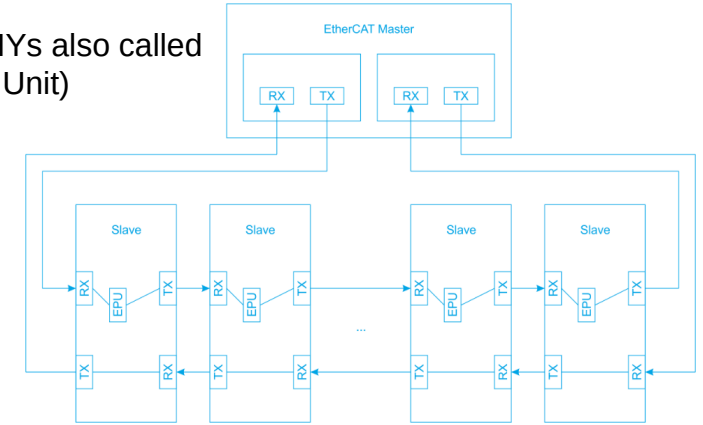
How to debug PHY level issues in EtherCAT networks

Hillman Lin, Gerome Cacho

Terminology

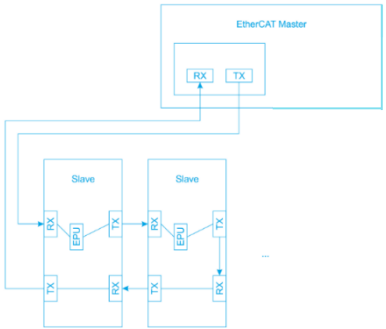
- An **EtherCAT master** device is solely allowed to transmit data across the network
  - Sends a string of data across the bus, eliminating the data collisions of an ethernet system and optimizing speed as a result.

- An **EtherCAT slave** device reads the data going through it, extracts data addressed to it, and inserts its own data in the frame as the frame is moving downstream
  - Normally has at least two PHYs; an Input PHY and Output PHY
  - Input PHY only received data frames and output PHY only send out data frames
  - Input PHY and Output PHY will switch depend on the data path
  - Has ASIC which communicates between the input and output PHYs also called ESC (EtherCAT Slave Controller) or EPU (EtherCAT Processing Unit)

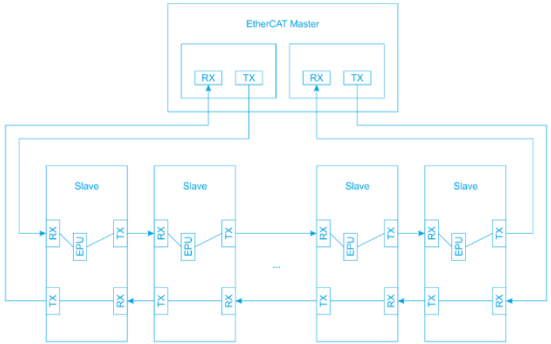


EtherCAT Network signal chains

- **EtherCAT network datapath:**
  - Every cycle, the master sends a single standard Ethernet frame which has multiple EtherCAT datagrams
    - Each frame contains process data for every slave device in the network
  - Slave devices read and write data from/to the frame while it passes through the device
  - Each slave has at least two ports; input and output
    - When data reaches end of chain it will return to the master
    - Ports will switch roles (from Input to Output and vise versa) to allow the frames to pass back up
- **Two types of signal chains for EtherCAT**:
  1. EtherCAT master with single ports
  2. EtherCAT master with multiple ports



