INCA-MCE EtherCAT Implementation Description

Document Revision: 1.0
Status: Released

Contents

1 Introduction 5

2 Basic Principles - Calibration and Measurement via EtherCAT 6

2.1 Basic data structure for communication 6
2.2 Unidirectional and time-triggered measurement and calibration 6
2.3 Static Configuration 6
2.4 Distributed Clocks 6
2.5 State Machine 7
2.6 Startup 7

3 INCA-MCE EtherCAT Slave 9

3.1 EtherCAT Slave Controller, Slave Information Interface 9
3.2 EtherCAT Slave Information file (Device description file) 9
3.3 Object Dictionary 9
3.3.1 Measurement - Input Objects & TxPDO's 9
3.3.2 Calibration State Variable – Input Objects & TxPDO's 9
3.3.3 Indexing and order of input objects & TxPDO's 9
3.3.4 Calibration – Output Objects & RxPDO's 10
3.4 Calibration State Variable 10

4 Legacy mode (differences to default mode) 12

4.1 Activation of Legacy Mode 12
4.2 EtherCAT Slave Information file (Device description file) 12
4.3 Object Dictionary 12
4.3.1 Measurement - Input Objects & TxPDO's 12
4.3.2 Calibration State Variable – Input Objects & TxPDO's 12
4.3.3 Indexing and order of input objects & TxPDO's 12
4.3.4 Calibration – Output Objects & RxPDO's 12
4.4 Calibration State Variable – Busy Flag 12

5 ETAS Contact Addresses 13

Figures

Figure 1: EtherCAT Slave Structure 5
Figure 2: EtherCAT Slave State Machine 7
Figure 3: Increment value of CSV by failed calibration 10

1 Introduction

This document provides a description of the EtherCAT relevant implementation details of INCA-MCE. The focus is on conventions between the EtherCAT master and the INCA-MCE EtherCAT slave as well as those details of the implementation that are relevant and visible for the EtherCAT master.

Most of the functionality is on the application layer, but some necessary details of the EtherCAT Slave Controller (ESC) as well as the EtherCAT Slave Information (ESI) file are described in this document. (The ESI file is the input to the EtherCAT Configuration Tool.)

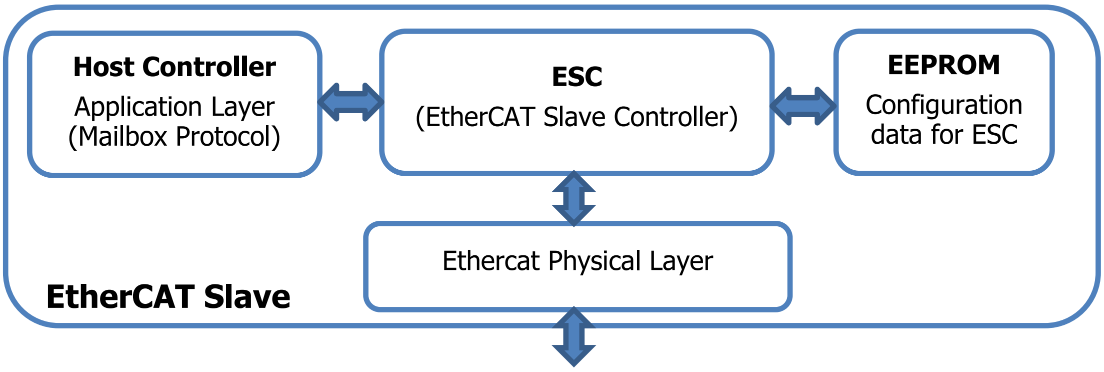

Figure 1: EtherCAT Slave Structure

Details of the EtherCAT specification will not be described here. A basic knowledge of EtherCAT is presumed. Further details about EtherCAT and the EtherCAT specification can be obtained from the EtherCAT Technology Group www.ethercat.org.

An initial version of the EtherCAT implementation was created and in use before the release of INCA-MCE V1.0. To support the transition to INCA MCE V1.x, a legacy mode is available in INCA-MCE. The introductory chapters of this document are generic and apply to both. The slave description is for the default mode; a final chapter describes the differences if legacy mode is used. The default (non-legacy) mode is recommended for all new implementations.

2 Basic Principles - Calibration and Measurement via EtherCAT

INCA-MCE allows the EtherCAT master (or other EtherCAT slaves, if so configured by the EtherCAT Master; for simplicity, this document will always just refer to the EtherCAT Master) to access calibrations and measurements in an ECU via an INCA-MCE EtherCAT slave.

