EtherCAT Slave Errata for PRU ICSS EtherCAT Firmware Version x.5.6

Single datagram accessing multiple FMMU mapped areas using LRD/LWR commands

- **Issue/ Failure Description or state**
  - SDOCM00092510/PINDSW-47 : Single datagram accessing multiple FMMU mapped areas in a single slave will only update the data corresponding to first FMMU in datagram
- **Conditions in which failures occur**
  - Single datagram accessing multiple FMMU mapped areas in single slave.

    FMMU0(0x1000:0x1007)->SM2 (Write SM)

    FMMU1(0x1000:0x1007)->SM3 (Read SM)

    FMMU2(0x1008:0x100F)->SM4 (Write SM)

    FMMU3(0x1008:0x100F)->SM5 (Read SM)
    - Single LRD to access (0x1000:100F) will only access SM3
    - Single LWR to access (0x1000:100F) will only access SM2
    - 2 LRD/LWRs are required in this case to access both pair of SMs - LRD1/LWR1(0x1000:0x1007) and LRD2/LWR2(0x1008:0x100F)
- **Root cause**
  - Increased code memory requirements in firmware to implement this support
- **Work around**
  - For above example instead of one LRD datagram to logical address 0x1000 and length 16, master needs to send two LRD datagrams, one to logical address 0x1000 and length 8 second to logical address 0x1008 and length 8 or use LRW in place of LRD/LWR which is more efficient for most of the use cases

NOTE: This issue will not be fixed

PD/PDI watchdog counter issue

- **Issue/ Failure Description or state**
  - SDOCM00098105/PINDSW-72: PDI/PD watchdog counter incremented by 1 whenever PDI/PD watchdog is disabled using EtherCAT master
- **Conditions in which failures occur**
  - Whenever EtherCAT master disables WD by writing zero to respective Watchdog Time registers (0x410:0x411 or 0x420:0x421)
- **Root cause**
  - This is PRU-ICSS h/w behavior

NOTE: This issue will not be fixed

LRD access on unused registers

- **Issue/ Failure Description or state**
  - SDOCM00098950/PINDSW-74: LRD access on unused registers results in WKC increment
- **Conditions in which failures occur**
  - LRD access on unused registers result in WKC increment
- **Root cause**
  - Firmware does not support register protection in LRD mode at this moment, it requires more firmware footprint to support, this minor spec compliance does not justify the footprint increase and no Write Only registers in ESC

NOTE: This issue will not be fixed

LRW access to non-interleaved input and output process data of multiple slaves does not work

- **Issue/ Failure Description or state**
  - SDOCM00105048/PINDSW-141: LRW access to non-interleaved input and output process data of multiple slaves does not work. SOEM accesses slaves in LRW mode this way
- **Conditions in which failures occur**
  - Single LRW datagram accessing FMMU mapped areas in multiple slaves and PD out is mapped
    FMMU0(0x1000:0x1007)->SM2 #1(Write SM) FMMU1(0x1008:0x100F)->SM2#2 (Write SM)
    FMMU2(0x1010:0x1017)->SM3 #1 (Read SM) FMMU3(0x1018:0x101F)->SM3#2(Read SM)
    - Single LRW access from (0x1000:101F)
- **Root cause**
  - Increased code memory requirements in firmware to implement this support as well as non-interleaved access I/O data is not a very optimal use of EtherCAT – it increases the cycle time overhead/datagram size and not effective use of LRW datagram which can perform read and write in the same cycle.
- **Work around**
  - Use LRD/LWR datagram to access process data
  - Use LRW datagram to access process data
    - Input and output overlaid on the same logical address range (TwinCAT usage)
    - Input and output of a given slave back to back in logical address space
      - FMMU0(0x1000:0x1007)->SM2 #1(Write SM) FMMU1(0x1008:0x100F)->SM3#1 (Read SM)
      - FMMU2(0x1010:0x1017)->SM2 #2 (Write SM) FMMU3(0x1018:0x101F)->SM3#2 (Read SM)

NOTE: This issue will not be fixed

Error frames with “SOF but no SFD” are not counted if arrived on reverse path port

- **Issue/ Failure Description or state**
  - PINDSW-2204 : If a frame with SOF but no SFD arrives on the reverse path, EtherCAT Slave cannot detect such frame. Therefore, it also doesn't count this frame in the error counters. According to EtherCAT specs it should be counted. These error frames are detected and counted on the forward path. The issue is only for the reverse path.

- **Conditions in which failures occur**
  - With at least two devices connected in the chain, on the device with both ports connected, if a frame without SFD arrives on the OUT port (reverse path port), then that frame is not counted in any of the error counters of the ESC
  - When the hardware forwards these frames (without SFD) it adds a SFD and a specific pattern pattern "0x5a 0x5a 0x5a 0x5a" at the end of the frame along with an odd nibble

- **Root cause**
  - Because of the way firmware handles the Ethernet frames on reverse path, there is no hardware/software provision for the firmware to even detect frames without SFD. Thus, it cannot increment the error counters accordingly.

- **Work around**
  - When such frame (without SFD, very rare phenomenon) is received by a slave, even though the receiving slave cannot detect the frame, it forwards the frame with added odd nibble. So all the following slaves will detect that frame as forwarded error.
  - Also when the slave forwards the frame, it adds a specific pattern "0x5a 0x5a 0x5a 0x5a" at the end of the frame, which can be identified using wireshark logs.

