

EtherCAT® MainDevice - Release Notes

(c) 2008 - 2025 by esd electronics gmbh

Supported Operating Systems

- Windows 7/8/10/11 (32-/64-Bit)
- VxWorks 5.4.x/5.5.x/6.x (x86, PPC)
- QNX Neutrino > 6.5.x (x86, x86_64, PPC, ARM32)
- Linux (x86, x86_64, PPC4xx, ARM32, ARM64, RISC-V)
- RTX64 4.x
- OS-9 V5.2 / 6.0
- FreeRTOS with lwIP stack (@ Xilinx Zynq)

Revision history

Listed below are the improvements, changes and fixes between different releases of the [EtherCAT MDevice](#) in reverse chronological order. All changes which affect the binary compatibility of the stack and in consequence might require a re-compilation and/or modification of the application are **marked**:

V1.11.4 - 2025-04-14

New Features:

- Linux: Initial support for the RISC-V (RV64) architecture.
- ENI parser enhancements:
 - Support for the new attribute "Disabled" of the keyword `<MainBox/EoE>` which allows to indicate that an EoE protocol capable SubDevice should not be connected to the virtual EoE switch of the MainDevice which is also reflected with the flag `ECM_FLAG_SLAVE_EOE_DISABLED` in struct `ECM_SLAVE_DESC` introduced with ETG.2100 V1.0.2.
 - Support for `<ProcessData/SmX>` with X=4..15 introduced with ETG.2100 V1.0.1.
 - Support and `<DC/PotentialReferenceClock>` introduced with ETG.2100 V1.0.1.

Changes:

- Updated expat library (if distributed or integrated with the EtherCAT MainDevice stack) from V2.6.2 to V2.7.1.
- Linux LLDF: Updated U-DMA-BUF to V5.2.0-esd. This enhanced version overcomes a limitation of the unmodified version to 'bind' an individual physical device to each DMA buffer instead of supporting just the same device for all DMA buffers.

Fixes:

- Linux: Removed unintended assertions from release builds.

V1.11.3 - 2024-11-30

New Features:

- New entries *ecmGetSmConfig()*, *ecmGetPdoConfig()*, *ecmGetPdoEntry()* and *ecmLookupPdoEntryByIndex()* to enable application access to the SM and RxPDO/TxPDO configuration defined in the ENI file for a SubDevice.
- Added support for ESD MDevice specific configuration keywords in ENI files based on XML Processing Instructions (PIs) which replaces proprietary XML elements as the latter breaks ENI schema validating parsers.
- QNX LLDF: Added LLDF support for SoC integrated network interfaces.

Changes:

- Updated zlib library (if distributed or integrated with the EtherCAT MainDevice stack) from V1.2.13 to V1.3.1.
- Updated expat library (if distributed or integrated with the EtherCAT MainDevice stack) from V2.5.0 to V2.6.3.
- If a SubDevice does not accept an EoE fragment sent from the MainDevice (WKC remains 0), the MainDevice didn't retry this request. Now, the MainDevice stack will retry transmitting the EoE fragment to the SubDevice up to three times before the complete Ethernet frame is discarded.
- Added 'Assign EEPROM access to PDI' command for IB transitions and 'Assign back to ECAT' comand for BI/II transitions to the internally created bus scan configuration (for the EtherCAT Workbench).
- Doubled internal EEPROM access timeout.
- Linux: In header ecm.h replace "LINUX" with compiler builtin "__linux__" if included on Linux platforms.
- Linux LLDF: Updated U-DMA-BUF to V5.0.1-esd. This enhanced version overcomes a limitation of the unmodified version to 'bind' an individual physical device to each DMA buffer instead of supporting just the same device for all DMA buffers and fixes a compile error for kernel version 6.8+.

Fixes:

- Fixed a bug inside the virtual EoE switch that in certain overload situations empty (malformed) EoE packets could be sent to a SubDevice.
- Linux: Fixed reception of VLAN tagged EtherCAT frames without LLD were broken since Linux 3.12.
- Linux LLDF: Fixed dot separated characters of the LLD argument were interpreted as decimal instead of hexadecimal values.
- Linux LLDF: Fixed regression causing a crash if LLD shared library code failed to load dynamically.

V1.11.2 - 2023-09-15

New Features:

- Linux LLDF: Support more than one LLD driver on (x86, x86_64) targets with and without IOMMUs.
- Linux LLDF: New flag `ECM_FLAG_INIT_LLD_PARALLEL` for to support parallel initialization of LLD drivers.

Changes:

- Linux LLDF: Updated U-DMA-BUF to V4.5.0-esd. This enhanced version overcomes a limitation of the unmodified version to 'bind' an individual physical device to each DMA buffer instead of supporting just the same device for all DMA buffers.

Fixes:

- N/A.

V1.11.1 - 2023-08-22

New Features:

- Added support for RTX64 4.x
 - Ethernet communication now based on RTX64 Network Abstraction Layer (NAL) instead of using the proprietary filter driver of previous releases.
 - Added support for *MDevice Clock Shift* in DC mode based on RTX64 API to adjust the HAL timer period count.

Changes:

- **Removed support for all RTX (32-bit) versions as OS support has ended in March 2021.**
- **Removed support for RTX64 2013 / 2014.**
- Linux: Changed internal timer implementation (for worker threads) from interval timer using signals to timerfd without using signals.
- Linux LLDF: Updated U-DMA-BUF to V4.5.0
- Linux LLDF: CPU cache for U-DMA-BUF based memory is disabled if the system indicates no DMA coherency otherwise the CPU cache remains enabled. The application can override this behaviour via `sync_mode` (refer to the U-DMA-BUF documentation for details).

Fixes:

- Fixed regression that the reply of the remote access interface echo command always returned after a delay of seconds. The regression was introduced with a change in V1.10.3 and affects the Round Trip Time (RTT) measurement test of the EtherCAT workbench.

V1.11.0 - 2023-06-06

New Features:

- Application API and remote support for ADS over EtherCAT (AoE) indicated with feature flag `ECM_FEATURE_AOE_SUPPORT`.
- Application API and remote support for Vendor Protocol over EtherCAT (VoE) indicated with feature flag `ECM_FEATURE_VOE_SUPPORT`.

Changes:

- Updated zlib library (if distributed with the EtherCAT MDevice) from V1.2.11 to V1.2.13.
- Updated expat library (if distributed with the EtherCAT MDevice) from V2.2.5 to V2.5.0.
- Linux LLDF: Updated U-DMA-BUF to V4.4.1
- Updated build environment for Windows version from VS2017 to VS2022.
- Windows: Skipped support for all Windows versions before Windows 7SP1.

Fixes:

- Windows: The installer no longer extracts the header <stdint.h> as the version shipped so far breaks C++20 compilation and Visual Studio comes with a compatible version since VS2013.
- QNX: Fixed problems if high resolution timestamp frequency exceeds 1GHz.

V1.10.5 - 2022-12-15

New Features:

- New flag ECM_FLAG_SLAVE_DIAG_DC for ECM_SLAVE_DESC to indicate support of slave specific DC SYNC Window (ESC System Time Difference Register) monitoring.
- New member IDcSysTimeDiff in struct ECM_SLAVE_STATE for DC slave specific SYNC window diagnostic populated if new flag ECM_FLAG_CFG_DIAG_DC is set in ECM_CFG_INIT.
- Enhanced initial DC drift compensation mechanism with a new mode to adapt the number of drift compensation frames dynamically to the target cycle runtime instead of using an initial fixed pre-calculated setup.

Changes:

- Extend the initial DC drift compensation frame with an additional command to monitor the ESC System Time Difference registers.
- Linux: Improved relation between locally captured timestamp and delay to frame transmission for *Direct DC*.

Fixes:

- LLDF: Fixed debug versions of Link Level Driver could not be loaded by release versions of the EtherCAT MDevice library.

V1.10.4 - 2022-05-30

New Features:

- Added support to return the slaves EoE configuration defined in the ENI file with *ecmEoeGetConfig()* in the new data struct ECM_EOE_CONFIG.
- Added support to register a callback handler via *ecmInitLibrary()* to capture the transmitted and received Ethernet frames. The application can define an individual capture filter for each device.
- Added support to register a callback handler via *ecmInitLibrary()* which is called by the (debug build) of the stack instead printing trace messages the default way.
- Extended ENI parser to silently ignore the proprietary TwinCAT specific data types 'AMSADDR', 'AMSNETID' and 'OTCID' instead of generating a warning.
- Linux: Assign unique names to threads internally created by the stack.
- Improved performance and robustness.

