

Electronic hardware identification V2

Technical memo

1 Decision

1.1 Subject

Electronic stored hardware identification data of PCBs (printed circuit boards).

This techmemo defines,

- Which hardware-identification-data has to be stored
- The way of storing that hardware-identification-data

1.2 Scope

This techmemo is valid:

- For PCBs (printed circuit boards) with development-date since 2013.
- All PCBs of the EtherCAT-platform (also PCBs with development-date earlier than 2013).

For PCBs with development-date earlier than 2013, the techmemo D00010849 is valid.

Exceptions:

Due to compatibility to older platforms, it is exceptionally necessary to use D00010849 also for new PCBs instead of applying this technical memo. (Example: ICP1601i004)

1.3 Storing hardware identification data

1.3.1 Physical memory

The memory-technology is not defined by this techmemo. But following rules must be regarded:

- Hardware-identification-data must be stored in non-volatile-memory.
- The location of hardware-identification-data should not be placed together with program-code and process-data (risk of overwrite).
("Together" has to be interpreted in the particular case as "memory-chip" or just as "flash-section".)
- EtherCAT-platform: hardware identification data must be located in the ESI (mandatory).
- Write protection on i2c-devices is not needed. (risk of overwrite is not that high).
(EtherCAT: Write protection of the ESI is not allowed.)

1.3.2 Interface

The interface to the hardware-identification-data is not defined by this techmemo.

For EtherCAT-Devices, the

1.4 Data format

Beside the needed hardware-information this techmemo also defines the format in which the hardware-information has to be stored. The data-format is based on the EtherCAT-Slave-Information (ESI) of an EtherCAT-Slave¹ and uses the vendor-specific category-types.

The main-advantage of defining the data-format is reuse of software-code.

¹ For example the I2C-EEPROM on a Beckhoff ET1100-Asic

1.4.1 Overview of hardware-information and their purpose and location in memory

Item	Mandatory / Optional	Memory location in non ECAT-Devices	Memory location in ECAT-Devices (ESI)	CANopen Idx.SubIdx ¹	Purpose and description
Vendor ID	Mandatory	Category type: HWInfo1 0x0811	SII-area: Word-address 0x0008	0x1018/0x01 (CanOpen DS301)	Protocol specifications of CANopen, EtherCAT Komax: 0x0057 (registered by CiA) remark: CiA has divided the Vendor ID in the section Bits 31-24 (Department-ID) and the section Bits 23-0 ("real" Vendor-ID).
ProductCode before 2013: Printart	Mandatory	Category type: HWInfo1 0x0811	SII-area: Word-address 0x000A	0x1018/0x02 (CanOpen DS301)	Definition of target-hardware ² for the firmware. Firmware-relevant bug-fixes needs a new ProductCode (old Firmware-Revisions won't run on this hardware). Getting a new ProductCode: https://wiki.komax.org/x/tgVpAQ
RevisionNo before 2013: Index	Mandatory	Category type: HWInfo1 0x0811	SII-area: Word-address 0x000C	0x1018/0x03 (CanOpen DS301) (0x1009/0x00)	Index of untested PCB. ³ Enhanced information for Product-Traceability. Handling of all HW-Revisions (also non-SW-relevant).
SerialNo	Mandatory	Category type: HWInfo1 0x0811	SII-area: Word-address 0x000E	0x1018/0x04 (CanOpen DS301)	Serial number of PCB Base-information for Product-Traceability.
ProductionYear	Mandatory	Category type: ProductionInfo 0x0813	Category type: ProductionInfo 0x0813	0x1018/0x06 (Komax extension)	Enhanced information for Product-Traceability.
ProductionLot	Mandatory	Category type: ProductionInfo 0x0813	Category type: ProductionInfo 0x0813	0x1018/0x05 (Komax extension)	Enhanced information for Product-Traceability.
PcbName	Mandatory	Category type: HWInfo2 0x0812 PcbNameIdx to String	Category type: General 0x001E NameIdx to Strings	0x1008/0x00 (CanOpen DS301)	Convenience-Purpose, Text-Translation of ProductCode / ArticleNo
ArticleNo	Mandatory	Category type: HWInfo2: 0x0812 ArticleNoIdx to Strings	Category type: General 0x001E OrderIdx to Strings	0x1018/0x0A (Komax extension, optional)	ArticleNo of untested PCB Base-information for Product-Traceability.
MacAddress	Optional	Category type: MacAddress 0x0814	Category type: MacAddress 0x0814	0x1110/0x00 (Beckhoff extension)	MAC-Address for Network-Applications Getting a new MAC-Address: https://wiki.komax.org/x/eAM
CrcHwInfoESI	Mandatory for ECAT		Category type: CrcHwInfoESI 0x0810		Only for EtherCAT-Devices. Check ProductCode for correct behavior of Firmware.