NOTE: This issue will not be fixed

System time of next Sync0 pulse register (0x990:0x993) is not updated instantaneously

- **Issue/ Failure Description or state**
  - PINDSW-2360: System time of next Sync0 pulse register (0x990:0x993) is not updated instantaneously, resulting in read of incorrect value if read immediately after sync pulse.
- **Conditions in which failures occur**
  - When Sync0 pulse register is read immediately after a sync pulse occurs.
- **Root cause**
  - The Sync0 pulse register is updated in the firmware with a typical delay of ~1.7us and worst case delay of ~5.6us. If the Sync0 pulse register is read after the sync pulse but before the register is updated (before the above limit expires) it can result into read value returning previous pulse time which is incorrect.

NOTE: This issue will not be fixed

Read permissions and byte level write permissions for RW type commands are not checked

- **Issue/ Failure Description or state**
  - PINDSW-5135: Read permissions and byte level write permissions for RW type commands are not checked. Write permissions will be checked at word level.
- **Conditions in which failures occur**
  - When the firmware processes APRW, FPRW, BRW and LRW commands
- **Root cause**
  - The firmware process path for this particular scenario is taking longer time to process if byte level read and write permission checks are present. This leads to firmware not able to process the command correctly and therefore failing the RW operation.

NOTE: This issue will not be fixed

RX_ER counter does not count errors outside frame in MII RX_CLK units precisely

- **Issue/ Failure Description or state**
  - PINDSW-5145: RX_ER counter does not count errors outside frame in MII RX_CLK units precisely
- **Conditions in which failures occur**
  - When the RX_ER error occurs
- **Root cause**
  - Firmware counts RX_ER only once per frame, i.e. only when RX_DV is asserted.
  - For any RX_ER seen when RX_DV is not asserted, firmware does not increment this counter.
- **Work around**
  - Added PHY RX Error Counter Register (0x0E28) for improving RX Error Counter accuracy. Configuring this register will track RX_ERs within a frame precisely using PHY registers.
  - Refer the “Register Exceptions -> RX Error Counter of Port 0 & 1 (0x0301 and 0x303)” section in “Industrial Communications Toolkit->Third Party Protocol Stacks or Customers using their own Stacks->EtherCAT Slave FWHAL->TI EtherCAT Slave Controller Exceptions” page of MCU+ SDK documentation for more details.

NOTE: This issue will not be fixed, workaround is available.

Issues fixed in PRU-ICSS-EtherCAT Firmware Version x.5.0

Lost Link Counter register (0x310/0x311) increments with "2" on every link down event instead of "1"

- **Issue/ Failure Description or state**
  - PINDSW-3120 : The Lost Link Counter register (0x310 and 0x311) increments with "2" on every link down event instead of "1"
- **Conditions in which failures occur**
  - On Port0/1 Link disconnect. The Lost Link Counter register counts the link down events on the Port.
- **Root cause**
  - Set the type of MDIO Link interrupt to Edge in tiesc_pruss_intc_mapping.h file
  - The reason for this issue is the MDIO link interrupt was mapped to PRU as PULSE interrupt. The Pulse nature of MDIO Link interrupt lead to PRU interrupt being set twice. Therefore, the firmware increments the counter twice for every link down event.
  - This change to Pulse mode was done to fix link stability issue in a certain rare condition after multiple link disconnect-connect cycles. The issue led to the Port always remaining closed until we have link disconnect/connect action on the other port.

DC timings are not correct after re-activation of DC

- **Issue/ Failure Description or state**
  - PINDSW-5194 : DC timings are not correct after re-activation of DC
- **Conditions in which failures occur**
  - When DC mode is reactivated, SYNC1 generation is not correct. For example, if SYNC1 is programmed to trigger once per 8 cycles of SYNC0 and before DC was deactivated, there were 4 cycles of SYNC0 after last SYNC1 pulse, then after activating DC, SYNC1 is seen after 4 SYNC0 cycles.
- **Root cause**
  - Internal counters were not being cleared

ESC firmware counts short frames in 0x0302 register

- **Issue/ Failure Description or state**
  - PINDSW-5229 : ESC firmware counts short frames in 0x0302 register
- **Conditions in which failures occur**
  - When both ports are connected and short frame arrives on Port 1
- **Root cause**
  - 0x0302 error counter is being incremented by firmware for short frames. Only 0x030C should be incremented.

Lost Link Counter register (0x310/0x311) is getting initialized to 1

- **Issue/ Failure Description or state**
  - PINDSW-5267 : Lost Link Counter register (0x310/0x311) is getting initialized to 1
- **Conditions in which failures occur**
  - During power up, this counter is updated to 1 even without a physical link loss.
- **Root cause**
  - It is being initialized to 1 by default, even when there is no link drop.

Issues fixed in PRU-ICSS-EtherCAT Firmware Version 5.4.243

RW type commands for address 0x300 and length 0x20 fails

- **Issue/ Failure Description or state**
  - PINDSW-4989 : EtherCAT slave does not update the WKC if a RW command is received for address 0x300 with length of 0x20
- **Conditions in which failures occur**
  - Master sends and FPRW command for address 0x300 and length 0x20
- **Root cause**
  - The firmware process path for this particular scenario is taking longer time to process. This leads to firmware not able to process the command correctly and therefore failing the RW operation.
  - If read permission checks and byte level write permission checks are skipped, firmware is able to process the commands. Write permissions will still be checked at word level. See PINDSW-5135 for more details.