Changes:

- Updated zlib library (if distributed with the EtherCAT MDevice) to from V1.2.11 to V1.2.12.
- Updated expat library (if distributed with the EtherCAT MDevice) from V2.2.5 to V2.4.8.
- Decreased detection time of a returned NIC link after it was indicated as lost.

Fixes:

- Fixed ENI parser does not support data type 'BIT' as alternative to the data type 'BOOL'.
- Fixed possible missing last segment indication for application defined SDO downloads.
- Fixed invalid DC start time calculation for *Direct DC* mode if the low part of the local DC reference clock overrun (wrapped) during DC initialization.
- Fixed problems during DC initialization in redundant mode.
- Fixed invalid mapping between some AL status codes and the error text returned with *ecmFormatError()*.
- Prevent 'DC Out-of-Sync' indication for a redundancy situation in *Direct DC* mode.

V1.10.3 - 2021-07-06

New Features:

- Linux LLDF: Support for the combination of PCI-STUB and U-DMA-BUF driver to overcome the deficits of the UIO driver supporting PCI/PCIe network interfaces on x86/x64_84 targets.

Changes:

- Maximum payload size of remote access communication buffer increased from 2 KiB to 4 KiB.

Fixes:

- Fixed missing endianness conversion on big endian architectures using *Direct DC* mode.
- Fixed toggle bit error in segmented CoE transfer.
- Fixed bus scan failed with timeout if 2nd NIC for cable redundancy is enabled.
- Fixed weak robustness in case of segmented TCP/IP packets using the remote access interface.

V1.10.2 - 2020-01-27

New Features:

- Initial release of native 64-bit Windows version.
- Added support for a configurable slave error handling automation which forces a re-initialization (INIT->OP) for slaves with communication errors and/or restores the last ESM state for slaves with unexpected state changes.
- Extend structure ECM_SLAVE_DIAG to return counter of communication disconnects and unexpected state changes.
- New entry *ecmUpdateSlave()* to update some slave instance configuration options after slave creation.
- Added support for new data types BIT9..BIT16 introduced with ETG.1020 V1.2.0.2

Changes:

- Skip DC clock initialization with pending link errors.

- Updated build environment for Windows version from VS2005 to VS2017

Fixes:

- Added missing VLAN tags to frames of initial DC drift compensation which caused an unexpected deviation if DC and VLAN support were enabled.
- Fixed regression introduced with V1.9.3 that the reply to an asynchronous request is transmitted as a new request to the slaves if `ecmProcessAcyclicCommunication()` is called in a separate thread.

V1.10.1 - 2019-06-27

New Features:

- N/A.

Changes:

- N/A.

Fixes:

- Added missing VLAN tags to frames of initial DC drift compensation which caused an unexpected deviation if DC and VLAN support were enabled.

V1.10.0 - 2019-06-14

New Features:

- Support for FreeRTOS with lwIP TCP/IP stack and FatFS filesystem.
- QNX: Initial support for QNX 7.0 and 64-Bit architectures.
- QNX: Support for a virtual EoE based interface to the OS based on TAP devices (requires QNX 6.6 or later).
- New flag `ECM_FLAG_MASTER_DC_CLOCK_LOCAL` to support a **Direct DC Mode**. If configured the EtherCAT MDevice becomes also the DC Master.
- New members `ulExtClockTickNs` in struct `ECM_LIB_INIT` which allows to define an external tick source with the given period that controls the MDevice cycle and which time base is different from the local timer used by the HAL.
- New defines `ECM_VAR_DT_XXX` for data types stored in member variable `usDataType` of `ECM_VAR_DESC`.
- New flag `ECM_FLAG_CFG_SKIP_DATA_TYPE` to prevent parsing variable data types.

Changes:

- Added missing support to parse variable data type definitions from ENI file and store them in the already existent member variable `usDataType` of `ECM_VAR_DESC`.

Fixes:

- N/A.

V1.9.10 - 2018-09-16

New Features:

- New members `ulCycleTimeExceeded` in struct `ECM_DEVICE_STATE` containing the number of times the cycle time was exceeded.
- New profiling type `ECM_PROFILE_FRAME_RX` to benchmark the HAL Rx performance.
- New member `ucDcDriftCompFrames` and `usDcDriftCompCycles` in struct `ECM_MASTER_DESC` to override the internal defaults of the DC drift compensation phase.
- ENI parser adapted to changes of ETG.2100 V1.0.1
 - Support for the attributes "Type" and "Signed" for the keyword .
- New flag `ECM_FLAG_CFG_SKIP_COMMENTS` to save memory by discarding the comments for variable descriptions (if present in the ENI file).
- New error code `ECM_E_NO_LINK`.
- Linux: Initial implementation of the *Link Level Driver Framework* (LLDF) with support for memory mapped as well as PCI based NICs with the Linux UIO driver.
- LLDF: Initial support for the "Enhanced Three Speed Ethernet Controller" (eTSEC) which is part of the Freescale P10xx QorIQ.
- LLDF: Initial support for the combination of AXI EMAC/DMA IP core in a Xilinx FPGA.

Changes:

- Updated zlib library (if distributed with the EtherCAT MDevice) to from V1.2.8 to V1.2.11.
- Updated expat library (if distributed with the EtherCAT MDevice) from V2.1.0 to V2.2.5.
- Linux: Support for a virtual EoE based interface to the OS completely revised and now realized with TAP devices.

Fixes:

- Fixed ENI parser error indication for keyword with unused ports in the middle of the descriptor.
- Fixed very sporadic MDevice crash parsing ENI file passed by an application (e.g. Workbench) using the remote access interface.

V1.9.9 - 2018-02-12

New Features:

- New error code `ECM_E_CYCLE_TIME` to indicate an erroneous (DC) cycle time configuration.
- ENI parser adapted to changes of ETG.2100 V1.0.1
 - Keyword for without section.
- Added support for missing AL status codes 0x03/0x06/0x07/0x0E/0x52/0x70 (mentioned in ETG.1020) in `ecmFormatError()`.
- Added support for missing FoE error codes 0x8010-0x8012 (mentioned in ETG.1020) in `ecmFormatError()`.
- Linux: Added definition of platform specific flag `ECM_FLAG_SCHED_FIFO`.

- Linux: Added support for systems with *Predictable Network Interface Device Names* for the physical Ethernet ports.
- QNX LLDF: Initial implementation of an universal **Link Level Driver Framework** (LLDF) which supports accessing network controller hardware (via PCI) in user mode without the overhead of the OS network stack for high speed, low latency and interrupt free Ethernet communication.
- LLDF: Initial support for the Intel NIC I210 series.

Changes:

- Change slave state to UNKNOWN if ECM_EVENT_SLV_NOT_PRESENT is indicated instead of indicating the last received slave state.
- Ignore early EoE frames during (IP port) configuration phase instead of terminating the startup process.

Fixes:

- Fixed regression that <EtherCATConfig/Config/Cyclic/CycleTime> values in ENI are not used for device initialization if preceding application specific definitions or the EtherCAT workbench vendor tag are missing.
- Fixed FoE return code not passed to the application for failed asynchronous FoE requests.

V1.9.8 - 2017-03-27

New Features:

- Windows: Distinguish between Windows 8 and 8.1.
- Added support for Linux ARMv8 targets in AArch64 mode.

Changes:

- No frame transmission if NIC link is indicated as lost.

Fixes:

- Fixed possible ENI file parser errors for element texts with whitespace characters where a <![CDATA[]]> section does not start/end immediately after/before the element tag.

V1.9.7 - 2015-08-26

New Features:

- New member ucDcSysTimeEpoch in struct ECM_MASTER_DESC to define the epoch of the system time. The MDevice flags ECM_FLAG_MASTER_DC_REF_NO_OFF and ECM_FLAG_MASTER_DCOFFS_TO_LOCALTIME are now deprecated but will still be supported for binary backward compatibility.
- New member usDcSyncWindow in struct ECM_MASTER_DESC to configure the sync monitoring window in ns.
- New entry *ecmDcToUnixTimestamp()* to convert a DC timestamp in nanoseconds into a UNIX timestamp in seconds.
- New flag ECM_FLAG_DEVICE_PROFILE_NO_IO to exclude the Ethernet I/O time in the HAL for the profiling entry ECM_PROFILE_CYCLIC_WORKER.

