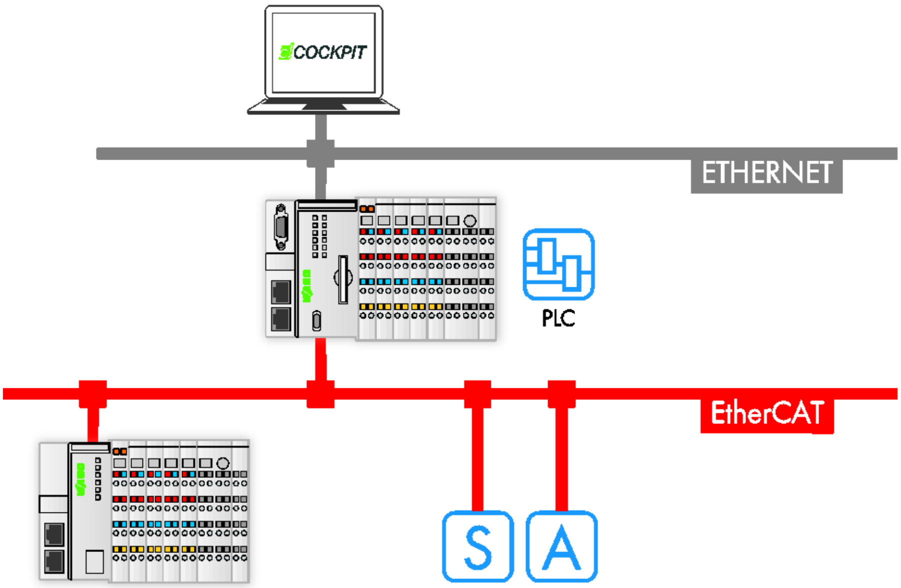
TECHDOCS

Technical Information

WAGO Software



2759-026x/0210-1000

e!RUNTIME EtherCAT Master

Controller PFC 200, 2. Generation and Touch Panels 600

Version 1.1.1, valid from FW/HW Version 13 and e!COCKPIT Version 1.5.1

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: www.wago.com

Technical Support

Phone: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

E-Mail: documentation@wago.com

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally protected by trademark or patent.

WAGO is a registered trademark of WAGO Verwaltungsgesellschaft mbH.

Table of Contents

1 Notes about this Documentation

Note

Always retain this documentation!
This documentation is part of the product. Therefore, retain the documentation during the entire service life of the product. Pass on the documentation to any subsequent user. In addition, ensure that any supplement to this documentation is included, if necessary.

1.1 Validity of this Documentation

This documentation applies to the licensed functionality “***e!****RUNTIME* EtherCAT® Master 300” used together with the ***e!****COCKPIT* software and a second-generation WAGO PFC200 Series Controller or a Touch Panel 600.

This documentation is only applicable from FW/HW Version 13 and e!COCKPIT Version 1.5.1.

Note

Observe other Applicable Documentation!
In addition to this documentation, also observe additional instructions and information provided in the operating instructions for the software and devices used. Download the operating instructions for the ***e!****COCKPIT* software and controller used (PFC200, second generation or Touch Panel 600) from the download area of the Internet site http://www.wago.com.

1.2 Copyright

This Manual, including all figures and illustrations, is copyright-protected. Any further use of this Manual by third parties that violate pertinent copyright provisions is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying) as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will involve the right to assert damage claims.

1.3 Property rights

Third-party trademarks are used in this documentation. This section contains the trademarks used. The “®” and “TM” symbols are omitted hereinafter.

- Adobe® and Acrobat® are registered trademarks of Adobe Systems Inc.
- AS-Interface® is a registered trademark of AS-International Association.
- BACnet® is a registered trademark of American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc. (ASHRAE).
- *Bluetooth*® is a registered trademark of the Bluetooth SIG, Inc.
- CiA® and CANopen® are registered trademarks of CAN in AUTOMATION – International Users and Manufacturers Group e. V.
- DALI is a registered trademark of Digital Illumination Interface Alliance (DiiA).
- EtherCAT® is a registered trademark and patented technology of Beckhoff Automation GmbH.
- EtherNet/IP™ is a registered trademark of Open DeviceNet Vendor Association, Inc (ODVA).
- EnOcean® is a registered trademark of EnOcean GmbH.
- IO-Link is a registered trademark of PROFIBUS Nutzerorganisation e.V.
- KNX® is a registered trademark of KNX Association cvba.
- Linux® is a registered trademark of Linus Torvalds.
- LON® is a registered trademark of Echelon Corporation.
- Modbus® is a registered trademark of Schneider Electric, licensed to the Modbus Organization, Inc.
- PROFIBUS® is a registered trademark of Siemens AG.
- PROFINET® is a registered trademark of Siemens AG.
- Subversion® is a registered trademark of Apache Software Foundation.
- Windows® is a registered trademark of Microsoft Corporation.

1.4 Symbols

⚠ DANGER

Personal Injury!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ DANGER

Personal Injury Caused by Electric Current!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING

Personal Injury!
Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION

Personal Injury!
Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE

Damage to Property Caused by Electrostatic Discharge (ESD)!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note

Important Note!
Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

Information

Additional Information:
Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

1.5 Number Notation

Table 1: Number Notation

Number Code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

1.6 Font Conventions

Table 2: Font Conventions

Font Type	Indicates
italic	Names of paths and data files are marked in italic-type. e.g.: *C:\Program Files\WAGO Software*
Menu	Menu items are marked in bold letters. e.g.: **Save**
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: **File** > **New**
Input	Designation of input or optional fields are marked in bold letters, e.g.: **Start of measurement range**
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under **Start of measurement range**.
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: **[Input]**
[Key]	Keys are marked with bold letters in square brackets. e.g.: **[F5]**

2 Important Notes

This section includes an overall summary of the most important safety requirements and notes that are mentioned in each individual section. To protect your health and prevent damage to devices as well, it is imperative to read and carefully follow the safety guidelines.