2.1 Basic data structure for communication

Every measurement and calibration is defined within an input object or output object respectively. The data addressing is defined with TxPDO’s (Transmission Process Data Object) or RxPDO’s (Receive Process Data Object), respectively.

All measurement and calibration data are sent in the EtherCAT bus cycle frequency. The transmitted values are cast to a float32 value. The values are sent in physical representation.

Measurement arrays must be split into the individual entries. Every entry is then cast to a float32.

Multidimensional calibration structures (Maps, Curves, etc.) are addressed by a single value. All values in the 1D or 2D structure are then set to the same value. It is not possible to address single entries in the vector or matrix.

Additionally, an input signal (in an input object and TxPDO) is defined to return information to the master about the calibration status.

The object dictionary is available according to the CANOpen standard and can be downloaded by the EtherCAT master via the EtherCAT mailbox.

2.2 Unidirectional and time-triggered measurement and calibration

Obviously, the measurements are unidirectional (measurements are sent from the INCA-MCE HW device to the EtherCAT master). Calibrations are also unidirectional. The EtherCAT master can request a change to an ECU parameter, but the EtherCAT master is not able to read the current ECU parameters.

In typical modern automotive ECU protocols (e.g. XCP), the measurement and calibration is event-triggered. For measurement variables, they are configured and requested from the ECU. When the measurement is started, the data are streamed to the client; whenever the data are available in the defined SW raster, they are sent to the client. For calibration variables, calibration values are only sent if the current value is to be changed.

By contrast, calibration and measurement in INCA-MCE via EtherCAT are done using a bus frequency defined by the EtherCAT master. The corresponding period is not necessarily the same as the ECU raster for the measurements; it is also not synchronized with it. All configured calibration values are also sent in this frequency; this is independent of whether the values change.

For further details of the allowed measurement and calibration variable types, etc., please see the user manual for INCA-MCE.

2.3 Static Configuration

The object dictionary has entries for the configured ECU variables. The object dictionary is created when INCA goes online (calibration access activated, e.g. MCD-3MC V2.2 “ConnectToModule”) and / or when a measurement in INCA is started (e.g. MCD-3MC V2.2 Start Collector).

If the configuration in INCA is changed (new variables are added or removed) and a new object dictionary is created, the INCA-MCE EtherCAT slave will take on the state “Pre-OP”. The new object dictionary will be downloaded to the INCA-MCE EtherCAT slave and the old object dictionary will be erased. An object dictionary is never modified; a new dictionary is always created.

For details of how to change and activate the configuration manually or via an automation interface of INCA, see the INCA-MCE manual.

2.4 Distributed Clocks

EtherCAT offers a synchronization mechanism based on IEEE 1588. INCA can be configured to use the EtherCAT distribute clock time as the time-base for saving measurements. The EtherCAT master must follow the parameter setting in the ESI file for the cycle time of SYNC0. The specified fixed value must be used. (See ETG 2000 specification, for details).

The synchronization within INCA is done globally. The EtherCAT distributed clock time will be used to generate a single offset time that is applied uniformly to all measurement data points in the recording.

2.5 State Machine

EtherCAT has 4 states Init, Pre-Operational, Safe-Operational & Operational.

Init is the default initial state of the EtherCAT slave.

In Pre-Op, the object dictionary of the INCA-MCE HW device can be read by the EtherCAT Master. This state is possible as soon as the INCA-MCE EtherCAT slave has fully booted.

Distributed clocks are not available in the states Pre-Op & Init. Therefore, a transition from other states back to Pre-Op or Init should be avoided during running measurements, since this could create conflicts (due to e.g. inconsistent time stamps).

The INCA-MCE EtherCAT slave only allows a transition from Pre-Op to Safe-Op (and then to Op) if a valid measurement and/or calibration configuration is available. Otherwise, the INCA-MCE EtherCAT slave will return error 0x0018 or error 0x0019 in the AL status code and stays in the state Pre-Op.

In the states Op or Safe-Op, a change to the measurement or calibration configuration could be triggered by INCA. In this case, the INCA-MCE EtherCAT slave switches to state Pre-Op and sets the AL status code to 0x0022. The old object dictionary will be erased and the new object dictionary will be loaded. Until the new object dictionary is available and can be read by the EtherCAT master, the INCA-MCE EtherCAT slave does not allow transitions to Safe-Op or Op.

