



Master Development Kit version 1.6. What's New

1 Multimaster

The Multimaster functionality allows a control application to service several independent EtherCAT® buses from the same target system simultaneously. Master 1.6 API provides an ability to create several master instances. Each master instance has its own unique identifier, so API calls can distinguish the right one. Master 1.5, in contrast to Master 1.6, uses the Master library (dll) that allows creating only one master instance.

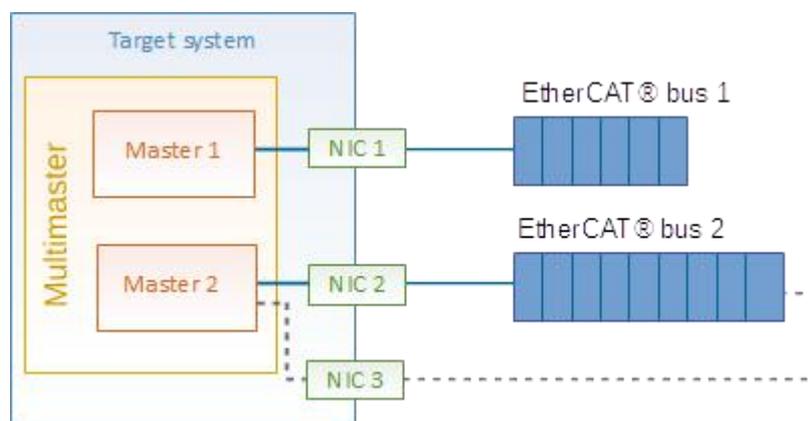


Figure 1: Example of the Multimaster usage

The Multimaster functionality provides the following features:

- Creating different configurations for each Master instance in one Master configuration file (different section of INI file) and corresponding API for working with them.
- Resources management (getting number of Master instances, getting NIC-s used by each Master instance, system timer).
- RPC for each Master instance.

Multimaster capabilities help to simplify the control of various complex systems.

A real system may already have independent control circuits (different processes, independent physical interfaces, isolated functional units). The Multimaster functionality allows splitting service tasks into isolated units and handling them separately.

In many cases a control system has the main unit and some subsidiary units (for diagnostics purpose, as temporary parts or even extension modules). An isolated master instance keeps the main functionality untouched. At the same time, another master instance may handle the subsidiary units which allows to extend the system functionality.

Often critical system parts may have strict execution requirements such as short control cycle, response time or even safety rules. Bringing them into one isolated circuit helps to meet the requirements flexibly. Using an independent unit allows controlling application tasks priority and system resources allocation precisely.

KPA Master library supports two execution models. With the first one, each master instance is launched in separate process. It allows implementing completely independent EtherCAT® buses and minimizing

impact one master on other. With the second model called “local Multimaster”, all master instances are created in the same application process. In this case, all master instances are as close to each other as possible. It brings flexible synchronization, minimum delays and fast response. It is suitable for the cases when Multimaster tasks are correlated.

Multimaster usage makes possible to synchronize work of several independent EtherCAT® buses. In case of synchronization with reference clock device, Master adjusts system time unit to follow the reference clock time. The resource management system allows only one master instance to adjust system time. Other master instances can use External Reference clock device (e.g. Master-to-Master) or “Master as external reference clock device” (when Master itself is a Reference Clock device).