2.1 Legal Bases

2.1.1 Subject to Changes

WAGO Kontakttechnik GmbH & Co. KG reserves the right to provide for any alterations or modifications. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

2.1.2 Personnel Qualifications

All sequences implemented on WAGO-I/O-SYSTEM 750 devices may only be carried out by electrical specialists with sufficient knowledge in automation. The specialists must be familiar with the current norms and guidelines for the devices and automated environments.

All changes to the coupler or controller should always be carried out by qualified personnel with sufficient skills in PLC programming.

2.1.3 Use of the 750 Series in Compliance with Underlying Provisions

Fieldbus couplers, controllers and I/O modules found in the modular WAGO-I/O-SYSTEM 750 receive digital and analog signals from sensors and transmit them to actuators or higher-level control systems. Using controllers, the signals can also be (pre-) processed.

The devices have been developed for use in an environment that meets the IP20 protection class criteria. Protection against finger injury and solid impurities up to 12.5 mm diameter is assured; protection against water damage is not ensured. Unless otherwise specified, operation of the devices in wet and dusty environments is prohibited.

Operating the WAGO-I/O-SYSTEM 750 devices in home applications without further measures is only permitted if they meet the emission limits (emissions of interference) according to EN 61000-6-3. You will find the relevant information in the section “Device Description” > “Standards and Guidelines” in the manual for the used fieldbus coupler or controller.

Appropriate housing (per 2014/34/EU) is required when operating the WAGO-I/O-SYSTEM 750 in hazardous environments. Please note that a prototype test certificate must be obtained that confirms the correct installation of the system in a housing or switch cabinet.

The implementation of safety functions such as EMERGENCY STOP or safety door monitoring must only be performed by the F-I/O modules within the modular WAGO-I/O-SYSTEM 750. Only these safe F-I/O modules ensure functional safety in accordance with the latest international standards. WAGO's interference-free output modules can be controlled by the safety function.

2.1.4 Technical Condition of Specified Devices

The devices to be supplied ex works are equipped with hardware and software configurations, which meet the individual application requirements. These modules contain no parts that can be serviced or repaired by the user. The following actions will result in the exclusion of liability on the part of WAGO Kontakttechnik GmbH & Co. KG:

- Repairs,
- Changes to the hardware or software that are not described in the operating instructions,
- Improper use of the components.

Further details are given in the contractual agreements. Please send your request for modified and new hardware or software configurations directly to WAGO Kontakttechnik GmbH & Co. KG.

2.2 Safety Advice (Precautions)

For installing and operating purposes of the relevant device to your system the following safety precautions shall be observed:

⚠ DANGER

Do not work on devices while energized!
All power sources to the device shall be switched off prior to performing any installation, repair or maintenance work.

⚠ DANGER

Install device in only one suitable enclosure!
The device is an open system. Install the device in a suitable enclosure. This enclosure must:

- Guarantee that the max. permissible degree of pollution is not exceeded.
- Offer adequate protection against contact.
- Prevent fire from spreading outside of the enclosure.
- Offer adequate protection against UV irradiation.
- Guarantee mechanical stability
- Restrict access to authorized personnel and may only be opened with tools

⚠ DANGER

Ensure disconnect and overcurrent protection!
The device is intended for installation in automation technology systems. Disconnect protection is not integrated. Connected systems must be protected by a fuse.
Provide suitable disconnect and overcurrent protection on the system side!

⚠ DANGER

Ensure a standard connection!
To minimize any hazardous situations resulting in personal injury or to avoid failures in your system, the data and power supply lines shall be installed according to standards, with careful attention given to ensuring the correct terminal assignment. Always adhere to the EMC directives applicable to your application.

NOTICE

Do not use in telecommunication circuits!
Only use devices equipped with ETHERNET or RJ-45 connectors in LANs. Never connect these devices with telecommunication networks.

NOTICE

Ensure proper contact with the DIN-rail!
Proper electrical contact between the DIN-rail and device is necessary to maintain the EMC characteristics and function of the device.

NOTICE

Replace defective or damaged devices!
Replace defective or damaged device/module (e.g., in the event of deformed contacts).

NOTICE

Protect the components against materials having seeping and insulating properties!
The components are not resistant to materials having seeping and insulating properties such as: aerosols, silicones and triglycerides (found in some hand creams). If you cannot exclude that such materials will appear in the component environment, then install the components in an enclosure being resistant to the above-mentioned materials. Clean tools and materials are imperative for handling devices/modules.

NOTICE

Clean only with permitted materials!
Clean housing and soiled contacts with propanol.

NOTICE

Do not use any contact spray!
Do not use any contact spray. The spray may impair contact area functionality in connection with contamination.

NOTICE

Do not reverse the polarity of connection lines!
Avoid reverse polarity of data and power supply lines, as this may damage the devices involved.

NOTICE

Avoid electrostatic discharge!
The devices are equipped with electronic components that may be destroyed by electrostatic discharge when touched. Please observe the safety precautions against electrostatic discharge per DIN EN 61340-5-1/-3. When handling the devices, please ensure that environmental factors (personnel, work space and packaging) are properly grounded.

2.3 Special Use Conditions for ETHERNET Devices

If not otherwise specified, ETHERNET devices are intended for use on local networks. Please note the following when using ETHERNET devices in your system:

- Do not connect control components and control networks directly to an open network such as the Internet or an office network. WAGO recommends putting control components and control networks behind a firewall.
- Limit physical and electronic access to all automation components to authorized personnel only.
- Change the default passwords before first use! This will reduce the risk of unauthorized access to your system.
- Regularly change the passwords used! This will reduce the risk of unauthorized access to your system.
- If remote access to control components and control networks is required, use a Virtual Private Network (VPN).
- Regularly perform threat analyses. You can check whether the measures taken meet your security requirements.
- Use “defense-in-depth” mechanisms in your system's security configuration to restrict the access to and control of individual products and networks.