¹ EtherCAT:Preferred interface is reading the ESI. The listed CoE-messages (CAN over EtherCAT) are not mandatory. Some PCBs (e.g. IOe) do not support any CoE or not all CoE-HW-Infos (e.g. 0x1018/0x05 on FMAe). Some PCBs supports CoE-HW-Info only under certain circumstances (read permission ESI-device).

² Target-Hardware means the type of PCB (printed circuit board) and not the type of machine/module.

³ RevisionNo: Index of untested PCB. This does not comply to CANopen (DS301, Obj. 1018) (See also chapter 2.1.4)

1.4.2 Data format rules

- If possible, data has to be handled through the given communication-standard.
(example EtherCAT: VendorID, ProductCode, RevisionNo, SerialNo is stored in SII-area of ESI)
- All other data are stored in categories

Rules for creating categories:

- The data-format is word-orientated.
 - All categories have an even number of bytes.
 - It is possible to store data byte-wise, but a whole category has an even number of bytes.
- The data-format is Little Endian.
 - 16Bit: A number of 0x2B1A is stored as 0x1A; 0x2B
 - 32Bit: A number of 0x4D3C2B1A is stored as 0x1A; 0x2B; 0x3C; 0x4D
 - 0x1A @ byte-address 0xYY;
 - 0x2B @ byte-address 0xYY+1
 - 0x3C @ byte-address 0xYY+2
 - 0x4D @ byte-address 0xYY+3
- Each category has a header which defines the category-type and the category-length.
- Each category can only be used once. (No redefinitions of the same information.)
- All categories are placed in the memory next to each other.
- The order of the categories in the memory is random.
- The format within the category-data is defined by the category-type.
- The end of the category-section in the memory is marked by the category "END" (category-type 0xFFFF).
- Data behind the category "END" is not interpreted as categories.

The set of category-types can be extended with new Komax specific category-types in the future. New category-types must be registered and defined in Komax-wiki: <https://wiki.komax.org/x/Rws7Ag>

- Range of category-type-values: 0x0800 .. 0x0FFF --> 2048 possible vendor specific category types
- new category-types should be defined in a reasonable way
--> do not hesitate to define a new type but don't waste the finite number of category-types

1.4.3 General data structure

Parameter	Address offset [word]	Data type	Value / Description
Header first category	0x0000	Header	Header including category-type and size category data
Data first category	0x0002	Category dependent	Data structure is category-type specific
Header second category	0x0002 + x	Header	
Data second category	0x0004 + x	Category dependent	
....			
Header last category		Header	Last category is the category "End" with no data. Category-type = 0xFFFF ; Data length = 0

1.4.3.1 Header of category type

Parameter	Address offset [Word Address]	Data type	Value / Description
Category type	0x0000 (word)	Unsigned 16	Category Type as defined in Chapter 1.4.4
Length of category	0x0001 (word)	Unsigned 16	

1.4.3.2 Application in ECAT-Devices

In ECAT-Devices, the HW-identification-data are located in the ESI (I2C-EEPROM for EtherCAT-Slave-Information). The section with the categories starts at address 0x0040

Parameter		Address [word]	Value / Description
Slave-Information-Interface-area		0x0000 .. 0x003F	Only in EtherCAT-Devices Length 64 Words (128 Bytes)
Categories	Categories defined by ECAT-protocol	0x0040	Category Strings
			Category General
			Category FMMU
			Category SyncManager
			Category Tx-/RxPDO for each PDO
		
	Komax specific categories (vendor specific)		Category CrcHWInfoESI
			Category HwInfo1
			Category ProductionInfo
			Category MacAdress
		
	Last category		Last Category is the category "End" with no Data Category-Type = 0xFFFF ; Data length = 0
Vendor specific area			Only in EtherCAT-Devices

Preserve Online Data:

The attribute "PreserveOnlineData" in the Beckhoff-ESI-XML-file defines, whether the according category should be overwritten by the EtherCAT-master¹. Set this attribute for all categories with hw-identification-data to '1'.

