EtherCAT® Slave - Release Notes

Revision history

Listed below are the improvements, changes and fixes between different releases of the EtherCAT Slave Stack in reverse chronological order.

The history covers changes in the stack (source) code as well as in the HAL code for different EtherCAT Slave Controller (ESC), System on Chip (SOC), esd ESC hardware interfaces and/or supported operating systems.

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:

V1.3.16 - 2022-04

New Features:

- N/A.

Fixes:

- N/A

Changes:

- The minimum GLIBC version for the shared library of the x86_64 Linux is set back to 2.14 from 2.25 to keep up the support of legacy Linux versions.
- If the ESC is configured for EEPROM emulation the slave stack returns with the new error code ESS_RESULT_ERROR_EEPROM_CONTROL if the stack is not compiled to support this operation mode.

Example Code:

- *Complex* example;
  - Added support to initialize the data of an emulated EEPROM from file.

V1.3.15 - 2022-03

New Features:

- New platforms: QNX 6/7 on ECS-xxx/FPGA and ECS-PCIe1100 cards.

Fixes:

- Fixed regression introduced with V1.3.8 which broke functionality big endian architectures.
- Linux: Fixed possible stall after ~1 day of continuous operation.

Changes:

- Changed last parameter of *essODUpdatePDOConfiguration()* from ESS_BOOL to ESS_PDO_CFG_FLAGS keeping binary compatibility.

V1.3.14 - 2020-05

Fixes:

- Fixed abort of CoE SDO Segmented Download caused by the erroneous detection of a toggle error in the third segment received from the EtherCAT master.

V1.3.13 - 2020-04

New Features:

- Added support for the new flag ESS_OD_OBJECT_FLAGS_UNIQUE_ENTRY_NAMES which returns unique names for RECORD entries with identical names (which otherwise brakes the dictionary export of CTT V2.2.0).

Changes:

- Return ETG.1020 compliant empty entries for PDO parameter sub index 6-9 if they do not exist instead aborting with unknown sub index (to prevent error messages in the log windows of the CTT V2.2.0).
- Added support to return an empty SDO Info structure according to ETG.2010 V1.1.0 and later if an entry of a RECORD does not exist instead sending an abort for an unknown sub index (to prevent error messages in the log windows of the CTT V2.2.0).

V1.3.12 - 2020-03

New Features:

- Added (HAL layer) support for TI-RTOS.
- Support for 'libprussdrv' version 2 on TI Sitara (Linux) which also works on newer kernel with uio-module-drv (i. e. kernel 4.14 and later).

V1.3.11 - 2020-01

New Features:

- Added defines COE_ACCESS_SAFEXXX and COE_DATATYPE_FSOE_FRAME_ELEMENTS for FSoE
- Added define ESS_OD_ENTRY_FLAGS_DYNAMIC_SIZE and CoE event callback type ESS_OD_ENTRY_CALLBACK_GET_UPLOAD_SIZE to support upload of objects with dynamic sizes.

V1.3.10 - 2019-08

New Features:

- N/A - Private release

V1.3.9 - 2019-05

New Features:

- New macro SWAP_WORD to exchange LSW/MSW of an 32-bit value.

Changes:

- Added defines for COE objects 0x1C32 and 0x1C33 as COE_OBJ_OUTPUT_SM_XXX.
- Changed default name from "SM Type" into "Sync Manager Communication Type" to suppress a warning in the CTT.
- Append sub index number to "PDO Mapping" to have unique object names.
- TI Sitara HAL updated to use PRU EtherCAT firmware 0x4EE

Fixes:

- Fixed race condition (more likely for Sitara) in SM_UpdateAll() by changing the order of reading SM physical address and SM control register which implicitly clears the SM activity bit. The previous implementation was prone to ignore an SM change if requests to different SMs came very consecutively (as newer versions of the CTT do).

V1.3.8 - 2018-04

New Features:

- New Windows (X86/x64) driver V1.1.0 which supports all desktop versions of Windows 10 started with or without secure boot.
- Enhanced DMA support for ECS-PCIe-FPGA in Windows HAL layer (requires device driver V1.1.0).
- Optimized performance to handle SM and ESM change indications.
- New flag ESS_SM_EVENT_FLAG_SAFE_OUTPUTS for member sm of struct ESS_CBDATA_SM_EVENT passed to cbDataOutputsUpdated() to indicate the output's 'safe state'.
- New macros ESS_MAP_ENTRY, ESS_MAP_EXTEND and ESS_MAP_DUMMY to define PDO mapping according to ETG.1020 with support for special padding object dictionary entries which extend a mapping for objects with more than 31 bytes.