3 Features and Function

EtherCAT is a powerful real-time ETHERNET fieldbus system that has become standard in many industrial automation applications. With a software license, the second-generation PFC200 Controllers (750-821x/xxx-xxx) or Touch Panels 600 (762-xxxx/8000-xxxx) can run as an EtherCAT master.

The following functions are available:

- Distributed clocks
- Hot connect
- Bus diagnostics: In the configurator and with the PLC application
- Supported layer 7 protocols:
  - CoE (CANopen/CAN over EtherCAT)
  - FoE (File over EtherCAT)
  - VoE (Vendor over EtherCAT)

The Controller can be used as an EtherCAT Master to control WAGO slaves with the EtherCAT fieldbus system (e.g., the EtherCAT Fieldbus Couplers 750-354 of the WAGO-I/O-SYSTEM 750). It is also possible to control field devices, which can be declared in ***e!****COCKPIT* via a standardized device description (see Figure 1).

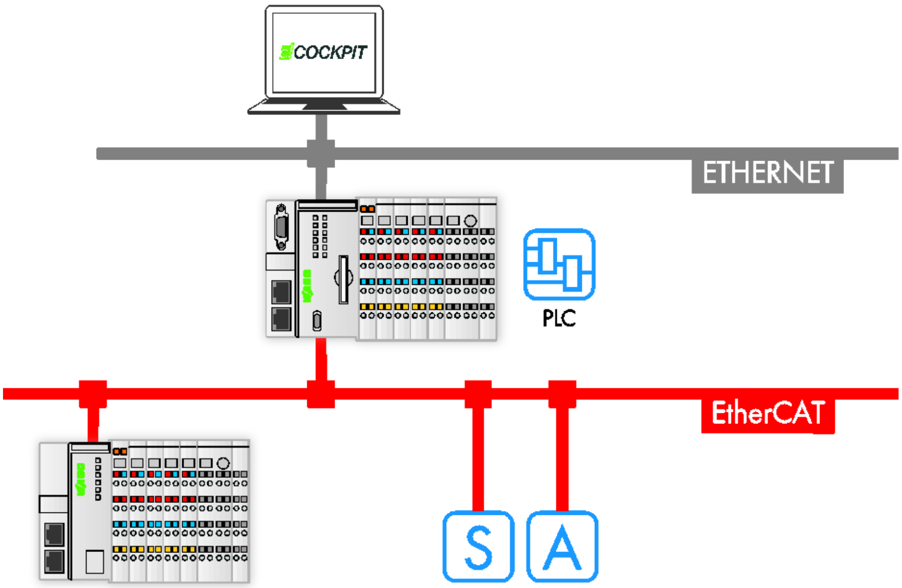


Figure 1: EtherCAT System: Example

Besides the protocol stack in the form of a library, the EtherCAT Master software also provides the run-time system with components for direct access to the ETHERNET interface and diagnostics.

The EtherCAT system is configured through a special configuration dialog in the ***e!****COCKPIT* engineering software. The following settings can be specified:

- Setting the controller as an EtherCAT Master
- Which field devices are communicated with
- Determining the network topology
- Parameter values to be sent to the slaves upon startup

Details of the setting dialogs and a simple example of setting up an EtherCAT network are described in the ***e!****COCKPIT* manual.

3.1 Relation between Bus Cycle and Task

Basically, for each IEC task, the input data used is read when the task starts (1) and the written output data is transferred to the I/O driver when it ends (3) (see figure below).
Implementation in the I/O driver is decisive for further transmission of the I/O data. It is responsible for determining during which time frame and at what specific time the actual transmission to the respective bus system occurs.

The PLC bus cycle task can be determined globally for all fieldbuses in the PLC settings. However, it is possible to change some of these tasks for specific fieldbuses independently of the global setting. The task with the shortest cycle time is automatically applied as the bus cycle task (setting: unspecified in the PLC settings). Normally, in this task the messages are sent to the bus.

Further tasks copy only the data from an internal buffer that is only exchanged with the real hardware during the bus cycle task.

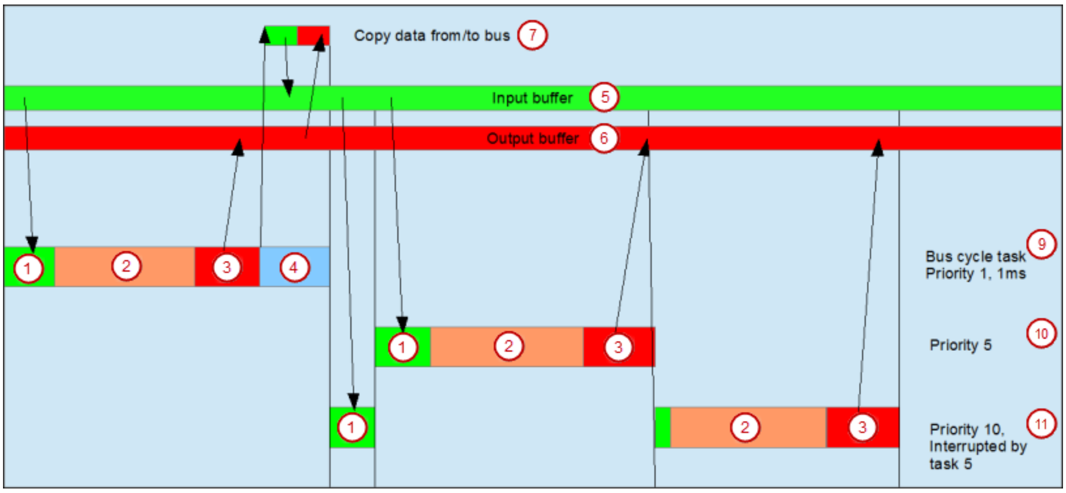

Copy data from/to bus 7
Input buffer 5
Output buffer 6
1
2
3
4
Bus cycle task Priority 1, 1ms 9
Priority 5 10
Priority 10, Interrupted by task 5 11

Figure 2: EtherCAT Tasks

Table 3: Legend to Figure “EtherCAT Tasks”

No.	Description
1	Read input buffer inputs.
2	IEC task
3	Write outputs in output buffer.
4	Bus cycle
5	Input buffer
6	Output buffer
7	Copy data from/to bus.
9	Bus cycle task, priority 1, 1 ms
10	Bus cycle task, priority 5
11	Bus cycle task, priority 10, interrupted by task 5

Before the IEC inputs are copied, the network messages pending from the last cycle are read.

