send support in samples

From now on the samples ManualCycleHandling and MultipleCyclesHandling support timed send mode.

14.1.11. RTX64 3.7 support

From now on Master supports RTX64 version 3.7.

14.1.12. MIP 1.6 compatibility

MRT 2.4 is compatible with MIP 1.6. From now on it is possible to install new version of MRT 2 with MIP 1.6. It allows using an application for MDK 1.6 with extended functionality of MRT 2.4.

14.1.13. 4-hour Demo time

We increased time of Master operating in Demo mode and for Trial versions. From now on an operating time limitation is four hours.

14.1.14. PI variable(s) for all inputs/outputs of the slave

Added PI variable(s) for all inputs/outputs of the slave.

14.1.15. Default transferring timeout calculation

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

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

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

14.1.16. 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 **oe.adapter** (default value is "opentaptun").

14.1.17. 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.

14.1.18. 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.

14.1.19. 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).

14.1.20. Asynchronous mailbox call EcatRPCReadMemoryByPosIndex

From now on EcatRPCReadMemoryByPosIndex() is called asynchronously.

14.1.21. 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.

14.1.22. 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.

14.1.23. EEPROM operations improvement

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

14.1.24. KPA Framework improvement

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

14.1.25. Python application for RPC client

New Python application to work with RPC client.

14.1.26. Writing PI inputs API

Added API functions EcatStartWriteInputs and EcatDoneWriteInputs to write Process Image inputs.

14.2. Fixed bugs

- Reading DL status cycles in case of cable redundancy break in the line.
- Framework initialization processes cycles (Init-OP cycle) if final initialization target is changed from "Ready" to "Request State".
- Memory leak happens at reading slave memory.
- No 'duplicate alias' events are printed by the sample or user application due to missing description of the event.
- A lot of Wrong Working counters events are generated.
- In case of second attempt of explicit identification for the same physical device it always accepts configured identification value regardless of the result of the value check.
- Command size limit is not checked while parsing ENI file and building a frame.

15. MDK version 2.4.29528.0, 2020-10-29

15.1. Common

15.1.1. Features

15.1.1.1. Explicit identification improvement

To speed-up slaves identification and reduce bus access methods, added a prohibition of explicit identification for devices which can be identified by their fixed address.

15.1.1.2. Auto-start RPC server in the 08_RPCServer sample

In order to simplify RPC server start/stop procedure, added a new template parameter `PARAM_MASTER_RPC_SERVER_CREATE`.

Set this parameter to enable automatic start/stop of RPC server with specified port number (default: 5000). Then Master Processor itself starts RPC server at Master creating and stops it at Master destroying.

15.1.1.3. Mailbox requests limit depending on slaves number

Improved the way of setting mailbox requests limit. From now on, if it is not set by the INI parameter **mailbox.maxrequestcount**, it is calculated depending on the number of slaves in the configuration which support mailbox communication.

15.1.1.4. Dynamic configuration in 05_ManualCycleHandling sample

From now on 05_ManualCycleHandling sample demonstrates how to dynamically allocate a configuration pool.

15.1.2. Fixed bugs

- No 'duplicate alias' events are printed by the sample or user application due to missing description of the event.
- A lot of Wrong Working counters events are generated.
- Exception happens on mailbox call if slave physical address is 0.
- Exception may happen on Master exit if mailbox communication is enabled.
- In case of second attempt of explicit identification for the same physical device it always accepts configured identification value regardless of the result of the value check.

16. MDK version 2.4.29520.0, 2020-09-25

16.1. Common

16.1.1. Feature

16.1.1.1. 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.

16.1.1.2. 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.

16.1.1.3. 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.

16.1.1.4. 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.

16.1.1.5. 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
$\dots < 64$	4096
$64 \leq \dots \leq 256$	8192
$256 \leq \dots$	16384

16.1.1.6. 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.

16.1.1.7. 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.

16.1.1.8. 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".