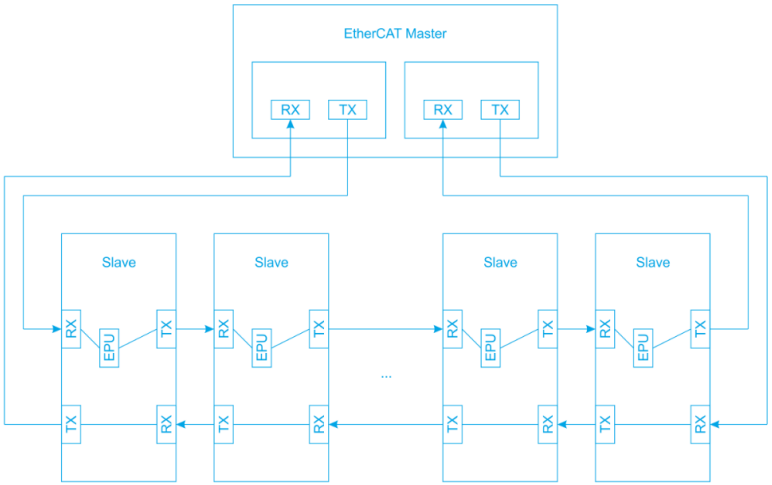
Master with single port



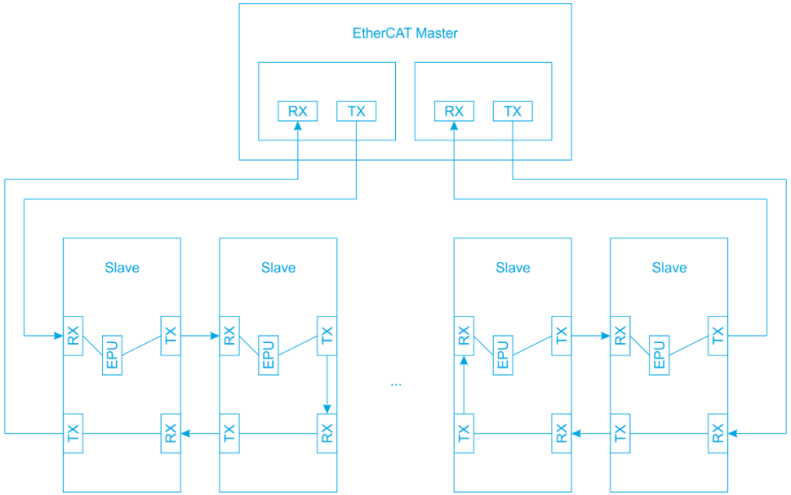
Master with multiple ports

EtherCAT error detection

- When EtherCAT master detects an open wire condition, the last node in a segment (aka drop line) detects the open port and sends the message back to the master
- Open port information will pass up to the EtherCAT master to tell open port is
- Total CRC errors in the systems will also be sent back to the master



Normal condition



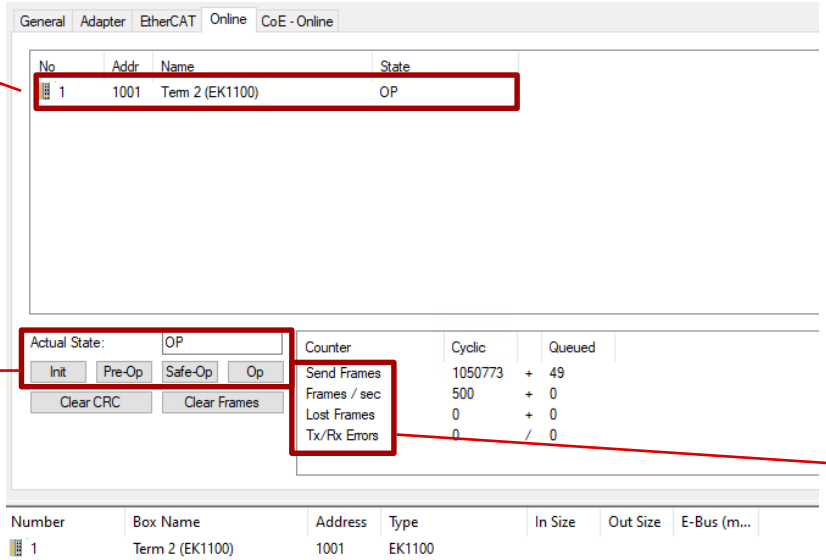
Open port condition

How is TwinCAT software useful to debug PHYs within the EtherCAT network?