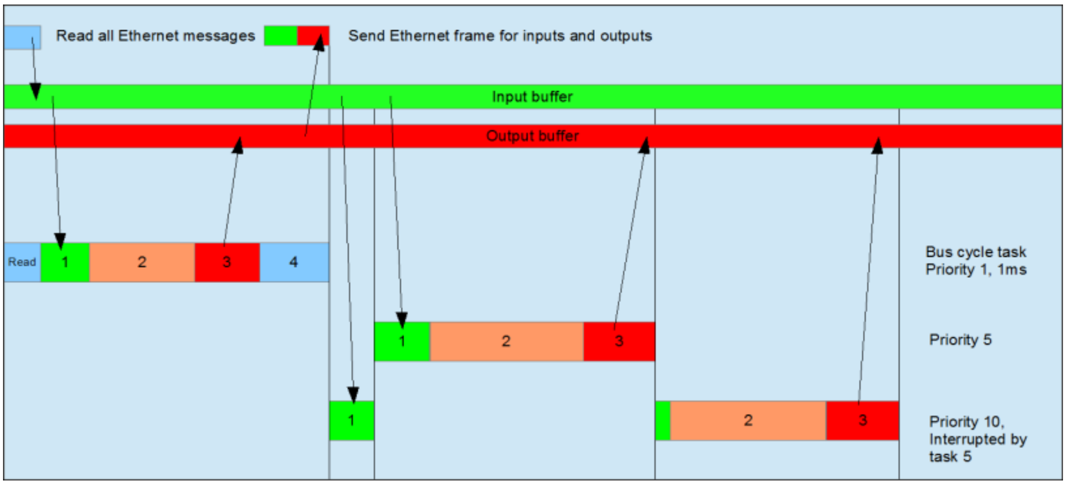


Figure 3: Task Processing without “Send/Receive per Task” Option

When the “Send/Receive per Task” option is activated in the EtherCAT Master settings, additional telegrams are sent for each task to the device and the inputs and outputs used. Channels used in a slow task are also transmitted less often. This helps reduce bus load.

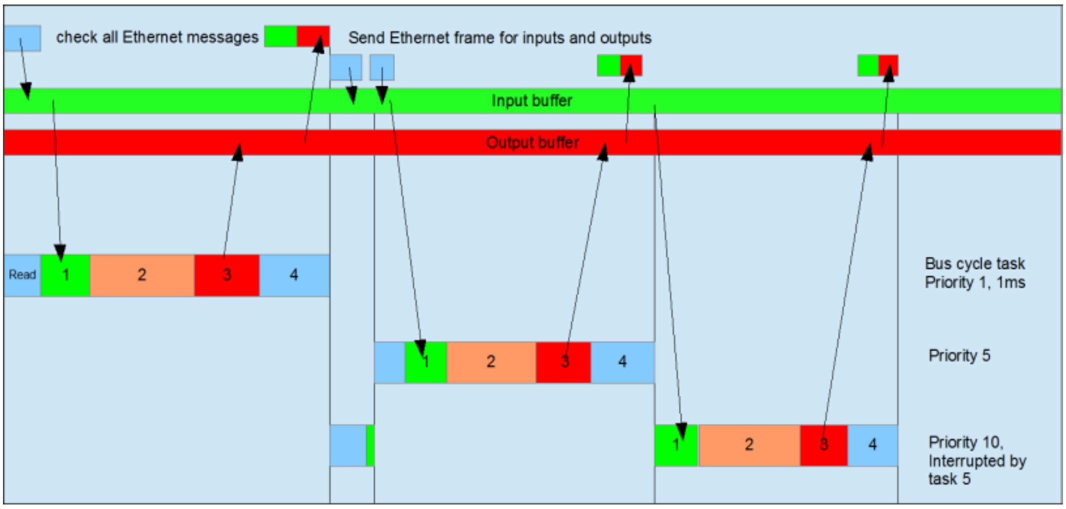


Figure 4: Task Processing with “Send/Receive per Task” Option

Note

Undefined States due to Writing and Reading in Several Tasks

If an output is written in several tasks, because it can be overwritten, the state is undefined.

If the same inputs are used in different tasks, sometimes the input changes while a task is processed. This happens when the task is interrupted by a higher-priority task, so the process image is read in again.

Remedy:

At the start of the IEC task, copy the input variables into the variables; then, in further code, only work with local variables.

Summary:

Using the same inputs and outputs for several tasks is not productive and in some cases, can cause unforeseen states.

3.2 EtherCAT-Specific Variables

When an EtherCAT Master is configured, an EtherCAT Master task is added to the task configuration of the current application. As usual, a call for a "Programmable Object Unit" (POU) can be added in the configuration of this task. EtherCAT-specific Boolean variables can be set in this POU to affect the action of the EtherCAT configuration within the application:

- Support for local devices

  When the fieldbus is started, the lack of EtherCAT devices configured in the application generates an error that prevents the stack from loading. After the variable

  ```
  <instance name of EtherCAT
  master>.StartConfigWithLessDevice := TRUE;
  ```

  is set at the start of the first PLC cycle, missing devices are treated as optional devices and will no longer prevent the bus from starting correctly.