Changes:

- Increased performance and robustness of acyclic (init) command processing.

Fixes:

- Fixed regression (since V1.9.6) that the ENI parser indicates a syntax error for valid but unsupported mailbox protocols and the transitions "PP" and "SS".
- Fixed configured DC cycle time is forced to 0 if *ecmProcessControl()* is called without configuring a cyclic handler.
- Fixed optional watchdog timeout to shutdown a remote connection announced for V1.9.2 was not enabled.

V1.9.6 - 2015-06-19

New Features:

- New flag `ECM_FLAG_CFG_ENI_ERR_REASON` to return a parser error codes `ECM_ENI_ERROR_XXX` together with the line number as payload of the event `ECM_EVENT_CFG_SYNTAX`.
- New error type `ECM_ERROR_ENI_PARSER_ERROR_CODE` for *ecmFormatError()* to return descriptions of ENI parser errors.

Changes:

- Linux: Cyclic timer implementation is now based on POSIX interval timers instead on the proprietary esd PSYS architecture which eases the stack usage and removes the limitation that the minimum cycle time is limited to 1 ms.
- ENI parser indicates an error if an XML element of type `xs:NMTOKEN` has an invalid value instead of continue silently.
- ENI parser stops processing the configuration file in case of an 'out of memory' situation.

Fixes:

- Fixed XML parser does not accept "true" and "false" as valid argument for data of type `xs:boolean` in ENI file.

V1.9.5 - 2015-04-16

New Features:

- New entry *ecmGetCycleRuntime()* to get the runtime of the current cycle in microseconds.
- New event `ECM_LOCAL_STATE_CTIME_EXCEEDED` to indicate that the cycle time is exceeded if process data exchange is based on cyclic worker threads.

Changes:

- Increased accuracy for values calculated by the profiling code by reducing rounding errors.

Fixes:

- Fixed *ecmAttachMaster()* failed with `ECM_E_DCM_SYNC_ACTIVE` after a previous call to *ecmDetachMaster()* if the MDevice clock drift compensation is configured.
- Fixed MDevice clock drift value not reset correctly on transition from any state to INIT.

- Fixed memory leak using *ecmSoeUpload()*.

V1.9.4 - 2015-02-26

New Features:

- Added support for QNX 6.6.
- New entry *ecmTraceEvent()* to generate a user defined trace event (if event trace support is available for the target OS).
- New member *sUserShiftMaster* in struct *ECM_MASTER_DESC* to configure a platform specific MDevice shift time.
- New member *ISmToSync0Delay* and *ulDcSysTimeDiff* in struct *ECM_MASTER_STATE* for DC diagnostic.
- Added support for configurable stack size of (worker) threads via *pszPlatformConfig* in struct *ECM_LIB_INIT*.
- QNX: Platform specific flag *ECM_FLAG_STACK_NOTLAZY* to disable lazy stack support for (worker) tasks.
- VxWorks: Platform specific flag *ECM_FLAG_FP_TASK* to support floating point operations in (worker) tasks.

Changes:

- Removed support for QNX 6.4.

Fixes:

- QNX: Fixed possible false positive indication about missing data on NIC during startup.

V1.9.3 - 2015-01-27

New Features:

- New entry *ecmBusyWait()* to (busy) wait for several microseconds without yielding.
- OS-9: Platform specific flag *ECM_FLAG_ENABLE_PREEMPTION* to enable system level preemption in the background worker threads for acyclic and cyclic communication.

V1.9.2 - 2014-12-02

New Features:

- New feature flag *ECM_FEATURE_MASTER_SYNC* to indicate the capability of the HAL to synchronize the MDevice clock with DC reference clock (initially implemented for OS-9).
- New macro *ECM_SETUP_REMOTE_WATCHDOG* to configure an optional watchdog timeout to shutdown a remote connection on the server side after a period of communication inactivity.
- OS-9: Added support for multiple SPF instances.

Fixes:

- OS-9: Catch and terminate active cyclic timer thread in case of a fatal signal

V1.9.1 - 2014-11-28

New Features:

- Support for RTX64 2014.
- New flag ECM_FLAG_EXACT_MATCH for ecmLookupVariable() to search exact variable names in addition to substrings.

Fixes:

- Fixed regression (since V1.7.0) that member usStatusCode in struct ECM_SLAVE_STATE was not updated in case of slave failures during the network initialization.
- Fixed "InfoData.State" variables might not be mapped correctly if they just differ in the initial characters of the variable name.
- Fixed problems with regular expression search in ecmLookupVariable().

V1.9.0 - 2014-08-01

New Features:

- General platform support for 64-bit operating systems (starting with RTX64 2013). A 64-bit version of an operating system is indicated with the flag ECM_OS_64BIT in the member usTypeOs of ECM_VERSION.
- Application API to return detailed ESC diagnostic data which is requested cyclically by the MDevice if ECM_FLAG_SLAVE_DIAG_ERRCNT is set.
- New entry *ecmGetSlaveDiag()* which returns the diagnostic data.
- New structure ECM_SLAVE_DIAG.
- New flag ECM_FLAG_SLAVE_DIAG_WDOG.
- Support of a subset of the Extreme Simple Device Configuration Protocol (ESDCP) which allows discovering MDevice instances in the network for remote access without knowing their IP configuration.
- ENI parser now supports Workbench specific keyword in <EtherCATConfig/Master/VendorSpecific> section which is returned in member *ulCycleTime* of struct ECM_DEVICE_DESC. If this keyword is not available the minimum cycle time from the several cycle domains is returned. Inconsistencies are indicated with the new configuration event ECM_EVENT_CFG_CYCLE_TIME.
- New MDevice flags ECM_FLAG_MASTER_DC_REF_NO_OFF and ECM_FLAG_MASTER_DCOFFS_TO_LOCALTIME to influence the DC offset during DC initialization.

Changes:

- Remote control: Use cycle time defined in ENI file in *Control Mode* (Freerun) instead of a fixed value of 1 ms.
- Many internal changes which increased performance and robustness.
- Updated zlib library (if distributed with the EtherCAT MDevice) to from V1.2.3 to V1.2.8.
- Updated expat library (if distributed with the EtherCAT MDevice) from V2.0.1 to V2.1.0.

V1.8.2 - 2014-04-04

New Features:

- Support for remote access in *Monitor Mode* which can be used in parallel to an active application in addition to the already existing remote access in *Control Mode*. The new mode is enabled with the flag `ECM_FLAG_MASTER_REMOTE_INSTANCE` in `ecmStartRemotingServer()`. Remote access related events are indicated with the new event `ECM_EVENT_REMOTE`.
- If `ecmGetMasterDescription()` is called with new flag `ECM_FLAG_MASTER_ENI_FILENAME` set in `ulFlags` the ENI file name without path is returned in which is returned in member `szName` of struct `ECM_MASTER_DESC`.

V1.8.1 - 2014-03-28

New Features:

- ENI parser now supports Workbench specific keyword in `<EtherCATConfig/Master/VendorSpecific>` section which is returned in member `szName` of struct `ECM_MASTER_DESC` if `ecmGetMasterDescription()` is called with new flag `ECM_FLAG_MASTER_PROJECT_GUID` set in `ulFlags`.
- If `ecmGetMasterDescription()` is called with new flag `ECM_FLAG_MASTER_ENI_FILENAME` set in `ulFlags` the ENI file name without path is returned in which is returned in member `szName` of struct `ECM_MASTER_DESC`.

Changes:

- Revised and optimized acyclic communication with respect to reduced resource usage and improved performance.
- Revised and optimized asynchronous communication with respect to reduced resource usage and improved performance.