Remark: The Attribute "Preserve Online Data" has no effects on Komax-machines with the EtherLab master (RT-Controller), because EtherLab does not use the ESI-XML-file.

1.4.4 Category Types

The binding list of all Komax specific category-types is documented in Wiki:

<https://wiki.komax.org/x/Rws7Aq>

In the following chapters are only the category-types specified, which are relevant for the hardware-identification.

1.4.4.1 General (Category type value = 0x001E)

Only for "ECAT-Devices".

The category General is relevant for the information:

- PcbName : Parameter **PcbNameldx** in Category HwInfo2 = **Nameldx** in Category General
- ArticleNo : Parameter **ArticleNoldx** in Category HwInfo2 = **Orderldx** in Category General

See detailed definition of the category General in specification of ETG².

¹ [ETG2000_S_R_V1i0i2_EtherCATSlaveInformationSpecification.pdf](#) Chapter 13.6.

² document ETG.1000.6 s (R) V1.0.2, Table 21.

1.4.4.2 Strings (Category type value = 0x000A)

The category Strings was defined by ETG¹.

All strings of the EEPROM are located in the category Strings. Besides the strings of hardware-identification like "PcbName" can also a lot of other strings be located in this category. For example are also the name-strings of EtherCAT-PDOs located in this category.

General rule: A String-Index in another category with the value n points to string str_n in the category Strings.

Parameter	Address offset [byte address]	Data type	Value / Description
nStrings	0x0000 (byte)	Unsigned8	Number of Strings
str1_len	0x0001 (byte)	Unsigned8	Length of String 1
Str_1	0x0002 (byte)	Byte[str1_len]	String 1 Data
str1_len	0x0002 + str1_len (byte)	Unsigned8	Length of String 2
Str_2	0x0003 + str1_len (byte)	Byte[str2_len]	String 2 Data
.....			
strn_len	0x000z (byte)	Unsigned8	Length of String n
str_n	0x000z+1 (byte)	Byte[strn_len]	String n Data
PAD_Byte	0x000y	Unsigned8	Padding if Category length is odd

1.4.4.3 CrcHwInfoEsi (Category type value = 0x0810)

Only used (and mandatory) for "ECAT-Devices".

VendorID, ProductCode, RevisionNo and SerialNo are stored in the ESI-SII-area and are not CRC-protected. Devices may control their Software-behavior according the mentioned information. To prevent wrong device-behavior in order of a read-error, VendorID, ProductCode, RevisionNo. and SerialNo should be CRC-protected with this category. The CRC is also useful for traceability-purposes in quality management.

Parameter	Address offset [word address]	Data type	Value / Description
CRC16	0x0000 (word)	Unsigned16	<p>CRC-CCITT Cyclic Redundancy Check over VendorID, ProductCode, RevisionNo, SerialNo (word 8 to word 16 of the ESI)</p> <p>CrcHwInfoESI is mandatory: Komax-ECAT-devices must store the Category CrcHwInfoESI.</p> <p>The analysis of the stored CRC is mandatory for PCB-Test-Devices for evaluation of warranty-claims. The analysis of the stored CRC is optional for devices and machines.</p> <p>The analysis of the stored CRC is strongly recommended for devices with HwInfo-depending Software-behavior.</p>

CRC-CCITT:

- 16bit CRC-CCITT
- CRC-Polynomial = $X^{16} + X^{12} + X^5 + 1$
- Initial-value: 0

¹ ETG: EtherCAT Technology Group, Definition in document ETG.1000.6 s (R) V1.0.2, Table 20.

1.4.4.4 HwInfo1 (Category type value = 0x0811)

HwInfo1: only used (and mandatory) for Komax-Non-ECAT-Devices.