Changes:

- ABI version changed to 7
- Members of struct ESS_CBDATA_SM are re-ordered to reduce number of ESC read operations for the SM update to improve performance of the SM change indication.
- The handler cbDataOutputsUpdated() is also called after the initial P->S state to show the same behavior as after a following O->S change.

Fixes:

- Fixed PDI events might not be acknowledged properly in SAFEOP which may lead to target/implementation specific missing or spurious PDI interrupts.

Example Code:

- *Complex* example;
  - Added support for ESS_MAP_XXX macros to define PDO mapping.
  - Added indication for usage of ESS_SM_EVENT_FLAG_SAFE_OUTPUTS in cbOutputsUpdated().
  - Support to request serial number (of esd ECS-xxx/FPGA interfaces) with ESS_IOCTL_GET_DEV_SERIALNO32.
  - Extended object 0x2100 with more sub entries for different data types and the possibility to force a CoE EMCY message if sub index 10 is written.

V1.3.7 - 2017-08

New Features:

- Enhanced DMA support for ECS-PCIe-FPGA in LINUX HAL layer (also needs uio driver v2.2.0)

Fixes:

- Fixed essHALGetTime() returning wrong timestamps in LINUX HAL.

V1.3.6 - 2017-07

New Features:

- Added ETG.1020 V1.1.0 and later compliant behavior to return maximum number of entries via the SDO Info Service instead of the number of available entries for PDO assignment and mapping objects if ESS_OD_SDO_INFO_MODE_ETG1020 flag is set with essODCreate().
- Added support with the new entry essODSetArrayMax() to define a maximum number of entries for arrays which exceeds the available number of entries if ESS_OD_SDO_INFO_MODE_ETG1020 flag is set with essODCreate().
- According to ETG.1020 V1.1.0 for non-existent entries in dynamic arrays which less elements than the maximum capacity an 'empty' definition is returned via the SDO Info Service. As the latter seems to fail the current CTT (V2.1.0) the previous behavior to return the data of an existent entry remains the default.

Changes:

- Return COE_ABORTCODE_ACCESS for SDO Info object and entry requests < index 0x1000 as defined in ETG.1020 V1.1.0 and later.

Fixes:

- Fixed sub-index 0 of the TxPDO/RxPDO parameter is always returned as 0x9 instead of the value which reflects the configuration.
- Fixed number of available entries was ignored in SDO info service replies for PDO assignment and mapping objects as well as arrays.

V1.3.5 - 2017-05

New Features:

- DMA support for ECS-PCIe-FPGA in LINUX HAL layer (also needs uio driver v2.1.0).
- New configuration options CFG_ESC_HAVE_WRITE_FUNC and CFG_ESC_HAVE_READ_FUNC which calls low level ESC read/write functions even if CFG_ESC_HAVE_POINTER is defined.

V1.3.4 - 2017-04

New Features:

- FoE net mailbox size is indicated to application in callbacks (see interface change below).
- Reduced code size for FoE handling.

Changes:

- ABI version changed to 6
    - ESS_CBDATA_FOE_OPEN and ESS_CBDATA_FOE_DATA have been extended to indicate the current net mailbox size.
    - **Fixes:**
- Fixed data type of 0x1018:00 was defined as UDINT instead of USINT.

Example Code:

- *Complex* example;
    - Added support for member maxDataLen in ESS_CBDATA_FOE_OPEN in cbFoeOpen().
    - Added support for member maxDataLen in ESS_CBDATA_FOE_DATA in cbFoeData().

V1.3.3 - 2017-03

New Features:

- Added backward ABI compatible support to explicitly define a special bootstrap mailbox configuration in ESS_SM_CONFIGURATION as type SM_TYPE_MBXIN_BOOTSTRAP/SM_TYPE_MBXOUT_BOOTSTRAP and the new member variable smConfigCountBootstrap of ESS_CONFIGURATION.
- Added ESC register defines ESC_REG_MII_XXX and ESC_REG_REG_PHY_XXX to the header <ecatDefs.h>.
- Added missing defines for FoE error codes defined in ETG.1020 FOE_ERRORCODE_XXX to the header <ecatDefs.h>.