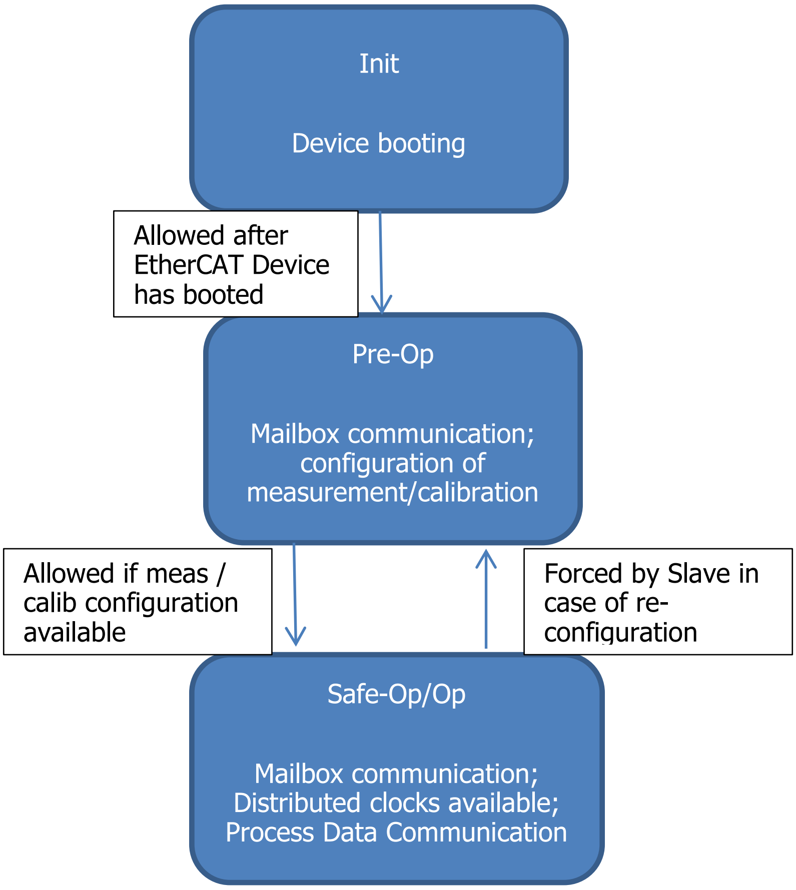

Figure 2: EtherCAT Slave State Machine

2.6 Startup

The calibration variables are sent cyclically to the INCA-MCE EtherCAT slave when the system is in the operational state. These values are not constantly forwarded to the ECU. If the value has not changed, there is no action for the ECU to carry out. In order to implement this filtering, the INCA-MCE EtherCAT slave must have a basis for comparison, when the calibration variables are read from the EtherCAT bus. In other words, it has to have a basis to compare the values to, in order to determine if a change has occurred or not. (See INCA-MCE manual for further details and consequences of this filtering).

The comparison basis is normally the last values received via the EtherCAT bus. To create the initial comparison basis, the EtherCAT master must send the current ECU values (e.g. obtained via an automation interface of INCA) in the first EtherCAT bus cycle. These are not sent to the ECU and are used as the basis for comparison for all further calibration parameter sets received via EtherCAT.

3 INCA-MCE EtherCAT Slave

The principle behavior and definition of the INCA-MCE EtherCAT slave are described in this chapter. The major sections deal with the Slave Information Interface (SII), ESI (or device description) file, the structure of the input and output objects as well as some details about the status variable for calibration.

3.1 EtherCAT Slave Controller, Slave Information Interface

The SII content in the INCA-MCE EtherCAT slave conforms to the EtherCAT specification.

The serial number of the INCA-MCE EtherCAT slave (hardware SN) is contained in the SII.

A station alias for the INCA-MCE EtherCAT slave can be set by the EtherCAT master. The station alias is stored persistently. If the SII is updated (e.g. with the Hardware Service Package (HSP) tool), the station alias is retained.

The EtherCAT Slave Controller is configured with 1 single port. (Current ETAS INCA-MCE EtherCAT slave hardware has only one Ethernet port available for EtherCAT).

3.2 EtherCAT Slave Information file (Device description file)

The EtherCAT Slave Information file conforms to the ETG.2000 specification V1.0.0.

The ESI file can be downloaded from the INCA-MCE EtherCAT Slave web server, see the corresponding hardware manual for details of accessing the web server (e.g., the ES910.3 manual).

3.3 Object Dictionary

The object dictionary conforms to the CANOpen standard.

3.3.1 Measurement - Input Objects & TxPDO's

There is one input object of type RECORD per ECU Device attached to the INCA-MCE EtherCAT slave for the measurement variables. The maximum number of input objects for measurement depends on the slave used (e.g. maximum of 5 for ES910.3).