- Suppressing transmission of additional messages

  The EtherCAT Master sends a message of its own for each individual task, to update the outputs as quickly as possible. If, for example, an attached drive is synchronized with the real-time output data, the bus cycle task should be the only task permitted to set the outputs. Additional messages would disrupt the synchronization. To suppress additional task messages, set

  ```
  <instance name of EtherCAT
  master>.EnableTaskOutputMessage := FALSE;
  ```

  once in the first PLC cycle. Alternatively, the "Send/Receive per Task" option can be set in the "Master" tab of the EtherCAT Master.

3.3 Libraries

3.3.1 Master

A type IoDrvEtherCAT instance is generated for each EtherCAT Master added to the device tree.

Table 4: Inputs (Master)

Name:	Data Type	Description
xRestart	BOOL	Rising edge: The master is restarted and all parameters are reloaded.
xStopBus	BOOL	TRUE: Communication is stopped. No further EtherCAT telegrams are sent. Then, because they were switched to error status, most devices must be restarted.

Table 5: Outputs (Master)

Name:	Data Type	Description
xConfigFinished	BOOL	TRUE: All configuration parameters were successfully transmitted. Communication is running on the bus.
xDistributedClockInSync	BOOL	When distributed clocks are used, the PLC is synchronized with the first EtherCAT slave for which the distributed clock option is activated. The output changes to TRUE as soon as the synchronization successfully ends. This signal can be used, for example, to ensure the SoftMotion function block is only started when the PLC is in synchronization mode, which will prevent potential position jumps. When the PLC starts, the output is FALSE and changes to TRUE after a few seconds. If the synchronization is lost for any reason, the output is reset to FALSE.
xError	BOOL	Output is TRUE when: • An error occurs when the EtherCAT stack starts. • Communication with the slaves is interrupted because no further messages can be received (e.g., due to a broken cable).
xSyncInWindow	BOOL	Output is TRUE when the option “Sync Window Monitoring” is activated and the synchronization of all slaves is within the synch window.

Examples

- Permit master restart via the “xRestart” variable:

```
EtherCAT_Master.xRestart := xRestart;
```

- Stop communication to the bus via the “xStop” variable:

```
EtherCAT_Master.xStopBus := xStop;
```

- Call the master to retrieve information regarding the success of the configuration parameter download:

```
EtherCAT_Master();
xFinish := EtherCAT_Master.xConfigFinished;
```

Table 6: Properties of the Master

Parameters	**Description**
AutoSetOperational	TRUE: Master tries to restart the slaves immediately after a communication interruption. Default: FALSE
ConfigRead	TRUE: Reading the configuration has ended and the user can edit the settings, e.g., to add self-defined SDOs.
DCInSyncWindow	Time frame for “XdistributedClockInSync” Jitter must remain within this frame so the output signal “XDistributedClockInSync” always stays at TRUE. Default: 50 microseconds
DCClockReferenceTime	Returns the current time of the first slave with activated distributed clock. This time serves as reference time for all other slaves and the PLC itself.
DCIntegralDivider	Integral factor for the distributed clock control loop; Default: 20
DCPropFactor	Proportional factor for the distributed clock control loop; Default: 25
DCSyncToMaster	Distributed-clock synchronization on the master when set to TRUE; then all slaves are synchronized to the master instead of synchronization of the first slave with the PLC. Default: FALSE
DCSyncToMasterWithSysTime	Distributed-clock synchronization on the master when set to TRUE; then all slaves are set to the system time on the master. The time read from “SysTimeRtcHighResGet” is used to distribute the PLC system time to all the EtherCAT slaves. Default: FALSE

Table 6: Properties of the Master

Parameters	Description
EnableTaskOutputMessage	Normally, EtherCAT messages are sent from the bus task and from each task that uses slave outputs. In the bus task, all outputs are written and all inputs read. In other tasks, the outputs are transmitted once more, so they immediately arrive to be written in the corresponding slaves. This is an attempt to keep dead time until writing as short as possible. In conjunction with distributed clocks, this may cause problems in some devices; for instance, servo controllers are not synchronized with “synch interrupt”; instead, they use the time of the writing for internal synchronization. In this case, multiple writing accesses may occur during a cycle. When “EnableTaskOutputMessage” is set to FALSE, only the bus cycle task is used. No other tasks will cause additional messages. Default: TRUE
FirstSlave	Pointer to the first slave below the master
FrameAtTaskStart	TRUE: To minimize jitter, the frame for the slaves is transmitted to start the task (before the IEC task). This command is used to ensure servo drive movements are smooth and not jerky. When this flag is set to TRUE, the frame of the output buffer is written in the next cycle – see diagram below. Default: FALSE
LastInstance	Pointer to linked master list -> preceding master
LastMessage	This property returns a string with the last message from the EtherCAT stack. “All slaves done” is returned if the start is successful. The string used is the same as the one for the diagnostics message displayed in the EtherCAT Master Device Editor window during online mode.
NextInstance	Pointer to linked master list -> next master
NumberActiveSlaves	This property returns the number of slaves that are actually connected. StartConfigWithLessDevice := TRUE: The number of real devices can be returned.
OpenTimeout	Timeout for opening the network adapter Default: 4 seconds

Table 6: Properties of the Master

Parameters	Description
StartConfigWithLessDevice	Used to affect the starting behavior of the stack. Normally, EtherCAT stack stops when, for example, five servo controllers are configured in a project but only three are connected. However, if StartConfigWithLessDevice := TRUE in the first cycle, the stack will try to start anyhow. This makes it possible to, for instance, arrange a universal configuration of ten servo controllers but still be able to vary the number that are actually connected. Note that in any case, the manufacturer and product IDs of each slave are checked. If any discrepancies are determined, the stack is stopped.