- Allows access to EtherCAT frame to determine location of open port and view total CRC errors in system
- Allows programming of EtherCAT master to read specific registers for each EtherCAT slave's PHY
  - Used to check status from PHY standpoint to help determine where errors are being generated in the system

How to read information in TwinCAT

TwinCAT is able to detect one ESC in the EtherCAT network



Different stages:

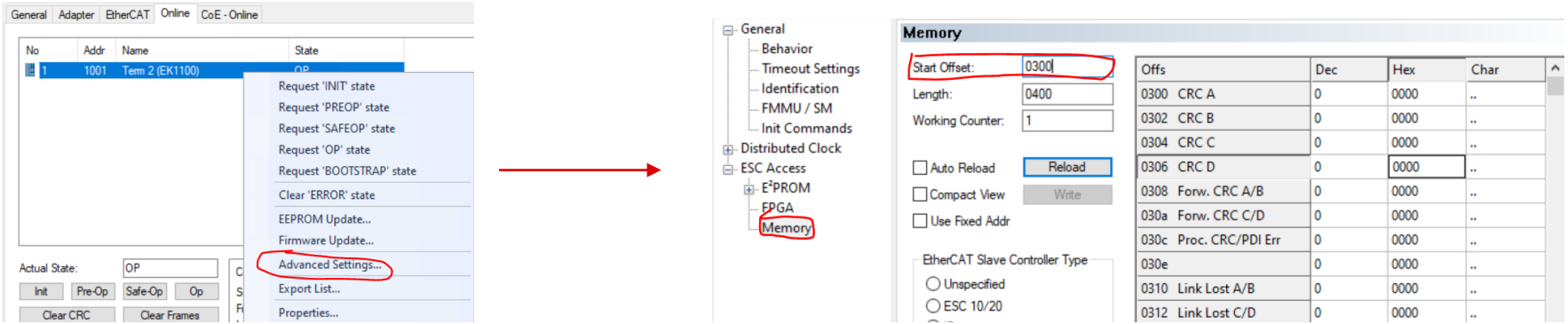
- Init
- Pre-Op
- Safe-Op
- Op

https://infosys.beckhoff.com/english.php?content=../content/1033/ax5000_usermanual/html/Bt_EcBasics_EcStateMachine.htm&id

Total EtherCAT system's send Frame, Lost Frames, and RX_errors

How to read the CRC error on each of the individual ESC

- Right click on the ESC → Advanced Setting→ Memory
- Change “Start Offset” to “0300”
  - Reg 0x300 to 0x312 are all the information for CRC error received and link lost on the ESC
  - These are slave’s registers to determine if errors are being seen by the MAC



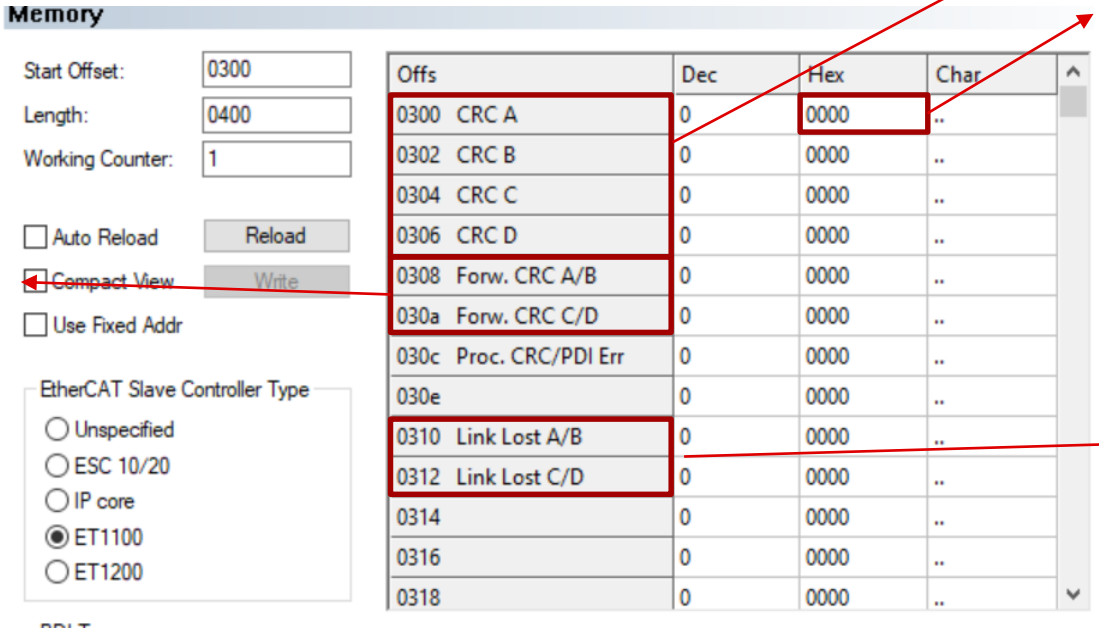
Note: The following examples use TwinCAT registers specific for ET1100 ESC. It is recommended to check with device vendor for analogous registers if using a different device.