V1.8.0 - 2014-02-25

New Features:

- Application API and remote support for Servodrive over EtherCAT (SoE) indicated with feature flag `ECM_FEATURE_SOE_SUPPORT`.
- New entries for SoE mailbox requests `ecmSoeDownload()` and `ecmSoeUpload()` with new SoE related defines and data types `ECM_SOE_XXX`.
- New structure for SoE in `ECM_MBOX_SPEC`.
- New flag `ECM_FLAG_SLAVE_SOE` for `ECM_SLAVE_DESC`.
- Definition of SoE specific error codes `ECM_SOE_ERR_XXX`.
- New event `ECM_EVENT_SOE` to indicate SoE notification commands (NFC).
- New error type `ECM_ERROR_SOE_ERROR_CODE` for `ecmFormatError()` to return descriptions of SoE error codes.
- Platform support for OS-9.

Changes:

- Use reserved member of ECM_VERSION for platform specific build information *pszBuildString*.
- Use reserved member of ECM_LIB_INIT for platform specific configuration string *pszPlatformConfig*.

Fixes:

- Fixed flag ECM_ESI_GENERAL definition does not match ETG.1000.6.
- Fixed internally created bus scan configuration (for EtherCAT Workbench) might contain on some platforms an erroneous command to clear the SM mailboxes which could cause a failure to change the affected slaves into PREOP in remote configuration mode.

V1.7.5 - 2014-01-21

New Features:

- New flag ECM_FLAG_REGEX for ecmLookupVariable() to search with regular expressions.
- New flag ECM_FLAG_IGNORE_CASE for ecmLookupVariable() to search case invariant.

Fixes:

- Fixed duplicate acyclic frame causes all pending acyclic frames to be re-transmitted.
- Fixed rare situation that a delayed frame in redundant mode received on primary NIC causes a duplicate.
- Prevent rare false spurious redundancy indications.
- Fixed ECM_LOCAL_STATE_ERROR_NIC2 for event ECM_EVENT_LOCAL indicated with wrong bit value.

V1.7.4 - 2014-01-17

Changes:

- Modified timeout handling of acyclic frames to prevent premature timeouts for EtherCAT configurations with more than 100 slaves.

Fixes:

- Fixed regression introduced with V1.7.3 processing commands in ENI file to configure slave-to-slave copy.

V1.7.3 - 2013-11-26

New Features:

- Extended ECM_SLAVE_STATE with new members usStatusCode which contains the last received value of the ESC AL Status Code register.

Changes:

- Event flags ECM_EVENT_SLV_ID_ERROR and ECM_EVENT_SLV_INIT_ERROR are reset with each call to *ecmRequestSlaveState()* instead of being sticky.
- Internal changes parsing ENI files.

Fixes:

- VxWorks: Fixed VxWorks 5.x does not support binding interface for remote access to a dedicated local NIC.

V1.7.2 - 2013-11-04

Changes:

- Changed EtherCAT commands `ECM_CMD_TYPE_XXX` from enums into defines as the data type was not used anywhere.

Fixes:

- Remote control: Fixed regression introduced with V1.7.0 that complex slave devices fail to enter PREOP on big endian architectures after bus scan.
- Fixed regression introduced with V1.7.0 that auto increment address relocation table used in redundancy mode is not setup correctly on big endian architectures.

V1.7.1 - 2013-09-27

New Features:

- Windows/RTX: Platform specific flag `ECM_FLAG_NO_WSA_STARTUP` to prevent the implicit call of platform specific `WSAStartup()` in `ecmInitLibrary()` which is necessary for the TCP/IP based communication of the remote support.
- Added support for AL status codes 0x61 and 0xF0 (mentioned in ETG.1020) in `ecmFormatError()`.
- Added support for RTX 2011 and RTX 2012.

Changes:

- Removed support for RTX 8.1.2.

Fixes:

- Remote control: Fixed binding to dedicated NIC in `ecmStartRemotingServer()` was ignored.
- Fixed regression with cable redundancy.

V1.7.0 - 2013-08-20

New Features:

- Added support for slave-to-slave copy capability indicated with new feature flag `ECM_FEATURE_SLAVE_TO_SLAVE_COPY` and the new structure `ECM_COPY_INFO`.* Added evaluation of commands in ENI file to configure slave-to-slave copy.
- New reason `ECM_LOCAL_STATE_TRIAL_EXPIRED` for event `ECM_EVENT_LOCAL` to indicate that the trial period of the MDevice has expired.

Changes:

- Use reserved member in ECM_COE_CYC_CMD as pointer to a linked list of ECM_COPY_INFO structures to support slave-to-slave copy. The list is evaluated only if the flag ECM_FLAG_CYC_CMD_COPY_INFO_VALID is set in *usFlags* so the change is binary backward compatible to applications compiled for previous versions of the MDevice.
- Increased robustness of bus scan on platforms with minimum system cycle times of more than 5 ms.

Fixes:

- Fixed misaligned data access problems on ARM platforms.

V1.6.1 - 2013-06-03

Changes:

- Improved method to detect slave initialization commands which failure should be indicated with an ECM_EVENT_SLV_ID_ERROR event.

V1.6.0 - 2013-05-30

New Features:

- Application API and remote support for File Access over EtherCAT (FoE) indicated with feature flag ECM_FEATURE_FOE_SUPPORT.
- New entries for FoE mailbox requests *ecmFoeDownload()* and *ecmFoeUpload()* with new FoE related defines and data types ECM_FOE_XXX. The upload/download of data to/from a slave with FoE support can be performed synchronously or asynchronously and data is provided in application callback handler.
- New entry *ecmFoeGetState()* to poll the state of an asynchronous FoE upload/download.
- New flag ECM_FLAG_SLAVE_FOE for ECM_SLAVE_DESC.
- Definition of FoE specific error codes ECM_FOE_ERR_XXX.
- New error type ECM_ERROR_FOE_ERROR for *ecmFormatError()* to return descriptions of FoE error codes.

Changes:

- Renamed member *type* in ECM_COE_OD_LIST into *ulType* and changed data type from enum ECM_COE_INFO_LIST_TYPE to *uint32_t* to preclude structure size problems on 64 bit platforms.
- Remote control: Removed internal limit of 1000 objects in CoE object dictionary list.

Fixes:

- Fixed two mailbox requests to the same slave with different mailbox protocols might be stored in the same EtherCAT frame.

V1.5.0 - 2013-05-02

New Features:

- New capability to support arbitrary process data layouts ("Packed Layouts") which contain just the process data in addition to the standard layout ("Framed Layout") described in ETG.2100 (EtherCAT Network Information Specification) where the process data always contained the complete Ethernet frame(s).
- New capability to transmit the cyclic frames in different "Cycle Domains".
- New member usDcStartTimeShift in struct ECM_MASTER_DESC to configure the the DC start time offset.
- New members ulRefClockDcHigh and ulRefClockDcLow in struct ECM_DEVICE_STATE containing the value of the DC reference clock.
- QNX: Support optional "SOCK" environment variable required to use an individual instance of io-pkt for NICs dedicated to EtherCAT communication.

Changes:

- Renamed ulShiftTime in ECM_SLAVE_DESC into lShiftTime and changed data type from unsigned to signed to allow negative shift times.

Fixes:

- Fixed that cyclic commands defined in the ENI file with the `0` instead of the `0x0` tag are sent with length 0.
- Fixed MDevice fails to initialize because of an invalid digital signature of esdenet.sys on 64-bit Windows.
- Remote control: Prevent receiving old events after re-connect.
- Fixed problems starting DC slaves.
- QNX: Fixed internal timeouts might expire too early on systems with high tick frequency.

V1.4.8 - 2012-07-16

New Features:

- Initial support for 64-Bit Windows versions with 32-Bit applications (WoW64).
- Remote control: Support to exchange completely user defined EtherCAT frames.
- Extended header ecm.h with additional defines for DC related ESC register.

Changes:

- Remote control: Updated remote protocol version to V1.2.0.

Fixes:

- Fixed EtherCAT frame size limited to 1486 instead of 1500 bytes for ecmAsyncRequest() and ecmAsyncRequests().
- Remote control: Fixed bugs writing to EEPROM introduced with V1.4.7.

V1.4.7 - 2012-06-10

Changes:

- Remote control: Improved EEPROM busy handling during EtherCAT busscan.
- Remote control: Changed mechanism to pass CoE EMCY objects to Workbench to prevent a local overrun if many EMCY messages are received for a single slave.
- Remote control: Updated remote protocol version to V1.1.1.