Fixes:

- Revised FoE busy handling and fixed missing implementation of "FoE Read with Busy" for the first packet.
- Fixed false CoE EMCY message for S->P transition under certain timing conditions.

Example Code:

- *Complex* example;
  - Adapted to change in ESS_CONFIGURATION for special mailbox configuration.
  - Simulate "FoE Busy" behavior, if FoE filename contains the string "busy".

V1.3.2 - 2017-02

New Features:

- Added missing support for ESS_RESULT_INVALID_PDI_CONFIGURATION into ecmFormatResult().
- Added ESC register defines ESC_REG_DCUNITCONTROL and ESC_REG_DCSYNCACTIVATION to the header <ecatDefs.h>.
- Added defines for MDP CoE objects COE_OBJ_MDP_XXX to the header <ecatDefs.h>.
- Improved indication of errors verifying input/output SMs (in the debug build).
- Linux AM355x HAL prevents concurrent driver instances by locking file handle of PDI IRQ exclusively.
- Linux AM355x HAL returns PRU firmware version with ESS_IOCTL_GET_DEV_DRIVERINFOSTR.
- VxWorks HAL now ready to support big endian architectures.

Changes

- TI Sitara HAL updated to use PRU EtherCAT firmware 0x3D4

Fixes:

- Fixed errors in Linux AM355x HAL that tick overruns could stall the stack.
- Fixed error in VxWorks HAL interrupt handling that IRQ was not masked correctly.

Example Code:

- *Complex* example;
  - Fixed endianness problem detecting esd hardware on big endian architectures.

V1.3.1 - 2016-08

New Features:

- Preset default abort code before calling cbCoEReadWrite() from COE_ABORTCODE_GENERAL to COE_ABORTCODE_SUBINDEX so the CTT will not fail because of a missing object (unhandled by the application callback).
- Windows HAL returns FPGA build info (for esd ECS-xxx/FPGA interfaces) in addition to driver info for ESS_IOCTL_GET_DEV_DRIVERINFOSTR.
- Windows HAL returns order number of ECS-PCIe/1100 with ESS_IOCTL_GET_DEV_ORDERNOSTR.
- Added missing AL status codes REG_VAL_ALSTATUSCODE_XXX defined in ETG.1020 and macro REG_VAL_ALSTATUSCODE_VENDOR to define vendor specific AL Status Codes to the header <ecatDefs.h>.
- Improved code quality and ANSI C compliance by applying very strict static code analysis rules.

Fixes:

- Fixed CTT 2.0.41 fails for TF-1201 #5 because the 'Explicit Device ID' support introduced with V1.3.0 was implemented based on ETG.1000.6 V1.0.3 which contains a bug for the I->B transition.
- Fixed error in VxWorks HAL tick overrun handling which stalls the call to cbCyclic().

Example Code:

- *Complex* example;
    - Removed code in cbDCEvent() to read ESC SYNCx status register which is handled by the stack internally since V1.2.0.
    - Choose interface to run on with a (command line) argument.
    - Create individual CoE entries for esd ECS-PCIe/1100 and ECS-xxx/FPGA interfaces.

V1.3.0 - 2016-07

New Features:

- Added support for the 'Explicit Device ID' mechanism 'Requesting ID' via AL Status Register (0x134) according to ETG.1020 if CFG_ESS_EXPLICIT_DEVICE_ID is set.

Changes:

- ABI version changed to 5
    - The ESS_CBDATA_STATE_REQUEST struct has been extended for 'Explicit Device ID' support.

Fixes:

- Fixed regression that SM is not disabled for O->S transition with error test in CTT introduced with stack V1.1.0.
- Fixed crash in Windows HAL if library is loaded/unloaded dynamically.

Example Code:

- *Complex* example;
    - Added support to use the ETG.1020 'Explicit Device ID' mechanism via 'Requesting ID'.
    - Added support for the ETG.1020 'Explicit Device ID' mechanism via 'Direct ID'.
      *: Added support for the ETG.1020 'ID Reload Object'.