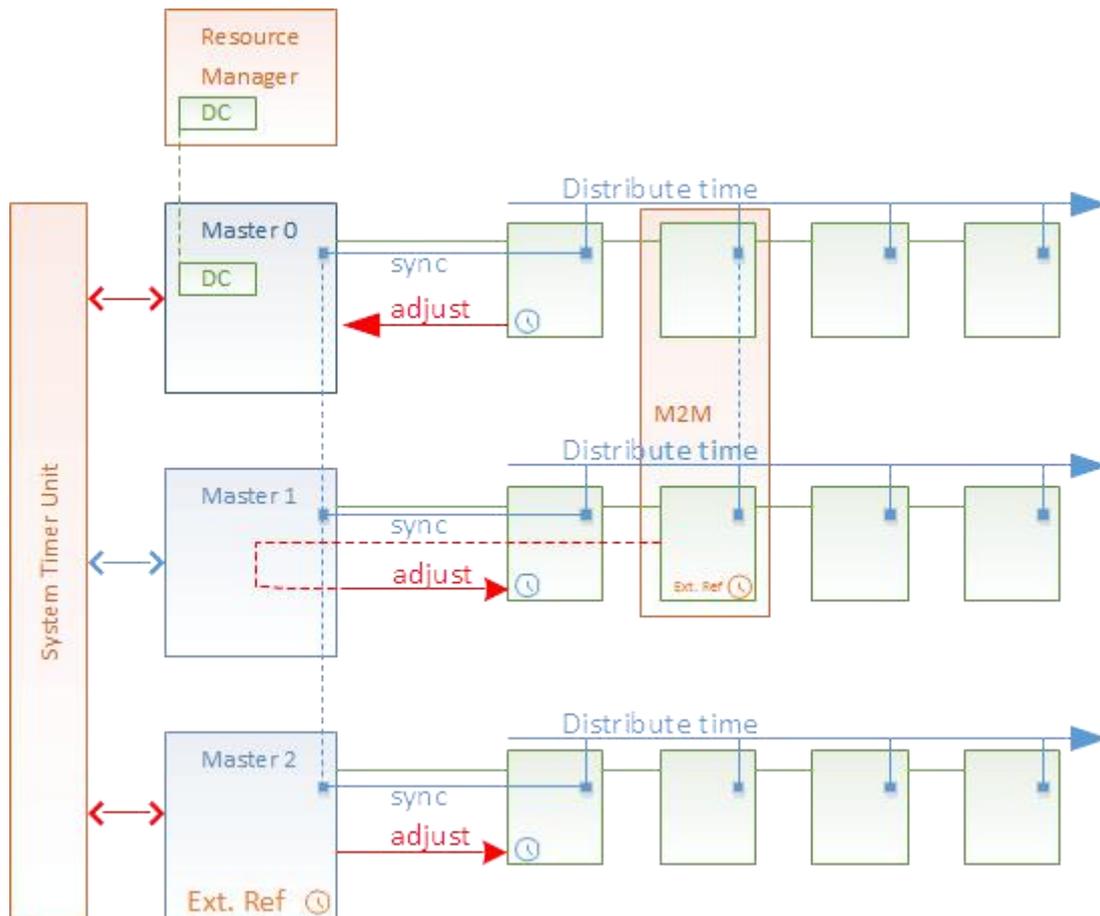


Figure 2: Example of Multimaster DC synchronization

The Multimaster functionality is provided as a Feature Pack that should be additionally activated through the license file.

2 Event Handler

Master 1.6 provides the Event Handler to obtain information about state of the system (e.g. for Diagnostics) and to synchronize different parts of a control application (e.g. PLC) by using custom events (or Master-specific events). Now it is not necessary to poll the bus diagnostics periodically as it was in the previous versions (e.g. Master 1.5). The user may do it only when something happens (by getting the event).

The Event Handler is an object which allows capturing, filtering and delivering Master internal events to the user application. The Event Handler API provides access to the specific data of the received event.

All events are generated in the identical format.

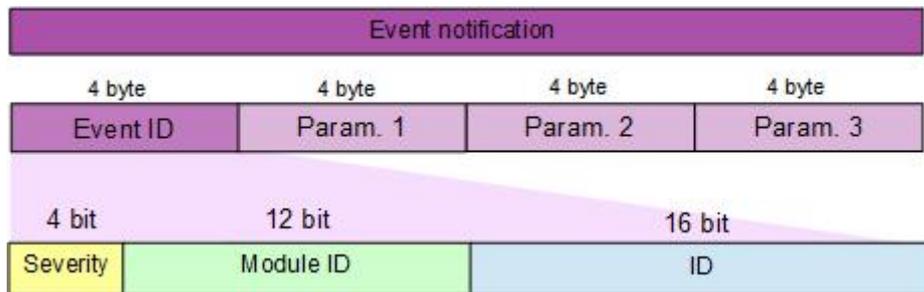


Figure 3: Event notification format

Where **Severity** – severity of Event notification, allowed values: INFO, WARNING, ERROR or TRACE (can be used for custom events).

Module ID – identifier of Master module which generates corresponding Event notification, e.g. Master module for diagnosing the bus topology (the bus state, connection of devices etc.), or slaves (their current and requested states, failed init commands etc.), or Master (its state, invalid configuration, broken connections etc.).

ID – event identifier in the Master module.

To get required notifications, it is possible to set various filters (by ID or by Severity) for each handler separately.