V1.4.6 - 2012-01-09

New Features:

- VxWorks: IDE integration (WindRiver Workbench / Tornado) for EtherCAT MDevice stack.
- VxWorks: Support for VxWorks 6.9 by changing library module names.

Fixes:

- VxWorks: Fixed returning compile time instead of runtime VxWorks version with ecmGetVersion() with VxWorks 6.x

V1.4.5 - 2011-11-15

New Features:

- New member usAcycFrameTimeout in struct ECM_MASTER_DESC to configure the acyclic frame timeout.

Changes:

- Remote control: Forced bus state to 'INIT' before changing between 'config mode' and 'freerun' to prevent slave error indications.

Fixes:

- Fixed missing re-initialization of acyclic frame timeout in case of a re-transmission.
- Fixed possible buffer overrun in ecmFormatError() for unknown AL status code.
- VxWorks: Fixed problems re-starting HAL timer if caller is running with a higher priority than the timer handler.

V1.4.4 - 2011-10-17

New Features:

- Windows: Initial release of a virtual port driver for Windows as a separate interface.
- New flag ECM_FLAG_DEVICE_VIRTUAL_PORT and member ucVport for struct ECM_DEVICE_DESC to control the local virtual port.

Changes:

- Use a faster, more flexible and more portable hash library for the virtual switch.

Fixes:

- Received EoE message stopped communication with EtherCAT MDevice in 'Config Mode' of 'EtherCAT Workbench'.

V1.4.3 - 2011-09-09

New Features:

- New member ulPlatformFlags in struct ECM_LIB_INIT to pass platform specific flags.
- Windows: Platform specific flag ECM_FLAG_NIC_FRIENDLY_NAME to return the user assigned friendly (alias) name in member szName of struct ECM_NIC instead of the system assigned (NDIS) name.
- New member ucEsiEepromDelay in struct ECM_MASTER_DESC to configure a delay after an EEPROM write operation in ecmWriteEeprom() to prevent overruns of slow PDI interfaces and handling of missing ACKs.
- Extended ecmReadEeprom() to trigger an EEPROM reload on the ESC if size of receive buffer is set to 0. This can also be achieved with the new macro ECM_RELOAD_EEPROM.
- New flag ECM_FLAG_ESI_SKIP_CRC_CHECK for parameter ulOffset of ecmWriteEeprom() to suppress CRC check of ESI data in the range from 0..7 (configuration data).
- New macro ECM_CHANGE_STATION_ALIAS to simplify the change of the station alias address of an EtherCAT slave.
- New macro ECM_EEPROM_TO_ECAT to change EEPROM access from PDI to ECAT.
- New member ulSize for struct ECM_ESI_CATEGORY to return the overall ESI data size if ecmGetEsiCategory() is called with ECM_ESI_CATEGORY_TYPE_END.

Fixes:

- CoE access failed in 'Config Mode' of 'EtherCAT Workbench' if slave device supports AoE mailbox protocol.

V1.4.2 - 2011-08-05

New Features:

- New data type ECM_ESC_EEPROM for ESI EEPROM access.
- Added already documented defines ECM_FLAG_GET_FIRST and ECM_FLAG_GET_NEXT.
- Extended ecmWriteEeprom() to force ESI EEPROM control back to EtherCAT MDevice if number of data to be written is set to 0.
- Support to handle "I->I" transitions.

Changes:

- `ecmReadEeprom()`, `ecmWriteEeprom()` and `ecmAsyncRequest()` will return more detailed error codes in case of communication failures.
- Remote control: Updated remote protocol version to V1.1.0.

Fixes:

- Fixed data in ENI file ignored if HexBin string exceeded 160 characters.
- Fixed mailbox commands in IP are already sent before mailbox is setup properly on slave.
- Fixed bug pending CoE mailbox request is terminated if an unsolicited EoE frame is received before the CoE reply.

V1.4.1 - 2011-06-27

Changes:

- Internal code optimizations and extended debug aids.
- VxWorks: Adapted task names of timer handler to VxWorks standard.

Fixes:

- Fixed bug in `ecmAddCyclicCommand()` that a cyclic frame may exceed the maximum MTU.
- VxWorks: Fixed re-starting timer handler is caller is running with a higher priority than the timer task.

V1.4.0 - 2011-05-16

New Features:

- Introduction of the remote control support with remote protocol version 1.0.0 to control the MDevice on a remote target with the EtherCAT Workbench.
- New entries `ecmStartRemotingServer()` and `ecmStartRemotingServer()` indicated with new feature bit `ECM_FEATURE_REMOTING`.
- New virtual state `ECM_DEVICE_ERROR_ACK` for `ecmRequestSlaveState()` to reset the error indication bit of a slave.

Changes:

- The variable data part of a `ECM_COE_ENTRY_DESCRIPTION` is now endianness converted before it is returned with `ecmCoeGetEntryDescription()`.

V1.3.1 - 2011-05-02

New Features:

- New member `usAcycCtrl` in struct `ECM_DEVICE_DESC` to limit number of acyclic frames per I/O cycle.
- Added support for EoE init commands in ENI file.
- Support for `ECM_COE_FLAG_COMPLETE_ACCESS` flag in `ecmCoeSdoUpload()` and `ecmCoeSdoDownload()` to do SDO requests for a complete object including all sub indexes with one request.
- Support to change slave device into bootstrap state with `ECM_DEVICE_STATE_BOOT` for `ecmRequestSlaveState()`.
- New macros and `ECM_COE_ENTRY_UNIT`, `ECM_COE_ENTRY_DEFAULT_VALUE`, `ECM_COE_ENTRY_MIN_VALUE`, `ECM_COE_ENTRY_MAX_VALUE` and `ECM_COE_ENTRY_NAME` to simplify getting a reference to the unit, min/max/default value and object name in the variable part of a `ECM_COE_ENTRY_DESCRIPTION` structure.

Changes:

- If `ECM_FLAG_MASTER_DIAG` is configured the slave's AL status code register is monitored, too.
- Internal changes to improve performance.

Fixes:

- Fixed that address and channel/priority of mailbox header for CoE requests triggered by application have been initialized to arbitrary values.

V1.3.0 - 2011-01-31

New Features:

- New entries to deal with EtherCAT Slave Information (ESI) EEPROM data `ecmGetEsiCategoryList()`, `ecmGetEsiCategory()` and `ecmCalcEsiCrc()` with several new defines and data types `ECM_ESI_XXX`.
- New entry `ecmGetSlaveHandleByAddr()` to return a slave handle based on position or node address.
- New entry `ecmRequestSlaveState()` to request a change of state for an individual slave.
- New entry `ecmCoeGetAbortCode()` to return the abort code of the last aborted CoE SDO (info) request.
- New entries `ecmSetPrivatePtr()` and `ecmGetPrivatePtr()` to link instance handle with application data.
- Added support for CoE emergency messages with the new event `ECM_EVENT_COE_EMICY` and the new entry `ecmCoeGetEmcy()` to retrieve received EMCY messages from a history buffer.
- Extended `ECM_SLAVE_STATE` with new members `usEmcyReceived` and `usEmcyDiscarded` to reflect the number of received and discarded CoE emergency messages.
- New error type `ECM_ERROR_AL_STATUS`, `ECM_ERROR_COE_ABORT_CODE` and `ECM_ERROR_COE_EMICY_CODE` for `ecmFormatError()` to return descriptions for AL status, CoE abort codes and CoE emergency error codes.
- Added support to parse ENI files compressed as GZIP (*.gz) archive with `ecmReadConfiguration()`.

- New define ECM_PRIO_DEFAULT as sensible platform specific default value for worker task priorities.
- New flag ECM_FLAG_SLAVE_AUTOINC_ADR in ECM_SLAVE_DESC to switch between (default) fixed and autoincrement slave address as argument of ECM_EVENT_SLV and ECM_EVENT_COE_EMCY event.
- New flag ECM_FLAG_MASTER_RESET_SLAVES to reset slaves to a well defined state before the start of the bus configuration.
- New error codes ECM_E_CONN_REFUSED, ECM_E_BROKEN_PIPE, ECM_E_ABORTED and ECM_E_CRC.
- Extended header ecm.h with additional defines and data types for ESC register.