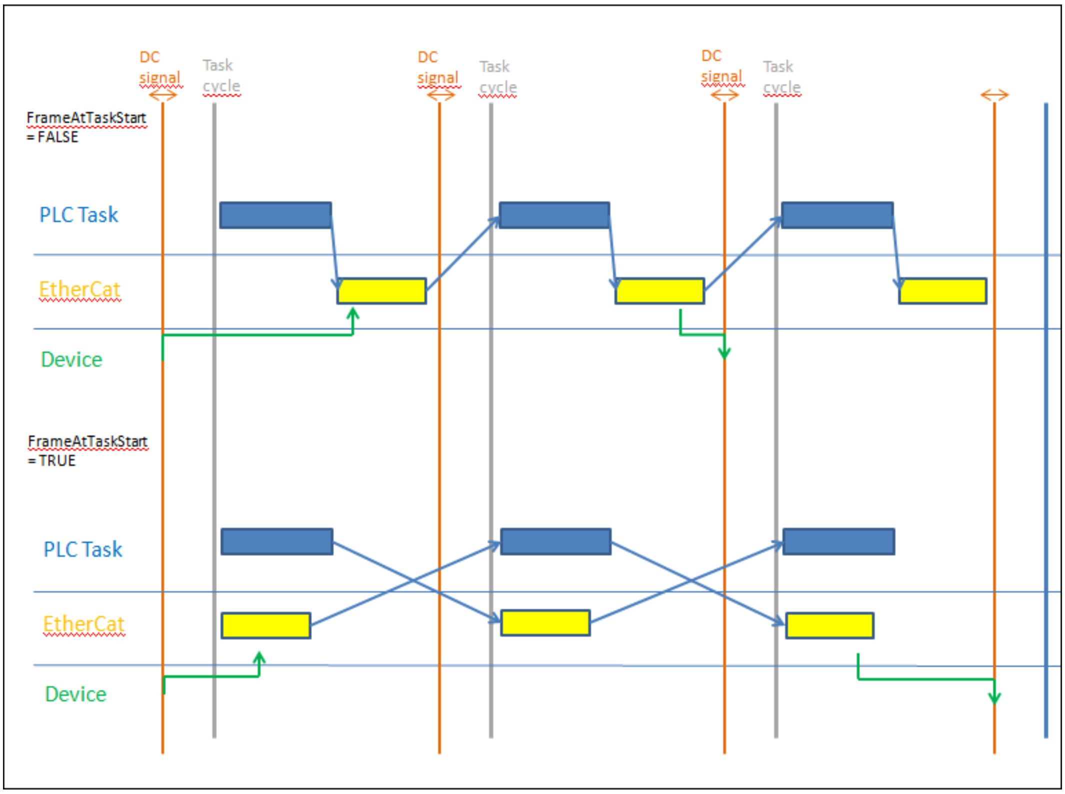


Figure 5: Effect of “FrameAtTaskStart”

3.3.2 Slave

A data type ETCSlave instance is generated for each EtherCAT slave added to the project. The instance name corresponds to the name of the slave in the project. The slave instance is used in the application to poll and/or switch the slave status.

Table 7: Inputs (Slave)

Name:	Data Type	Description
xSetOperational	BOOL	Rising flank: An attempt is made to switch to ETC_SLAVE_ OPERATIONAL mode.

Table 8: Outputs (Slave)

Name:	Data Type	Description
wState	ETC_SLAVE_ STATE	Current status of the slave: • 0: ETC_SLAVE_BOOT • 1: ETC_SLAVE_Init • 2: ETC_SLAVE_PREOPERATIONAL • 4: ETC_SLAVE_SAVEOPERATIONAL • 8: ETC_SLAVE_OPERATIONAL Configuration was successfully completed. If errors occur during the configuration, the slave can revert to a prior status.

Table 9: Properties of the Slave

Parameters	Description
VendorID	After the EtherCAT stack is started, this property returns the manufacturer ID read from the device.
ConfigVendorID	Reads the manufacturer ID from the configuration.
ProductID	After the EtherCAT stack is started, this property returns the project ID read from the device.
ConfigProductID	Reads the product ID from the configuration.
SerialID	After the EtherCAT stack is started, this property contains the device serial number.
LastEmergency	When a message is received, this information is saved in the slave and can be requested by the application via this property. A log report is also added.

If the check of the manufacturer or product IDs is activated in the expert settings, stack startup is stopped as soon as any discrepancies between VendorID and ConfigVendorID or between ProductID and ConfigProductID are determined.

3.3.2.1 Linked List to Check All Slaves

To monitor individual slaves, the instance is called up in the program and the status is determined through wState. To simplify this, all masters and slaves can be determined through linked lists and all slaves can be checked with a simple WHILE loop. The properties NextInstance and LastInstance exist for the masters as well as the slaves. These properties return a pointer to the next or previous slave. There is also the property FirstSlave for the master, which provides a pointer to the first slave The example below outlines how all slaves can be checked.