16.1.1.9. 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.
```

16.1.1.10. 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.

16.1.1.11. 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)

16.1.1.12. Asynchronous mailbox call EcatRPCReadMemoryByPosIndex

From now on EcatRPCReadMemoryByPosIndex() is called asynchronously.

16.1.1.13. 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.

16.1.1.14. 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.

16.1.2. EEPROM operations improvement

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

16.1.2.1. KPA Framework improvement

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

16.1.2.2. 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.

16.1.2.3. Python application for RPC client

New Python application to work with RPC client.

16.1.3. 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.
- 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.
- Last slave on the bus is missing or not recognized.
- Memory leak happens at reading slave memory.

16.2. Windows

16.2.1. Features

16.2.1.1. 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.e.adapter** (default value is "opentaptun").

16.2.1.2. 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.

16.2.1.3. Review of Windows PCAP driver usage

Improved safety, synchronization and performance of PCAP driver usage.

16.2.2. Fixed bugs

- Master default folder is not available in the network.
- Attach to Master configured with non-existing network card is successful.

17. MDK for INtime 2.4.26504.0, 2020-06-18

17.1. Features

17.1.1. 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 among 2013, 2015 or 2017 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).

17.1.2. 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.

17.1.3. 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.

17.1.4. 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.

17.1.5. 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.

17.2. Fixed bugs

- API Help item in the Start menu is displayed with a wrong name.
- Slaves follow master state if transition.disablesslavefollowstate=1.

18. MDK for Windows 2.4.25003.0, 2020-05-25

18.1. Features

18.1.1. Autoconfigurator

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).

18.1.2. 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).

18.1.3. 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.

18.2. 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.disable_slave_follow_state=1`.

19. MDK version 2.3.43629.0, 2020-04-20

Initial release of KPA EtherCAT Master Development Kit 2.3 for Rt-Preempt, 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

19.1. Rt-Preempt

19.1.1. Features

- Redundant Master
- Hardware Timed send mode
- Autoconfigurator

20. MDK version 2.4.15804.0, 2019-12-27

Initial release of KPA EtherCAT Master Run-time 2.3 for INtime, Xenomai 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
- Autoconfigurator

21. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.41907.0, 2019-10-18

21.1. Features

21.1.1. Separate IP core for licensing

Maser licensing has been moved into the separated IP core (KPA License Provider) as an interface for getting FPGA device DNA. The corresponding hardware project and BSP were updated in accord with new IP core usage.

For more detail, refer to `Zynq.FreeRTOS.Manual.pdf`.

21.1.2. Xilinx Device Primitives by default

From now on Xilinx Device Primitives are used in IP cores by default. The define `XILINX_KPA` was deleted from the corresponding hardware project, IP cores were updated in accord with new default primitives usage.

21.2. Fixed bugs

- Hardware timed send feature does not work with A53 and R5 processors (Zynq Ultrascale+ board).
- Incorrect memory allocation. When it is impossible to allocate predefined memory pool during Master initialization, Master continues to work with system memory allocation calls instead of reporting an error.
- Benchmark sample returns the value of a current allocated memory instead of a maximum allocated memory value.

22. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.39209.0, 2019-07-31

22.1. Features

22.1.1. Hardware-bound license

From now on it is possible to bind Master license to the target where it runs.

For more details, refer to MRT Manual (Licensing).

22.1.2. New slave's variables in Process Image

From now on variables InputAll and OutputAll are available in Process image for each slave. They describe total size (BitSize) and location (BitStart) of slave's Process Data in Process Image.

<slave_name>.InputAll.BitSize - total size of received data

<slave_name>.InputAll.BitStart - start address of received data in Process Image

<slave_name>.OutputAll.BitSize - total size of sent data

<slave_name>.OutputAll.BitStart - start address of sent data in Process Image

22.2. Fixed bugs

- Master hangs after transferring it from Init to Bootstrap and then to Init state if EL6695 is in the bus configuration.
- An attempt to scan bus configuration is failed if the configuration was extended (e.g. added one more segment) after last successful scan.
- Master hangs on attempt to read CoE sub-indexes via SDO.
- It is impossible to attach to Master if it was detached in Safe Operation/Operational state.
- Connection with Master is lost on attempt to scan bus configuration from KPA EtherCAT Studio.

23. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.38100.0, 2019-06-28

23.1. Features

23.1.1. 4 hours Demo time

We increased time of Master operating in Demo mode and for Trial versions. From now on an operating time limitation is four hours.

23.1.2. Redundant Master support

KPA provides a new master feature - Redundant Master. It allows co-existing of several master instances started on separate target systems in a network segment, significantly increasing fault tolerance and reducing downtime. All master instances are connected to the bus and run.

For more details, refer to MRT Manual.pdf (Redundant Master section).

23.1.3. RPC Server support

From now on it is possible to get access to the bus which is connected to FreeRTOS target via RPC Server (for example, scanning the bus in KPA EtherCAT Studio).

24. MDK for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.19400.0, 2018-12-17

Initial release of KPA EtherCAT Master Development Kit for Zynq/Zynq Ultrascale+ FreeRTOS version 2.3.

25. MRT for RTX64 2.2.10600.0, 2018-01-31

Initial release of KPA EtherCAT Master Run-time 2.2 for RTX64 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

26. Imprint and disclaimer

26.1. Trademarks

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

straton® is a registered trademark of Ing. Punzenberger COPA-DATA GmbH, Austria.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

26.2. Disclaimer

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event, shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

26.3. Quality Management



koenig-pa GmbH Quality Management System is certified according to DIN EN ISO 9001:2015. Initial certification in 2008.

26.4. Contacts

For more information about products and services, please visit company website: www.koenig-pa.de. For getting technical support or solving issues arise from the use of our products, there is a ticketing system in the [Customer Portal](#) where you can apply for assistance. Note that the Customer Portal is available for registered users only.

In urgent cases, you have the following options:

- Contact resellers in your country or region.
- Get assistance by phone: +49 9128 725 614, +49 9128 725 330.
- Contact our Support Team at support@koenig-pa.de.

26.5. Mailing address

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

26.6. Copyright

© koenig-pa GmbH, Germany. All rights reserved.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.