Parameter	Address offset [word address]	Data type	Value / Description
VendorID	0x0000 (word)	Unsigned32	Komax: 0x0057 The number 0x57 is registered by CiA and is already used in Komax-CANopen-Devices. remark: CiA has divided the Vendor ID in the section Bits 31-24 (Department-ID) and the section Bits 23-0 ("real" Vendor-ID).
ProductCode	0x0002 (word)	Unsigned32	Definition of Target-Hardware for the Firmware. Firmware-Relevant bugfixes which won't run with old Firmware or Firmware-Revisions needs a new ProductCode (formerly Printart). Getting a new ProductCode: https://wiki.komax.org/x/tgVpAQ
RevisionNo	0x0004 (word)	Unsigned32	Index of untested PCB
SerialNo	0x0006 (word)	Unsigned32	Serial number of PCB
CRC16	0x0008 (word)	Unsigned16	CRC-CCITT Cyclic Redundancy Check over VendorID, ProductCode, RevisionNo, SerialNo (word 8 to word 16 of the ESI) The analysis of the stored CRC is mandatory for PCB-Test-Devices for evaluation of warranty-claims. The analysis of the stored CRC is optional for devices and machines. The analysis of the stored CRC is strongly recommended for devices with HwInfo-depending Software-behavior.

CRC-CCITT:

- 16bit CRC-CCITT
- CRC-Polynomial = $X^{16}+X^{12}+X^5+1$
- Initial-value: 0

1.4.4.5 HwInfo2 (Category type value = 0x0812)

Only for "Non-ECAT-Devices". ECAT-Devices hold this Information in the Category General (chapter 1.4.4.1).

Parameter	Address offset [byte address]	Data type	Value / Description
ArticleNoldx	0x0000 (byte)	Unsigned8	<p>Index to a string in the Category Strings which holds the ArticleNo of the untested PCB.</p> <p>Use just the needed characters. That means 7 characters for the 7 digits in the article-number. All zeros of the article-number have to be written. "0000000" .. "9999999"</p> <p>Remark: In the old data-structure of techmemo D00010849 always 10 characters must have been used.</p>
PcbNameldx	0x0001 (byte)	Unsigned8	<p>Index to a string in the Category Strings which holds the name of the PCB.</p> <p>Use just the needed characters. To stay compatible with all types of software, do not use more than 20 characters.</p> <p>Remark: In the old data-structure techmemo D00010849 always 20 characters must had been used.</p>

Remark:

The separation of HwInfo2 and HwInfo1 is first due to the CRC-Check. HwInfo2 contains String-Indexes, which may change during the Product-Life-Cycle although the according strings do not change. The second reason is to keep it as equal as possible with the non-ECAT-Devices.

1.4.4.6 ProductionInfo (Category type value = 0x0813)

Parameter	Address offset [word address]	Data type	Value / Description
ProductionYear	0x0000 (word)	Unsigned16	Production-Year with 4 digits (example 1900 0x076c)
ProductionLot	0x0001 (word)	Unsigned16	<p>Production-Lot in a Year. Lot is reset to "1" at each new year.</p> <p>example: 0x000b = eleventh lot of the year</p>

1.4.4.7 MacAddress (Category type value = 0x0814)

MAC-Address for Network-Applications (Media Access Control)

Getting a new MAC-Address: <https://wiki.komax.org/x/eAM>

Parameter	Address offset [byte address]	Data type	Value / Description
MAC6	0x0000 (byte)	Unsigned8	6. Byte of MAC-address (example 0x03)
MAC5	0x0001 (byte)	Unsigned8	5. Byte of MAC-address (example 0x01)
MAC4	0x0002 (byte)	Unsigned8	4. Byte of MAC-address (example 0x00)
MAC3	0x0003 (byte)	Unsigned8	3. Byte of MAC-address (example 0xFB)
MAC2	0x0004 (byte)	Unsigned8	2. Byte of MAC-address (example 0x0D)
MAC1	0x0005 (byte)	Unsigned8	1. Byte of MAC-address (example 0x00)

Example for MAC-Address 00-0D-FB-00-01-03 including category header and little-endian-format:

Byte-Address	Data
0x0000	14 08 03 00 03 01 00 FB
0x0008	0D 00 ?? ?? ?? ?? ?? ??