The input object is named Measurement_<ECU-Name>. ECU-Name corresponds to the device name for the ECU in the INCA Hardware Configuration. The input object contains up to 254 measurement signals as 32 bit floats.

The corresponding TxPDO elements (one per measurement input object) are named TxPDO_Meas_<ECU-Name>. The TxPDO's are all assigned to Sync Manager 3.

If no measurements are configured for an ECU device (calibrations only), no input object and TxPDO is created for this ECU device.

The measurement signals are named <Signal-Name>. The Signal-Name corresponds to the name given in the ECU description (*.a2l file).

3.3.2 Calibration State Variable – Input Objects & TxPDO's

There is one input object of type RECORD per ECU Device attached to the INCA-MCE EtherCAT slave for the calibration state variable. The maximum number of input objects for the calibration state depends on the slave used (e.g. maximum of 5 for ES910.3).

The input object is named Cal_State_<ECU-Name>. The input object contains exactly one calibration state variable as a 16 bit unsigned integer.

The corresponding TxPDO elements are named TxPDO_Cal_State_<ECU-Name>. The TxPDO are all assigned to Sync Manager 3.

If no calibrations are defined for an ECU device, the input object and TxPDO for the calibration state variable will not be created for this ECU device.

The calibration state variable entry is named State_Variable.

3.3.3 Indexing and order of input objects & TxPDO's

The measurement input objects are numbered consecutively 0x6000, 0x6001, etc. The calibration state variable input objects are numbered consecutively with the next available index after the measurements (e.g. if 2 measurement input objects exist, the first calibration state variable input object has the index 0x6002). If no measurements are configured, the first calibration state variable input object has the index 0x6000.

3.3.4 Calibration – Output Objects & RxPDO's

There is one output object of type RECORD per ECU Device attached to the INCA-MCE EtherCAT slave for the calibration variables. The maximum number of output objects for calibration depends on the slave used (e.g. maximum of 5 for ES910.3).

The output object is named Calibration_<ECU-Name>. The output object contains up 254 calibration signals as 32 bit floats.

The corresponding RxPDO elements (one per calibration output object) are named RxPDO_Cal_<ECU-Name>. The RxPDO's are all assigned to Sync Manager 2.

If no calibrations are defined for an ECU device (measurement only), no output object and RxPDO is created for this ECU device.

The calibration signals are named <Signal-Name>. The Signal-Name corresponds to the name given in the ECU description (*.a2l).

3.4 Calibration State Variable

Calibration via EtherCAT is done in the bus cycle frequency. The calibration requests are not explicitly sent (only inferred by comparing the last value sent to the latest value), are not acknowledged and the master is theoretically able to send new requests even if the last request is still being processed.

The INCA-MCE EtherCAT slave sends all changes to calibration parameters to the ECU. If the communication with the ECU is with a serial protocol, these changes will be sent serially to the ECU; nevertheless, the calibration request is handled as a block in the INCA-MCE EtherCAT slave. All RxPDOs are assigned to a single sync manager; as a result, all (combined) calibration requests for all ECUs attached to a single INCA-MCE EtherCAT Slave must be completely processed before new requests can be accepted. Until the last block sent is completely processed, no further changes to the RxPDOs can be considered. When the block is processed (all values acknowledged from the ECUs), the values in the next RxPDOs received will be compared to the values sent with the last valid calibration request to determine if a new request is available.

The calibration state variable gives feedback about the current status of calibration. The calibration state variables allows the EtherCAT master to determine if

- The last calibration request has been completed, and if
- The last calibration request was successful or unsuccessful;

as well as the returning the error code of the first failed calibration, if the calibration was not successful.

There are separate calibration state variables for each output object / RxPDO. Their individual values are determined independently by the status of the corresponding ECU devices.

The calibration state variable comprises 16 bits.

The calibration state variable is incremented by the number of calibrated variables after successful execution of a (combined) calibration request.

The calibration state variable is incremented by the concatenation of (from most to least significant bit) if at least one calibration request was unsuccessful:

- 1
- the 7-bit error code belonging to the first failed calibration of a variable
- the number of calibrations from the (combined) calibration request that failed

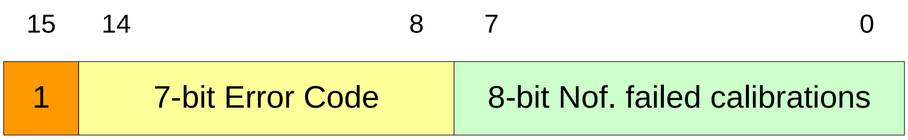

Figure 3: Increment value of CSV by failed calibration

Further explanation

If the CSV has a value of *x* and a calibration request is received, the value remains *x* until the calibration request is completed (successfully or unsuccessfully).