Changes:

- Improved parameter validation and robustness of algorithm to read/write ESI EEPROM data in ecmReadEeprom()/ecmWriteEeprom().
- The slave's ESC configuration area of the ESI EEPROM data (at word address 0-7) can only be written as complete block and with correct CRC. Otherwise ecmWriteEeprom() will return with an error.
- If memory for process data is allocated by the EtherCAT MDevice stack with ecmReadConfiguration() the process image sizes are stored in the configuration data.
- Fixed spelling of define ECM_LOCAL_STATE_MASK.

Fixes:

- Fixed of-by-one error in ecmCoeGetObjDescription() if entry description has more than 255 bytes.
- Fixed ecmCoeSdoDownload() does not allow downloading data of less than 4 bytes.
- Fixed ecmAddAcyclicCommand() failed for slave commands if called by an application.
- Fixed ecmAttach() failed if a slave had only an input process image and no output process image or vice versa.
- Fixed auto increment address relocation table used in redundancy mode is not restored correctly in some cases after a change back into error free mode.

V1.2.1 - 2010-10-04

New Features:

- VxWorks 6.x: Support for VxWorks network driver END_ERR_LINKDOWN/END_ERR_LINKUP events to indicate PHY link state in a hardware independent way.

Changes:

- VxWorks: Improved internal resource management in HAL, if frame transmission failed.

Fixes:

- VxWorks: Fixed page fault in ecmDeleteMaster() cleaning up synchronization objects for mailbox communication if no mailbox support is configured (in ENI file). This problem only occurs if the memory page starting at address 0x00000000 is not readable.

V1.2.0 - 2010-09-01

New Features:

- New entry `ecmFormatError()` to get an error message string corresponding to an error number.
- Support for trial versions of the EtherCAT MDevice stack which is indicated to the application with the new feature flag `ECM_FEATURE_TRIAL_VERSION` and a new error code `ECM_E_TRIAL_EXPIRED` which is returned if the trial period is expired.

Changes:

- Extended `PFN_ECM_HANDLER` of the `pfnXxxCycle` handler to pass a status code to the application in addition to the device handle.

Fixes:

- RTX: Fixed invalid calculation of timeouts > 430 seconds.
- RTX: Fixed error in automatic library registration for RTX 8.1 on Windows versions which have spaces in the destination path name.

V1.1.3 - 2010-08-24

New Features:

- Tornado integration for VxWorks 5.x.
- Improved trace messages in debug build.

Changes:

- Fixed spelling of member `ulCyclicPeriod` of `ECM_PROC_CONTROL`.

Fixes:

- Fixed bug counting NICs in VxWorks 5.5.x.
- Prevent orphan worker task if started with a cycle period of 0.

V1.1.2 - 2010-08-17

New Features:

- Platform support for VxWorks 5.4.x /5.5.x (x86)
- New feature flag `ECM_FEATURE_DEBUG_BUILD` to indicate the debug version of the stack to the application.

V1.1.1 - 2010-05-18

New Features:

- Platform support for RTX 8.1.2
- Support for virtual variable 'FrmXWcState' even if the number of cyclic frames is optimized with the flag ECM_FLAG_CFG_REDUCE_FRAMES which caused a modified layout of frame and/or command index.

Changes:

- Reduced memory footprint of release version for all systems by ignoring all command comments which are just used for trace messages in debug version.
- New error code ECM_E_NO_DRV returned if adapter initialization failed because of a missing driver.

Fixes:

- QNX: Fixed crash in debug version if a command without a comment in the ENI file failed.
- Fixed wrong count of Tx frames and errors in ECM_DEVICE_STATISTIC if the optimized block transmission introduced with the last release is used.
- Fixed asynchronous mailbox CoE requests ecmCoeXXX() always return ECM_E_INVALID_STATE if only polled mailbox devices are defined in the ENI file.

V1.1.0 - 2010-03-19

New Features:

- Platform support for RTX 2009.
- HAL extended by optional optimized block receive and transmit code which drastically improves I/O performance and reduces CPU utilization of QNX in combination with optimized EtherCAT resource manager.
- HAL completely revised and simplified by moving code for cable redundancy handling into device layer.
- Enhanced support for DC drift compensation by adapting the MDevice clock.
- New flag ECM_FLAG_DEVICE_DC and member usCycleDcCtrl in struct ECM_DEVICE_DESC.
- New flag ECM_FLAG_MASTER_DCM_CLOCK_SHIFT in struct ECM_MASTER_DESC to enable MDevice clock drift compensation.
- New member IDeviation in struct ECM_MASTER_STATE.

Fixes:

- Fixed broken support for ECM_FLAG_MASTER_DST_ADDR_VALID in ecmReadConfiguration().
- Fixed wrong signature of ecmAddMboxCommand().

V1.0.1 - 2010-01-15

Fixes:

- Fixed configuration of commands with more than 1472 data bytes fails with error code ECM_E_INVALID_SIZE.

V1.0.0 - 2009-12-01

Changes:

- Extended several data structures with spare parameters for future extensions without the need to change the ABI.

Fixes:

- Fixed possible division by zero exception in ecmGetProfilingData().

V0.9.9 - 2009-11-24

New Features:

- Implementation of high resolution counter in HAL which is returned with ecmGetClockCycles().
- Support for internal and application defined profiling with ecmGetProfilingData() and ecmUpdateProfilingData().
- New entry ecmGetDataReference() as replacement for now deprecated ecmInputRef() and ecmOutputRef().
- New entries ecmCreateCyclicFrame() and ecmAddCyclicCommand() replacing ecmAddCycFrame().
- New flag ECM_FLAG_CFG_REDUCE_FRAMES for struct ECM_CFG_INIT to reduce number of cyclic frames by combining commands of several cyclic frames in ENI data.

Changes:

- Extended ECM_LIB_INIT structure to support new high resolution HAL counter.
- New type ECM_PROC_DATA as argument for ecmGetCopyVector() to determine input/output process image.
- New flag ECM_FLAG_ACYC_CMD_NO_CNT_CHECK to skip internal WC check as replacement for removed definition ECM_IGNORE_WCNT.
- Refactored the variable names in ECM_SLAVE_DESC, ECM_CYC_CMD, ECM_NIC_STATISTIC, ECM_DEVICE_STATISTIC, ECM_MASTER_STATISTIC and ECM_PROFILING_DATA to adapt to the common naming convention with data type related prefix for simple integer types.
- Renamed ecmGetXXXDescription() into ecmGetXXXState() which returns object specific state information in new types ECM_DEVICE_STATE, ECM_MASTER_STATE and ECM_SLAVE_STATE.
- Extended ECM_DEVICE_DESC with member usCycleLinkState and usCycleWatchdog to configure the cycle after which the NIC link state is checked or the watchdog is triggered.

Fixes:

- Fixed all device instances using the same internal tick which caused a wrong timing of acyclic operations in multi device configurations.
- Fixed virtual slave count variables not always updated correctly after recovering from redundancy situation.

V0.9.8 - 2009-09-25

New Features:

- Support to start network with Cable Redundancy if the redundancy situation is already present at startup.
- Support for continuous slave state monitoring if ECM_FLAG_MASTER_DIAG is defined for ECM_MASTER_DESC and ECM_FLAG_SLAVE_DIAG_STATUS is defined for ECM_SLAVE_DESC. To configure this feature for ENI based configuration the new flag ECM_FLAG_CFG_DIAG_STATUS has to be set in ECM_CFG_INIT.
- New reason ECM_EVENT_CFG_INCOMPLETE for event ECM_EVENT_CFG to indicate sections in ENI files which are ignored because mandatory entries are missing.
- Ignore dummy slaves in ENI file (e.g. bus end cap) without any EtherCAT functionality.
- New flag ECM_FLAG_CYC_CMD_NO_CNT_CHECK for ECM_CYC_CMD to indicate that a working counter mismatch for this cyclic command has to be ignored. This becomes the default for all cyclic commands defined in ENI file without section.