Detail reading on CRC error

CRC errors for the ESC

This value will auto-increment with every error. Can be cleared by writing 0x0 to field and clicking “Write”

Forwarded CRC errors: CRC error come from the previous slave



- Ports A and B are the input PHY(s)
- Ports C and CRC D are the output PHY(s)

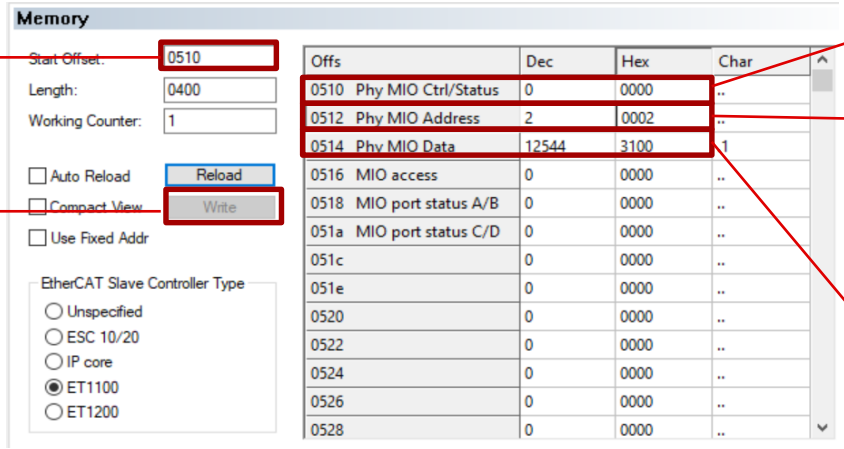
How many link loss events happen within the slave

How to read PHY registers using TwinCAT

- Change “Start Offset” to 0510
- Set TwinCAT (TC) Reg 0x510 to 0x100
- Change TC Reg 0x512 2 LSB’s to the corresponding PHY ID
- Change TC Reg 0x512 2 MSB’s to PHY register you want to read (Reading it this way can only access up to 0x01F registers)
- Click Write
- Result will display in TC Reg 0x514

Start Offset set to “510”

Click write after changing TC Reg 0x510 and 0x512



Write 0x100 to TwinCAT (TC) register 0x510 to read

Writing 0x2 to TC register 0x512 means reading PHY ID 2 (Output PHY) PHY register 0x0

Writing 0x0102 to TC register 0x512 means reading PHY ID 2’s register 0x1

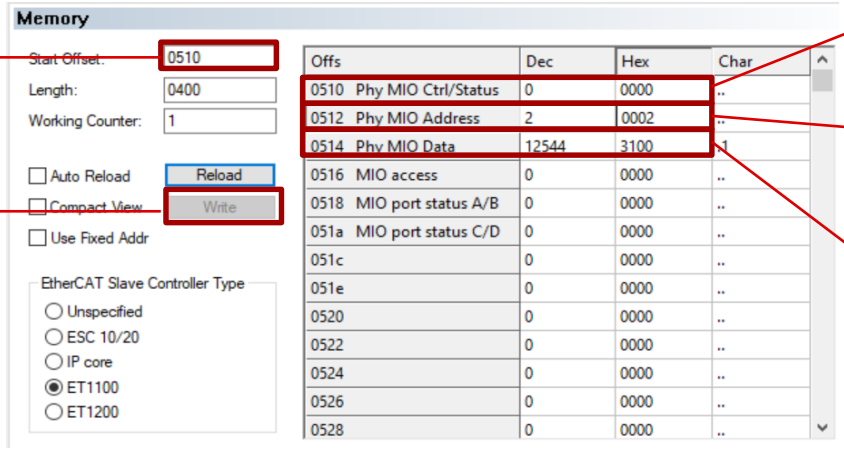
0x3100 is the value of the output PHY ID 2’s register 0x0