Hint: The byte-order is different to the old techmemo D00010849.

2 Context

2.1 Reasons

2.1.1 Why a new datastructure?

With the introduction of the EtherCAT-platform, Komax needed an adequate and flexible way of storing hardware identification data. The existing techmemo D00010849 did not fit the actual requirements of the EtherCAT-platform.

- The existing techmemo did not care about already stored data in the ESI.
 - It would not be easy, to synchronize redundant data.
 - Matching of "Printart" and "ProductCode" has to be defined.
- No additional i2c memory on EtherCAT-PCBs because it is possible to store all data within the ESI.
- The concept with the vendor-specific categories
 - Is orientated to a standard (EtherCAT)
 - Is flexible for new information.
 - Only stores needed information. Unused information can be dropped.

2.1.2 Why electronically stored hardware identification data?

- Before a Software-Download to a PCB starts, the Software-Downloader can check automatically whether the target is valid or not. So invalid software won't be downloaded to a PCB and prevents the PCB from destruction.
- Software-readable hardware-identification can be delivered to the customer-service by a machine-backup-file. So the customer-service gets a clear view of the state of a machine without doubts of correctness of the information.
- The electronically stored serial number helps to recognize PCBs which are in the repair-shop twice.
- The electronically stored serial number and production info helps in decisions about warranty-claims.
- The electronically stored serial number can't be faked as easy as the serial-number-sticker for warranty-claims (an example is swapping of id-stickers).

2.1.3 Why is ProductCode related to a PCB and not to a machine/module

2.1.3.1 Why should ProductCode relate to a machine module?

- Komax uses already a product code for machine modules. This product code is written on the type plate of the machine module.
- The type plate is readable without opening the machine module. For getting the identification of a PCB (printed circuit board), the customer has to open the machine module.
- TwinCAT uses this ProductCode and will assign a name to the device.
- Every EtherCAT-Slave has implemented the function for reading out the product code.

2.1.3.2 Why should ProductCode relate to the PCB?

A PCB can be used in different types of machine modules. Example IP2000.400 is used in mci712, mci721, mci722.

A machine module can hold more than one PCB of one or more types of PCBs. Example: Kappa 330 has a ICP1101, ICP1601 and a ICP1265.

- The Information on the type plate is not accessible by software.
- If the stored hardware identification relates to the machine/module,
 - the logistic cost would rise for replacing a PCB by the customer. Before sending the PCB to the customer, Komax would have to reprogram the hw-id to the specific machine/module.
 - swapping PCBs between machines/modules for diagnostic reasons could bring some problems.

2.1.4 Why does RevisionNo (CANopen object 0x1018.3) not comply to CANopen)?

2.1.4.1 Situation in CANopen standard DS301

- Obj. 0x1018.3 Revision number:
 - Bit 31..16 : major revision number: specific CANopen behaviour
 - Bit 15..00 : minor revision number: different versions with same CANopen behaviour
 - On EtherCAT-devices, this Revision number is stored in the EtherCAT-I2C-Eeprom in the SII-area on address 0x000c. (ETG.1000.6 S (R) V1.0.2, Table 16)
- Obj. 0x1009: Manufacturer Hardware Version
- Obj. 0x100A: Manufacturer Software Version

2.1.4.2 Reason

The index of the untested PCB is stored in the EtherCAT-I2C-Eeprom under the revision number (address 0x000c) because:

- No update of ESI when Firmware changes (mostly also change of CANopen behavior)
- CANopen behavior is entirely defined by the software version in CANopen object 0x100A
- Standard tools will read the revision number of the identity object (0x1018/0x03)

2.1.4.3 Implicit decision: usage of CANopen object 0x1009: manufacturer hardware version

In Komax-devices, the CANopen object 0x1009 (manufacturer hardware version) should be automatically generated from the object 0x1018/0x03 (revision number). The unsigned32 revision number should be converted to a string with a length of 3 characters. Leading zeros should be showed.