Example:

Declaration:

```
pSlave: POINTER TO ETCSlave;
```

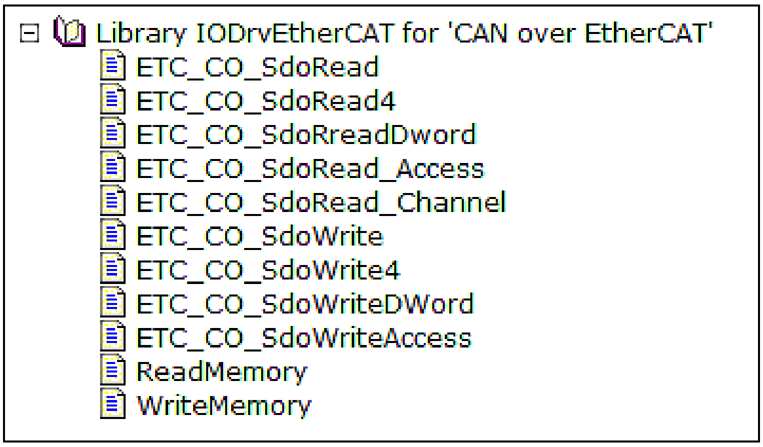
Program:

```
pSlave := EtherCAT_Master.FirstSlave;
WHILE pSlave <> 0 DO
pSlave^();
IF pSlave^.wState = ETC_SLAVE_STATE.ETC_SLAVE_OPERATIONAL
THEN
;
END_IF
pSlave := pSlave^.NextInstance;
END_WHILE
```

To start, the first slave on the master is called via “EtherCAT_Master.FirstSlave.” The respective instance is called up during the WHILE loop, thus determining wState. Then the status can be checked. The pointer on the next slave is called with “pSlave^.NextInstance.” At the end of the list, the pointer is NULL and the loop is exited.

3.4 Reading and Writing Device Parameters

The library “IODrvEtherCAT” is automatically integrated into a project when an EtherCAT configuration is supported and has been set. It receives function blocks to read and write parameters using the protocol “CAN over EtherCAT.” This allows the individual parameters to be checked and changed even during runtime. Several function blocks can be active at the same time. The individual requests are internally managed in a loop and processed sequentially.



⊟ Library IODrvEtherCAT for 'CAN over EtherCAT'
- ETC_CO_SdoRead
- ETC_CO_SdoRead4
- ETC_CO_SdoRreadDword
- ETC_CO_SdoRead_Access
- ETC_CO_SdoRead_Channel
- ETC_CO_SdoWrite
- ETC_CO_SdoWrite4
- ETC_CO_SdoWriteDWord
- ETC_CO_SdoWriteAccess
- ReadMemory
- WriteMemory

Figure 6: “CAN over EtherCAT” Library Functions

Information

Description of the Library Functions

Tables with complete details of the library functions are available in the ***e!****COCKPIT* Online Help.

4 Performance

The cycle time that can be reached with an EtherCAT Master depends on several significant environmental conditions, which may differ widely in different projects. This section provides information to guide project planning and on how to identify whether the device has the capacity to perform the required functionalities.

These factors are decisive:

- Which controller hardware is the software used on?
- How large is the bus's total process image?
  - Depending on the devices and in some cases, their modular structure:
- What is the maximum cycle time that can be used to perform the control processes needed?
- Which task priority can be designated to the EtherCAT task?
- How heavy is the load other tasks place on the controller?

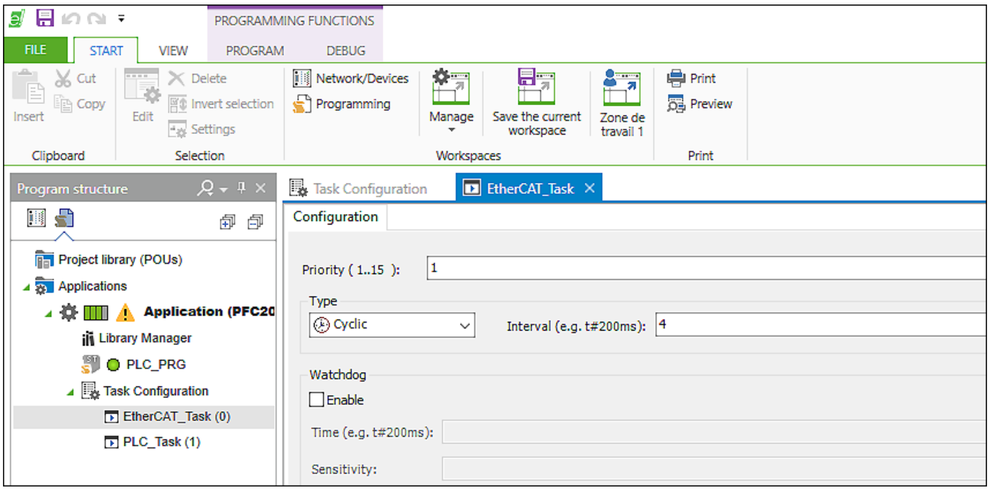
Depending on the answers to these questions, there may be other performance restrictions to this function. The following procedure is recommended for the startup:

Assumptions:

- A PFC200 (second generation) is used as controller.
- The process image does not exceed 200 bytes (e.g., any number of 750-354 fieldbus couplers with a total of 25 eight-channel digital modules).
- The process permits a maximum cycle time setting of five milliseconds.

Procedure:

- Plan the I/O node modular structure as described in the ***e!****COCKPIT* manual.
- Write the PLC program, visual application, etc.
- Set the EtherCAT task cycle time to five milliseconds.
- Set the EtherCAT task priority to 6.



Priority (1..15): 1

Type: Cyclic   Interval (e.g. t#200ms): 4

Watchdog

☐ Enable

Time (e.g. t#200ms):

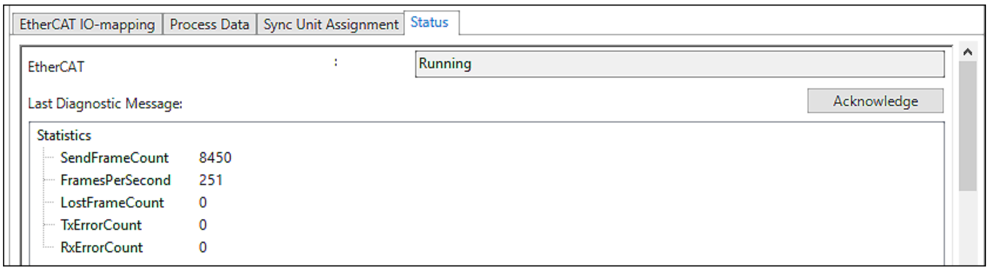
Sensitivity:

Figure 7: ***e!**COCKPIT* Settings under Program Structure > Application > Task Configuration

- Download the application and start the system.