How to read PHY registers using TwinCAT

- Change start offset to 0510
- Set TC Reg 0x510 to 0201
- Change TC Reg 0x512 2 LSB's to the corresponding PHY ID
- Change TC Reg 0x512 2 MSB's to PHY address you want to write to
- Type in the value you want to write into TC Reg 0x14 register
- Click Write

Start Offset set to "510"

Click write after you change TC Reg 0x510 and 0x512



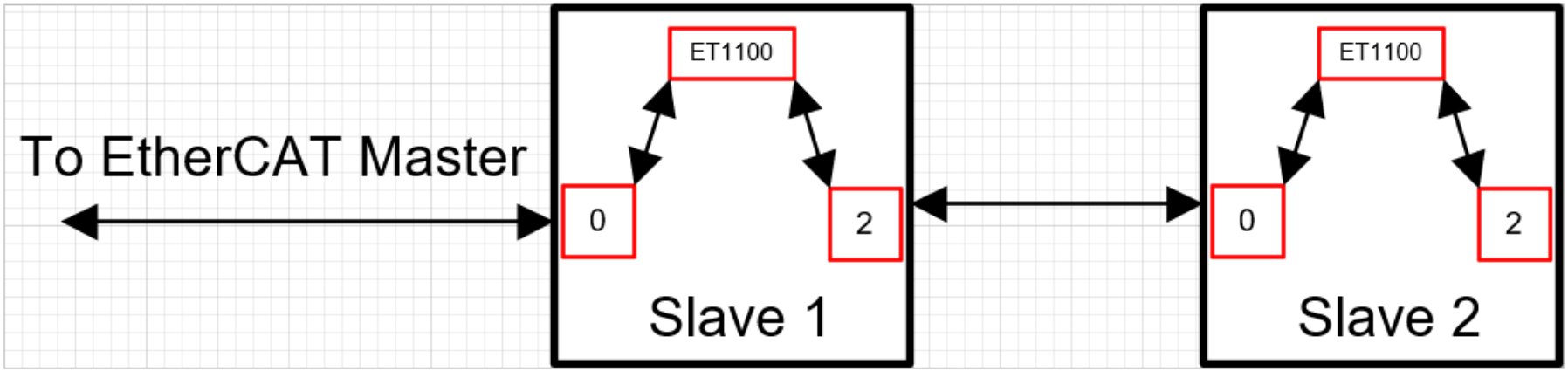
Set 0x201 to TwinCAT (TC) Reg 0x510 to write to PHY registers

Write 0x2 to TC Reg 0x512 to address PHY ID 2's (Output PHY) register 0x0

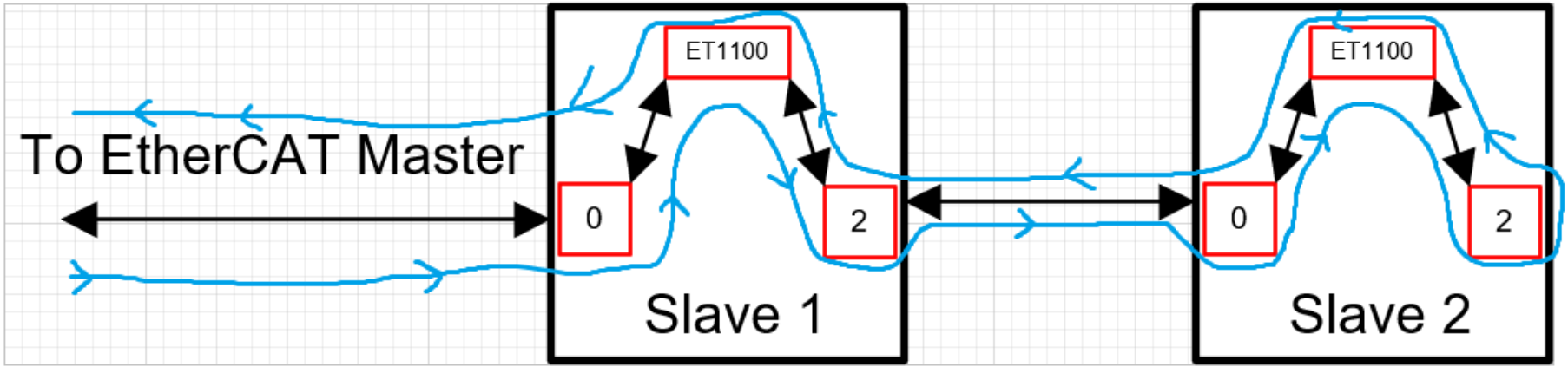
0x3100 is the value we write into PHY register 0x0