V1.2.0 - 2016-05

New Features:

- New platforms
    - HAL for Linux on TI Sitara AM335x.
    - HAL for Infineon XMC4800.
    - HAL for VxWorks 7 (on esd ECS-xxx/FPGA interfaces).
- Internal changes

    - The stack is extended with two internal entries Sync0() and Sync1() which can be called directly from the HAL layer if the ESC DC sync signals are connected to the MCU. The latter can be controlled with the new configuration options CFG_HAL_USE_SYNC0_IRQ and CFG_HAL_USE_SYNC1_IRQ.
    - The stack autonomously figures out based on the (SII) configuration if SYNC is configured in ACK mode (SYNC pulse length 0) or not and applies a required ACK internally so this is no longer necessary in the application as in previous versions of the stack.
- Driver for esd PCIe EtherCAT slaves updated
    - Added support for the new esd PCIe hardware {ECS-PCIe/FPGA}.
    - Enable MSI support for ECS-xxx/FPGA cards on Windows via a revised INF file.

V1.1.0 - 2015-05

New Features:

- API changes
    - New entry essIoctl() to allow the application to do custom/hardware specific tasks with the stack/HAL.
- Driver for esd PCIe EtherCAT slaves updated to V1.0.3
    - Stack's HAL can now obtain build info string from driver with its version etc...
    - Added support for new esd {ECS-XMC/FPGA} and {ECS-PMC/FPGA} cards

Changes:

- ABI version changed to 4
    - essODSyncInputMappedEntries() renamed into essSyncInputs().
    - Added flag to force immediate copy.
    - essHALInit()/essHALFinish() renamed into essHALOpen()/essHALClose().
    - Support for individual device references instead of HAL global operations.

Fixes:

- Application was able to block incoming mailbox request handling by crowding send mailbox with EoE frames.
- CoE emergencies created by slave stack itself weren't sent.
- Output SMs were disabled in Op-SafeOp transition
- State change to Init required explicit ErrAck bit in 0x0120 to clear error
- FoE upload sent trailing 0 byte chunk even if not required
- LED handling was not working with multiple slaves if first slave was not running

Example Code:

- *Complex* example supports essIoctl() to show more hardware info.

V1.0.5 - 2014-08

Fixes:

- essODUpdatePDOConfiguration()

    - Did not delete PDO object with entries NULL as stated in the manual
- Error LED behavior
    - State change to ErrorInit did not enable it. Reset did not occur on AL Control Err Ack
- essStop() / cleanup
    - Wasn't allowing to completely start over
- essOpen() / essClose()
    - The documentation now correctly states that these functions must be serialized if multiple devices are handled by using a different task/thread for each SM verifications updated for CTT compliance.

Example Code:

- ECS-PCIe/1100 Sample Slave
    - ESI now includes DC modes

V1.0.4 - 2014-06

Fixes:

- EEPROM Emulation
    - Error bits now cleared by any valid command, not just Idle command
- Driver for esd's ECS-PCIe/1100 updated to V1.0.2
    - Updating error LED could take very long

Example Code:

- Minor changes for *Complex* example.

V1.0.3 - 2014-03

New Features:

- Added support for segmented SDO list transfers
- Added basic support for SoE and AoE (similar to VoE: stack handles mailbox protocol but no API for AoE/SoE/VoE itself)
- Supporting fragmented mailbox transfers with AoE/SoE/VoE
- Added more debug outputs and added more sanity checks to essOpen()
- Added CFG_ESS_SERVE_CUSTOM_LED to allow to disable all LED handling
- Added EoE rx/tx frames to ESS_STATISTICS

Example Code:

- *Complex* example
    - Added several macros to allow building without changing the source.

V1.0.2 - 2013-05

New Features:

- Added support for TI Sitara (AM335x) PRU
- API Changes
    - Added functions to handle PDO parameter objects
    - New callbacks for EEPROM emulation and distributed clocks
    - New ESS_OD_ENTRY_FLAGS
- Simplified handling of multiple ESCs.
- Other
    - Added ability to manually abort in some CoE events.
    - Implemented SERVE_RUN_LED support.
    - Windows drivers updated for Windows 8.

Changes:

- ABI version changed to 3
    - ESS_CONFIGURATION struct has been changed
        - See ESS_CONFIGURATION and new ESS_CALLBACKS initialization in the samples.
    - Appended parameter to essODGetPDOConfiguration()
        - Just for future use -> Has to be set to 0.

Fixes:

- SM verifications updated for CTT compliance.

Example Code:

- *Complex* example now includes FoE samples and Bootstrap support.
- ESI for ECS-PCIe/1100 now includes Bootstrap support.

V1.0.0 - 2012-05

New Features:

- Initial public release with ABI version 1.