The system will now attempt to perform the required tasks. The success of this attempt can be determined with the ***e!**COCKPIT* online functions:

- Go online with ***e!**COCKPIT*.
- Open the EtherCAT fieldbus configurator “Status” tab.
- In this dialog, check whether:
  - the required cycle time is reached (cycle time = 1 / FramesPerSecond [s]; in this example 1 / 201 s = 47.6 ms);
  - any telegram losses or errors are displayed in the statistics.



EtherCAT : Running

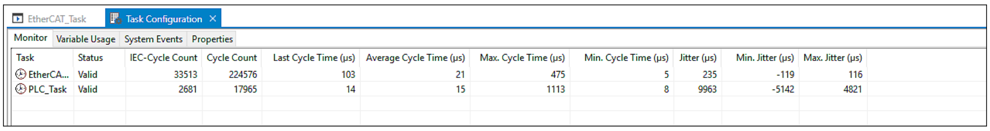
Last Diagnostic Message:

Statistics

SendFrameCount	8450
FramesPerSecond	251
LostFrameCount	0
TxErrorCount	0
RxErrorCount	0

Figure 8: EtherCAT Fieldbus Configurator – “Status” Tab

- Check whether the cycle times are maintained on average, or if the jitter information changes frequently.
  - Jitter (µs): Last measured jitter – Jitter is the difference between the time the task should start and the time it actually does. Example: If a task should start every 1000 µs and it only starts again after 1100 µs, the jitter is 100 µs.
  - Min. jitter (µs): Minimum measured jitter
  - Max. jitter (µs): Maximum measured jitter



Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)	Min. Cycle Time (µs)	Jitter (µs)	Min. Jitter (µs)	Max. Jitter (µs)
EtherCA...	Valid	33513	224576	103	21	475	5	235	-119	116
PLC_Task	Valid	2681	17965	14	15	1113	8	9963	-5142	4821

Figure 9: ***e!****COCKPIT* Settings under Program Structure > Application > Task Configuration

If there are no telegram losses or errors and the jitter is not too high, the CPU load is within the required range. The system can be operated with these settings. If necessary, the cycle time can be restricted and the process checked again. However, if errors do occur during the check, try to change the settings described above to improve this state.

If there are telegram losses or errors or if the jitter is too high, even though the CPU load is within the required range (~80 %), this can be improved by setting a higher task priority (= lower number). If this does not succeed, the only remaining option is to restrict the cycle time. Or, decide whether the application can accepted even with sporadic telegram loss.

The cycle time for a PFC200 (second generation) should not be set to less than two milliseconds.

5 Licensing

A license from WAGO is required to use the ***e!****RUNTIME* EtherCAT Master function. The license is acquired in ***e!****COCKPIT*, assigned to a controller and loaded into the controller together with the project. No other installation steps are required.

Table 10: Information about Licensing and Compatible Controllers

Item Description	**Item Number**
e!*RUNTIME* EtherCAT Master 300 Single license	2759-0263/0210-1000
Compatible Controllers	
PFC200; G2; 2ETH RS	750-8212
PFC200; G2; 2ETH CAN	750-8213
PFC200; G2; 2ETH RS CAN	750-8214
PFC200; G2; 4ETH CAN USB	750-8215
PFC200; G2; 2ETH RS CAN DPS	750-8216

In addition to the basic controller variants in the table, the license can also be used on all variants of these controllers.

Table 11: Information about Licensing and Compatible Touch Panels

Item Description	**Item Number**
e!*RUNTIME*-EtherCAT-Master 600 Single license	2759-0266/0210-1000
Compatible Touch Panels	
TP600 4.3 480x272 PIO2 VP	0762-4201/8000-0001*)
TP600 5.7 640x480 PIO2 VP	0762-4202/8000-0001*)
TP600 7.0 800x480 PIO2 VP	0762-4203/8000-0001*)
TP600 10.1 1280x800 PIO2 VP	0762-4204/8000-0001*)
TP600 4.3 480x272 PIO3 CP	0762-4301/8000-0002
TP600 5.7 640x480 PIO3 CP	0762-4302/8000-0002
TP600 7.0 800x480 PIO3 CP	0762-4303/8000-0002
TP600 10.1 1280x800 PIO3 CP	0762-4304/8000-0002
TP600 7.0 800x480 PIO2 VP	0762-5203/8000-0001*)
TP600 10.1 1280x800 PIO2 VP	0762-5204/8000-0001*)
TP600 7.0 800x480 PIO3 CP	0762-5303/8000-0002
TP600 10.1 1280x800 PIO3 CP	0762-5304/8000-0002
TP600 4.3 480x272 PIO2 VP	0762-6201/8000-0001*)
TP600 5.7 640x480 PIO2 VP	0762-6202/8000-0001*)
TP600 7.0 800x480 PIO2 VP	0762-6203/8000-0001*)
TP600 10.1 1280x800 PIO2 VP	0762-6204/8000-0001*)
*) For these devices an additional license „***e!****RUNTIME* PLC 600“ is necessary!	

Minimum ***e!****COCKPIT* version	V1.5.1
Delivery type	License certificate via email (***e!****COCKPIT* already contains the software itself)
Data sheet and further information	www.wago.com/2759-0263/0210-1000 www.wago.com/2759-0266/0210-1000

An Internet connection to the PC with the ***e!****COCKPIT* installation may be required for license activation.

The single license allows installation on one controller. One license per controller is required.

List of Figures

List of Tables

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D - 32385 Minden
Hansastraße 27 • D - 32423 Minden
Phone: +49 571 887 – 0
Fax: +49 571 887 – 844169
E-Mail: info@wago.com
Internet: www.wago.com