Changes:

- Extended ECM_VERSION with new member usVersionHal to check the current HAL layer revision.
- Extended ECM_PROC_CTRL with the new callback handler pfnBeginCycle and pfnEndCycle which can be optionally called at the beginning and end of every process data exchange cycle.
- Changed definition of cycle time in ECM_PROC_CTRL from multiple of ms into multiple of us.
- Renamed reserved member in ECM_SLAVE_DESC into ECM_ETHERNET_ADDRESS ucPhysics to reflect the physical port configuration with new ECM_PHYS_TYPE defines.
- New member ucAlignment for ECM_MASTER_DESC to define the alignment for copy vector and process data size extension for virtual variables.
- Removed parameter alignment from ecmGetCopyVector() because the alignment defined for the MDevice is used.
- Use auto increment address defined in ENI file instead creating them based on slave order.
- New error code ECM_E_INCOMPLETE returned by ecmProcessInputData() to indicate that not all previously transmitted cyclic frames are received.
- Fail DC initialization if no slave as reference clock is configured with new error code ECM_E_NO_DC_REFCLOCK.
- Cyclic data handler is no longer called if the last master instance is detached from the device instance.

Fixes:

- Send DC clock synchronization frames on address of configured reference clock instead of on address of 1st DC slave in line.

V0.9.7 - 2009-09-16

New Features:

- Implementation of Cable Redundancy support indicated with new feature flag ECM_FEATURE_CABLE_REDUNDANCY.
- Re-synchronization of distributed slave clocks can be enabled/disabled with new flag ECM_FLAG_MASTER_DC_RESYNC in ECM_DEVICE_DESC.
- Win32: The EtherCAT MDevice stack comes with a complete C99 compatible header <stdint.h> instead of defining only some types in ecm.h.

Changes:

- Use enums ECM_NIC_XXX instead of defines ECM_XXX_NIC.

Fixes:

- Fixed ECM_EVENT_SLV always indicated the old and not the new state on change.
- Fixed cyclic frames with circular bit set in EtherCAT command header are discarded.

V0.9.6 - 2009-09-09

New Features:

- Implementation of Distributed Clocks (DC) support indicated with new feature flag ECM_FEATURE_DC.
- New flag ECM_FLAG_MASTER_DC for ECM_MASTER_DESC.
- New member dcPrevPhysAddr, dcPrevPort, CycleTime0, CycleTime1, ShiftTime for ECM_SLAVE_DESC.
- New flag ECM_FLAG_SLAVE_DC, ECM_FLAG_SLAVE_DC64 and ECM_FLAG_SLAVE_DC_REFCLOCK for ECM_SLAVE_DESC.
- ENI parser extended to process DC and topology related keys of slave configuration.
- New entry ecmGetCopyVector which returns and optimized vector to copy the data from/into the flat input/output process image.
- New macro ECM_INIT to initialize data structures.
- Support to control order of sending and receiving data in cyclic worker thread via the new flag ECM_FLAG_DEVICE_OUT_BEFORE_IN in ECM_DEVICE_DESC.

Changes:

- Validation of incompatible ABI changes in ecmGetVersion() if member usVersionMaster of ECM_VERSION structure is initialized to the EtherCAT MDevice version the application was build for. On error the call returns with ECM_E_COMPAT.
- Renamed feature ECM_FEATURE_CONFIG_FILE into ECM_FEATURE_ENI.
- MAC addresses in ECM_DEVICE_DESC for primary/redundant adapter are stored in array.
- Input data of a NOP command is no longer copied to application process image.

Fixes:

- Fixed bug in virtual switch if the oldest MAC entry has to be removed.

V0.9.5 - 2009-08-31

New Features:

- Implementation of Ethernet over EtherCAT (EoE) protocol support with a virtual internal switch indicated with new feature flag `ECM_FEATURE_EOE_SUPPORT`.
- New flag `ECM_FLAG_MASTER_VIRTUAL_SWITCH` for `ECM_MASTER_DESC`.
- New member `ulMaxFrames`, `ulMaxMACs` and `usMaxPorts` for `ECM_MASTER_DESC`.
- New flag `ECM_FLAG_SLAVE_EOE` for `ECM_SLAVE_DESC`.
- ENI parser extended to process EoE related keys of MDevice and slave configuration.

Changes:

- Renamed `ecmXxxEEPROM()` into `ecmXxxEeprom()` and changed address parameter to allow fixed as well as increment addresses.
- Changed address parameter of asynchronous request macros to allow fixed as well as increment addresses.

Fixes:

- Fixed bit position from variables in ENI file in `ECM_VAR_DESC` are wrong for values > 65535.

V0.9.2 - 2009-08-11

New Features:

- Support to parse a (ZIP-) compressed ENI configuration with `ecmReadConfiguration()` using minimal memory resources because the complete ENI file is never extracted. The feature is indicated via new flag `ECM_FEATURE_COMPRESSED_ENI`. For this purpose `ECM_CFG_INIT` has been changed.
- Extended `ecmRequestState()` by a 3rd parameter which allows the application to wait for the state change within a given timeout.
- New error codes `ECM_E_OPEN_FILE`, `ECM_E_ENI` and `ECM_E_ARCHIVE` as result of `ecmReadConfiguration()`.
- Extended `ECM_VERSION` with new member to indicate type, version and endianness of target operating system, version of the compression library and the minimum cycle time.
- Win32: Improved accuracy of timer implementation.

Changes:

- Removed `EC_BYTE_ORDER` from header `ecm.h` because target endianness is now returned in `ECM_VERSION`.

Fixes:

- Fixed `ecmReadConfiguration()` doesn't enable mailbox support for MDevice if not at least non slave in the ENI file is configured for FMMU based mailbox communication.

V0.9.1 - 2009-07-30

New Features:

- New define ECM_IGNORE_WCNT to make the MDevice skip the working counter validation for a command.
- New event ECM_EVENT_SLV with new flags ECM_EVENT_SLV_XXX to indicate the slave state to the application.
- Added support to extend the process image input data size if virtual 'InfoData.' variables are defined by the configuration tool with offsets usually outside of the indicated process image size in ecmReadConfiguration().
- Support for slave specific virtual variable 'InfoData.Sate' to reflect it's current state in the process image.

Changes:

- Increased ECM_SZ_NAME from 31 to 63 characters affecting the size of ECM_MASTER_DESC and ECM_SLAVE_DESC.
- Changed signature of ecmInitLibrary() now called with the new ECM_LIB_INIT structure.

Fixes:

- Fixed invalid decimal values in ENI files are not always detected correctly.
- Fixed virtual variable 'FrmXWcState' mapped to wrong position in process data.
- Fixed WC copied to the wrong position in process data in case of WC mismatch of a cyclic command.
- Fixed XML parser doesn't accept "1" as valid argument for data of type xs:boolean in ENI file.

V0.9.0 - 2009-07-21

New Features:

- Implementation of application defined asynchronous mailbox requests which are processed in parallel to the internal mailbox communication indicated via new feature ECM_FEATURE_ASYNC_MBOX_SUPPORT.
- Implementing asynchronous mailbox CoE requests ecmCoeGetOdList(), ecmCoeGetOdEntries(), ecmCoeGetObjDescription(), ecmCoeGetEntryDescription(), ecmCoeSdoUpload() and ecmCoeSdoDownload() with new CoE related defines and data types ECM_COE_XXX to read and write the object dictionary of a complex slave supporting CoE.
- Implementation of ecmGetDeviceDescription() to return the device configuration.
- Support to configure network interface in promiscuous mode if support by the hardware via the new flag ECM_FLAG_DEVICE_PROMISCUOUS in ECM_DEVICE_DESC.
- Support to use the MAC address defined in the ENI file instead of the broadcast address as destination address in the Ethernet frames indicated with the new flag ECM_FLAG_CFG_USE_DST_MAC for ECM_CFG_INIT.

Fixes:

- Fixed ecmGetMasterDescription() returning MAC source instead of MAC destination address.
- Added missing initialization of source address in mailbox protocol header.

- VxWorks: Fixed possible error in timeout handling if the timer does a wrap around.

V0.8.3 - 2009-06-26

New Features:

- QNX: Support for accumulated error counter in NIC statistic.
- QNX: Support for trace messages in slogger.

Fixes:

- Fixed invalid CoE SDO header for expedited transfers on big endian targets.
- Fixed missing re-initialization of timeout handling before re-sending an asynchronous frame.
- QNX: Fixed exchanged counter of Tx/Rx frames and Tx/Rx bytes in NIC statistic.
- Win32: Fixed possible error in timeout handling if the timer does a wrap around.

V0.8.2 - 2009-06-19

Fixes:

- Fixed problems in configurations where not all slaves use the same mechanism (e.g state polling) to check the mailbox.