→ If a request is sent and the value of the CSV does not change, the calibration request is still ongoing. Further requests will not be processed.

If the CSV has a value of *x* and 8 calibration requests are sent simultaneously, the value will increase to *x+8* if and after the calibration was successfully completed.

→ When *x+8* is received, the master can deduce that the calibration was successful, confirm that the request contained 8 requests in total and know that the ECU is available for new calibration requests.

If the CSV has a value of *x* and 8 calibration requests are sent simultaneously and 1 or more calibration were not successful, the value of the CSV will change after the sequence is completed. Let us assume that

- *x* = 00000000 10011001
- 2 calibrations were not successful
- The error code of the 1st failed calibration is 0x24 = 0100100

Then the CSV will be 10100100 10011011.

→ When the new CSV value is received, the master can deduce that an error has occurred (leading bit of CSV has changed), the error code of the first failure was 0x24 (value in the 7 bits following the leading bit) and that in total 2 requests failed (the last 8 bits were incremented by 2).

4 Legacy mode (differences to default mode)

A compatibility mode exists in INCA-MCE to support the migration of implementations based on the INCA-MCE evaluation kit (AS solution) to the INCA-MCE default mode. All new implementations should be based on the default mode, since it is substantially less restrictive / more flexible. This chapter describes only the differences to the standard implementation.

4.1 Activation of Legacy Mode

The legacy mode can be activated using the web server of the INCA-MCE EtherCAT Slave. See the manual for details of accessing the web server (e.g. ES910.3 user's guide). This setting is persistent.

4.2 EtherCAT Slave Information file (Device description file)

There is a separate ESI file for the legacy mode. It is called "ETAS_EtherCAT_Devices_Legacy.xml".

This ESI file contains RxPDO and TxPDO elements inside the Device Type element. The RxPDO and TxPDO elements contain PDO Type elements.

The names of the entries in the PDO Type elements for the TxPDOs are measurement_<x>. There are exactly 200 elements, numbered 0 to 199.

The names of the entries in the PDO Type elements for the RxPDOs are parameter_<x>. There are exactly 50 elements, numbered 0 to 49.

4.3 Object Dictionary

4.3.1 Measurement - Input Objects & TxPDO's

There is one input object of type RECORD per INCA-MCE EtherCAT slave for the measurement variables.

The input object is named Measurement. The input object contains up to 200 measurement signals as 32 bit floats.

The corresponding TxPDO element is named "Ecu input Mapping". The TxPDO is assigned to Sync Manager 3.

The measurement signals are named <Signal-Name>/<ECU-Name>#FULI.

4.3.2 Calibration State Variable – Input Objects & TxPDO's

There is no calibration state variable. Instead of this, there is a float 32 entry reserved in the measurement object / TxPDO, called the busy flag. It has the possible values of 0.0 and 1.0. No separate input object / TxPDO exists. This measurement signal is named EcatBusyFlag.

4.3.3 Indexing and order of input objects & TxPDO's

The Input Object has the index 0x6000. The corresponding TxPDO has the index 0x1A00.

4.3.4 Calibration – Output Objects & RxPDO's

There is one output object of type RECORD per INCA-MCE EtherCAT slave for the calibration variables.

The output object is named Calibration. The output object contains up to 50 calibration signals as 32 bit floats.

The corresponding RxPDO element is named "Ecu output Mapping". The RxPDO is assigned to Sync Manager 3.

The calibration signals are named <Signal-Name>/<ECU-Name>#FULI.

4.4 Calibration State Variable – Busy Flag

The INCA-MCE EtherCAT Slave sets the busy flag to "1" in the TxPDO when any calibration is ongoing. This bit is global for all attached ECUs. The value is reset to "0" when the last (combined) calibration request is completed, regardless whether it was successful or not.

5 ETAS Contact Addresses

ETAS HQ

ETAS GmbH		
Borsigstraße 14	Phone:	+49 711 89661-0
70469 Stuttgart	Fax:	+49 711 89661-106
Germany	WWW:	www.etas.com

ETAS Subsidiaries and Technical Support

For details of your local sales office as well as your local technical support team and product hotlines, take a look at the ETAS website:

ETAS subsidiaries	WWW:	www.etas.com/en/contact.php
ETAS technical support	WWW:	www.etas.com/en/hotlines.php