For example, the Event Handler can be used in the following cases:

- To select diagnostic data. The user may configure its application to listen to certain diagnostic events instead of getting the complete Master diagnostics.
- To decrease system resources utilization. While using Event Handler functionality, the user may acquire the diagnostics when the Master/Slave state changes and the corresponding diagnostic events appear. Therefore, the user does not need to acquire the diagnostics periodically and may use the system resources spent on it for additional tasks implementation.
- To synchronize the work of several Masters running on the same PC and combined into the one system. For example, let us assume the Multimaster is used in water management system. The first Master instance controls the pumping of water through the system (from one functional node to another, several nodes in a row). The other instances control water treatment, a separate instance for each functional node (e.g. filtration, heating, etc.). Consider the node of water heating. The first Master fills the tank with cold water (Task 1) and generates Event 1. The second Master receives Event 1. It indicates that the tank has been filled and the second Master starts heating (Task 2). After completing Task 2, the second Master generates Event 2. It indicates to the first Master that the water temperature has achieved the required value and can be pumped out of the tank to the next node (Task 3).

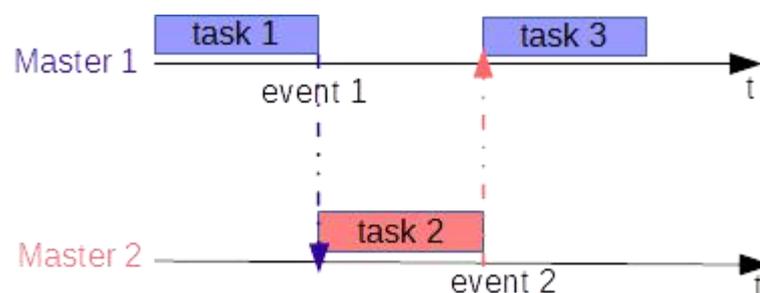


Figure 4: Example of Masters synchronization by using events

The Event Handler functionality is provided as a Feature Pack that should be additionally activated through the license file.

3 Explicit Device Identification

Master 1.6 completely supports the Explicit Device Identification (EDI) as defined by EtherCAT specification. This is necessary for the following cases:

- Hot Connected slaves' identification.
- Prevention against cable swapping (in case there are at least two identical devices on the bus).

In addition to reading/writing alias from EEPROM supported by Master 1.5, Master 1.6 supports the following mechanisms of Explicit Device Identification:

- Requesting ID
- Direct ID
- SII Configured Station Alias

The Explicit Device Identification is provided as a part of Master Basic Class, therefore it does not require additional activation through the license file.

4 Virtual Serial COM driver

KPA offers an easy way of data exchange with devices with serial interface (RS232/RS422/RS485). The KPA Virtual Serial COM driver has been added to KPA MRT package. It allows you to communicate with the serial device from your Windows application via Virtual COM port. Master transmits data got from the serial terminal / the Virtual COM device to the recipient.

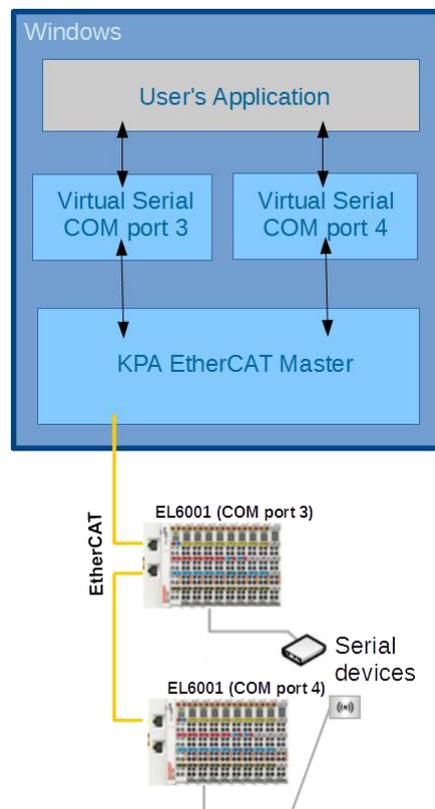


Figure 5: Example of Virtual Serial COM driver implementation

KPA Virtual COM driver is installed during installation of KPA EtherCAT Master version 1.6 or higher. The corresponding feature must present in Master's license. It is not required additional configuring with COM settings (baudrate, data bits, parity etc.).

More details see in the documentation on the corresponding KPA EtherCAT Master.

The EL600x/EL602x serial interface terminals enable the connection of the serial interface devices. To configure the terminal, KPA provides the Virtual Serial COM plug-in. More details about the plug-in, see in Studio Help.

The Virtual Serial COM driver and plug-in are interconnected. Therefore, for a proper work both the driver and the plug-in must be installed and the corresponding features must be added into the products' licenses.

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