Example for index of untested PCB = 003

0x1018/0x03 = (unsigned32) 3

0x1009 = (visible string) "003"

2.1.5 ArticleNo versus PartNo versus OrderNo

	pro	Contra
Article No	<ul style="list-style-type: none"> Official term in Komax-glossary, therefore every official document like drawings use the term article number. 	
Part No	<ul style="list-style-type: none"> Term may be more common than article number (especially by suppliers of electronic components) Is used in the Komax I2C-Master 	<ul style="list-style-type: none"> Is not official in Komax Is not in common use in Komax
Order No	<ul style="list-style-type: none"> Term is used in the Category General of the EtherCAT-Slave-Information (ESI) Information is available on the fieldbus EtherCAT also on PCBs which has no microprocessor. Example: IOec (0314230). 	<ul style="list-style-type: none"> Is not official in Komax Is not in common use in Komax 2nd usage in following context : Order No > Delivery No > Invoice No The term "Order No" is a specific definition by the ETG¹. It may be redefined in a new version of the EtherCAT-Specification. The term may not be used in a future fieldbus.

Result:

- ArticleNo is the term which should be used whenever possible
 - Reason: Compatibility with the existing glossary and the common usage in Komax.
- In ECAT-Devices, ArticleNo is stored as "OrderNo" in the ESI-Category General.
 - Reason: Availability of the information on the fieldbus without running a firmware on the PCB.

2.2 Considered specifications

- D00066869: EtherCAT HW-ID Konzepte Speicherorte HW-Info
- Hardware Identifikation
http://info.ch10.ch.komax.org/doc-fb-elektronik-techmemo_d00010849.pdf
- CANopen
K:\ENTW\Bibliothek\CAN\CANopen\CANopen.pdf
K:\ENTW\Bibliothek\CAN\CANopen\Communication Profile\ds301.pdf
- EtherCAT Modular Device Profile
K:\ENTW\Projekte\EtherCAT\Bus\Specs\Device\Modular_Device_Profiles_0i0i2.zip
- CAN-Object 0x1110:
K:\Proj\PLT\EtherCAT\Entwicklung\EtherCAT\Praesentationen\IntegrationCANopenDevices.pdf

2.3 Rejected alternative ideas

2.3.1 No memory-chips with GUID (global unique identifier)

A memory chip with integrated GUID would improve the traceability with redundant and unchangeable id. The idea of using an i2c-memory-chip with integrated GUID is rejected because of:

- Compatibility problems with the EtherCAT-ASIC (ET1100, ET1200)
- Restricted availability of such memory-chips, no 2nd-source
- Restricted size of such memory-chips
- No free choice in the type of memory

3 Side effects

3.1 Implicit decisions

Vendor ID, product code, revision number and serial number are collected as identity object in CANopen standard. An EtherCAT-master can read this identity object out over the DL (data link layer). If an EtherCAT-

¹ ETG: EtherCAT Technology Group

slave has an AL (application layer) and supports CoE (CANopen over EtherCAT), the AL has to support the identity object.

3.2 Opportunities and risks inherent in the decision

3.2.1 Chances

- Possibility to store all data in one EEPROM.
- Solution that fits to the CANopen- and EtherCAT-standard

3.2.2 Risks

- EtherCAT-Slaves: The access of a local microprocessor to the ESI during bootup-procedure is restricted.
- The acceptance of the new data format depends on the availability of the toolchain (i2c-master).

4 Bureaucratic details

4.1 Persons to which memo is being distributed for comment

Alfred Braun, Benno Häfliger, Benno Zemp, Christoph Peter, Daniel Signer, Dominik Staubli, Hanspeter Koch, Kurt Ulrich, Martin Stocker, Peter Flury, Stefan Viviroli, Thomas Agustoni

4.2 Decision-Makers

Benno Häfliger, Benno Zemp, Christoph Peter, Daniel Signer, Dominik Staubli, Hanspeter Koch, Peter Flury, Stefan Viviroli

4.3 Filing location

Eigner PLM

4.4 Status

See footer

5 Document-History

Index 0	Document created	2008-03-03	tst
Index 1	Renaming of document: - Old : Hardware Identifikation EtherCAT - New: hardware identification V2 Translation to english Adding new data format with category types	2013-11-21	zb
Index 2	Chapter 1.4.1 : adding CANopen Addresses	2014-06-05	zb, bla
Index 3	Chapter 1.4.1 : adding optional CANopen Adresse of Article	2016-05-17	pc, bla