V0.8.1 - 2009-06-17

Changes:

- Wrong working counter are not indicated to the application if the command in the cyclic frame is replaced by a NOP in the current network state.

Fixes:

- Fixed problems if ecmProcessControl() was called multiple times with different values.
- Fixed problems if ecmAttach() is called again after ecmDetach().
- QNX: Fixed errors in timeout handling.
- Win32: Prevent background worker threads running multiple times in parallel.

New Features:

- Platform support for QNX Neutrino.
- Implementation of mailbox protocol support (cyclically polled and/or FMMU based) to configure complex slaves with internal support of CANopen over EtherCAT (CoE) indicated with new feature flag ECM_FEATURE_MBOX_SUPPORT and ECM_FEATURE_COE_SUPPORT.
- New mailbox protocol related error codes.
- New flag ECM_FLAG_MASTER_MBOX for ECM_MASTER_DESC.
- New flag ECM_FLAG_SLAVE_MBOX, ECM_FLAG_SLAVE_MBOX_POLLING and ECM_FLAG_SLAVE_MBOX_DLL for ECM_SLAVE_DESC.
- New entry ecmAddMboxCommand() and new type ECM_MBOX_CMD add a mailbox command to a slave instance.
- ENI parser extended to process mailbox related keys of MDevice and slave configuration as well as the CoE init commands.
- Implementation of OS independent background worker threads for acyclic and cyclic communication configured with new entry ecmProcessControl() and new type ECM_PROC_CTRL. Every cycle is indicated to the application via a callback handler.
- Implementation to iterate or search variables defined in the ENI configuration with ecmGetVariable() and ecmLookupVariable() returning the description in new type ECM_VAR_DESC if new flag ECM_FLAG_CFG_KEEP_PROCVARS is defined in ECM_CFG_INIT.
- Support of virtual variables which may be part of the ENI configuration to refer to diagnostic information in the process image if new flag ECM_FLAG_CFG_VIRTUAL_VARS is defined in ECM_CFG_INIT. The following virtual variables are supported:

Virtual Variable	Content description
DevState	Device state (Bitmask)
SlaveCount	Number of active slaves (primary adapter)
CfgSlaveCount	Number of configured slaves.
FrmXWcState	Working counter mismatch in cyclic frame

- Implementation of support for virtual variables a callback handler registered with ecmRegisterHandler() to return the Ethernet link state for platforms which have no hardware independent support for this.
- Implementation of ecmGetMasterDescription() to return the MDevice configuration.
- Implementation of diagnostic and statistic functions ecmGetNicStatistic(), ecmGetDeviceStatistic() and ecmGetMasterStatistic() indicated with new feature flag ECM_FEATURE_DIAGNOSTIC.
- New flag ECM_FLAG_CYC_CMD_SLAVE_STATE to mark a cyclic BRD command requesting ESC AL status register.

Changes:

- Renamed `ecmRegisterHandler()` into `ecmInitLibrary()` because of its extended configuration purpose.
- Renamed struct `ECM_DEVICE_INIT` into `ECM_DEVICE_DESC` (affects 2nd parameter of `ecmCreateDevice()` and member of `ECM_CFG_INIT`).
- Renamed struct `ECM_MASTER_INIT` into `ECM_MASTER_DESC` (affects 2nd parameter of `ecmCreateMaster()` and member of `ECM_CFG_INIT`).
- Extended name of several flags starting with `ECM_FLAG_XXX` to indicate the related data structure in the name.

Fixes:

- Fixed some memory leaks in cleanup.

V0.7.1 - 2009-05-29

Fixes:

- Fixed resource leak returning transmitted Ethernet packets back to the HAL layer which might stop network initialization in the Win32 implementation depending on the configuration.

V0.7.0 - 2009-04-24

New Features:

- Implementation of a callback handler registered with `ecmRegisterHandler()` to indicate errors/events asynchronously to the application. The following event types are supported:

Event	Indication
<code>ECM_EVENT_CFG</code>	Errors parsing the configuration file.
<code>ECM_EVENT_INTERNAL</code>	Internal EtherCAT MDevice and I/O communication events.
<code>ECM_EVENT_WCNT</code>	Wrong working counter (WC) of cyclic commands.
<code>ECM_EVENT_STATE_CHANGE</code>	Events for changes in the EtherCAT MDevice state machine.

Implementation of a callback handler registered with `ecmRegisterHandler()` to return the Ethernet link state for platforms which have no hardware independent support for this.

- Implementation of `ecmGetSlaveDescription()` to return the slave configuration.
- Implementation of diagnostic and statistic functions `ecmGetNicStatistic()`, `ecmGetDeviceStatistic()` and `ecmGetMasterStatistic()` indicated with new feature flag `ECM_FEATURE_DIAGNOSTIC`.
- New feature flag `ECM_FEATURE_ASYNC_FRAME_SUPPORT` to indicate support of asynchronous requests introduced in V 0.6.0.
- New flag `ECM_FLAG_CYC_CMD_SLAVE_STATE` to mark a cyclic BRD command requesting ESC AL status register.

Changes:

- Renamed struct `ECM_SLAVE_INIT` into `ECM_SLAVE_DESC` (affects 2nd parameter of `ecmCreateSlave()`).

Fixes:

- Fixed bug that output data of cyclic commands not marked with `ECM_FLAG_CYC_CMD_COPY_OUTPUTS` are not initialized to zero before transmission.

V0.6.0 - 2008-11-28

New Features:

- Implementation of application defined asynchronous requests which are processed in parallel to the standard exchange of EtherCAT telegrams via `ecmAsyncRequest()` and `ecmAsyncRequests()`.
- Support for common asynchronous requests to EtherCAT slaves implemented as macros based on asynchronous requests:

Macro	Action
<code>ECM_GET_SLAVE_COUNT</code>	Returns the current number of active slaves
<code>ECM_GET_ESC_INFO</code>	Returns the ESC information register of a slave
<code>ECM_GET_DL_STATUS</code>	Returns the ESC DL status register of a slave
<code>ECM_GET_ESC_ERROR_COUNTER</code>	Returns the ESC error counter register of a slave
<code>ECM_CLR_ESC_ERROR_COUNTER</code>	Resets the ESC error counter register of a slave

- Implementation of special asynchronous request sequences to read/write the ESC EEPROM with `ecmReadEEPROM()` and `ecmWriteEEPROM()`.
- New helper code `ecmCpuToLe()` to convert a data structure from little endian into host byte order.
- New error codes `ECM_E_INVALID_CMD` and `ECM_E_INVALID_ADDR`.
- Support to read the configuration file from a buffer in addition to file I/O indicated with the new feature flag `ECM_FEATURE_FILE_IO`.
- Extended header `ecm.h` with many additional defines and data types for ESC register.

Changes:

- Changed signature of `ecmReadConfiguration()` now called with the new `ECM_CFG_INIT` structure.
- Extended `ECM_MASTER_INIT` structure with the parameter `pInput` and `pOutput` to support user defined buffer for the process image.

V0.5.3 - 2008-11-10

New Features:

- Implemented missing cleanup support.
- New flag ECM_DEVICE_STATE_TRANSITION in state returned by ecmGetState() to indicate that the MDevice currently processes a state transition.

Changes:

- Renamed ecmAttach() into ecmAttachMaster().

Fixes:

- Fixed bug in ENI parser which might cause a failed network configuration as only a sub-string of a value was evaluated.
- Fixed bug that init commands are sent more than once if the origin of a state change isn't ECM_DEVICE_STATE_INIT.

V0.5.2 - 2008-10-29

New Features:

- Support for timeout handling of init commands.

Fixes:

- Fixed bug that the check of replied frames with more than one init command is only completed successful for the 1st init command on big endian architectures.

V0.5.1 - 2008-10-23

Fixes:

- Fixed bug that only the state machine of the 1st MDevice instance was executed in case of a multi-segment configuration.

V0.5.0 - 2008-09-30

New Features:

- Initial release for Windows (Little Endian architecture) and VxWorks 6.x (Big Endian architecture) with support for the following core services:
 - Parser for an EtherCAT network configuration file (ENI).

- Implementation of EtherCAT state machine.
- Transmission of init commands.
- Cyclic exchange of process data.
- Addressing of different EtherCAT segments via VLAN tags and virtual MDevice instances.