How to debug EtherCAT issues

Block Diagram



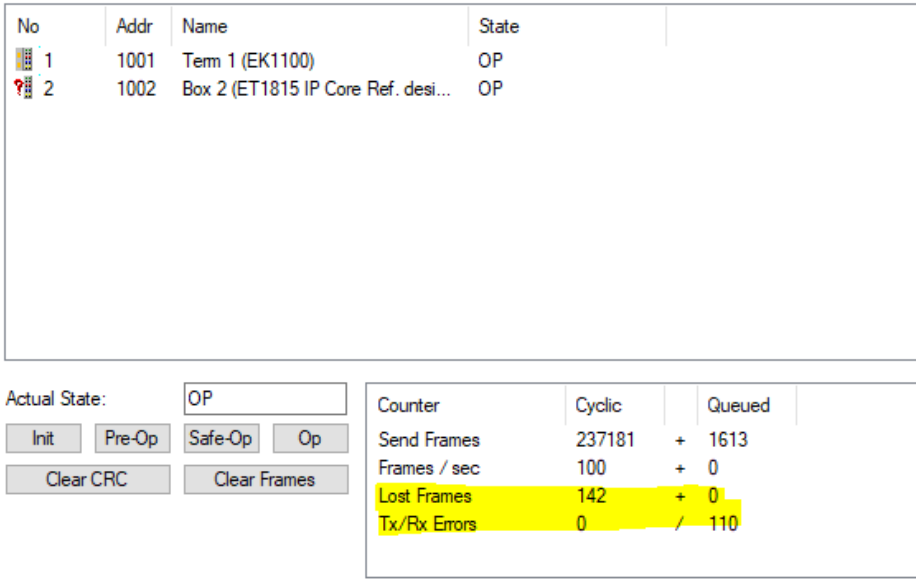
Block Diagram



As there is no link on Slave 2's PHY 2, data will be looped back at ET1100 back up the chain

How to determine there is issues

- Misaligned states
  - Slave in network has unexpected state
  - Correct ESCs should be in “OP” mode
- CRC errors
  - The Slave is showing CRC errors (Reg 0x300-312) on its port
    - Port C = input port
- Tx/Rx errors
  - Indicates errors are coming back to the master in the closed loop



How to determine which PHY to look at

- Each PHY should have a register available to showcase RX errors the PHY detects on the MDI line
  - For example, a majority of PHYs within TI's portfolio have this counter on register 0x15
  - These are errors which the PHY is receiving on its MDI line from the link partner
  - To access a PHYs register 0x15:
    - Set ESC reg 0x510 = 0x100 to initiate read PHY register command AND
    - Set ESC reg 0x512 = 0xYY0X, where YY is the register of interest and X is the port of interest
      - YY being a 2 hex value field means only 0x0-0x1F can get accessed directly. Use extended register access (0xD, 0xE 4 step method) for any extended registers
      - X can be 0 for port A (input) or 2 for port C (output) in a dual PHY slave design

How to determine which PHY to look at

- The following photos are from Slave 1
- This example shows Slave 1's output PHY's RX error counter register is not 0
  - This means that the error was seen on the PHY's MDI RX line

Offs		Dec	Hex	Char
0300	CRC A	0	0000	..
0302	CRC B	0	0000	..
0304	CRC C	65419	ff8b	..
0306	CRC D	0	0000	..
0308	Forw. CRC A/B	0	0000	..
030a	Forw. CRC C/D	1	0001	..
030c	Proc. CRC/PDI Err	0	0000	..
030e		0	0000	..
0310	Link Lost A/B	0	0000	..
0312	Link Lost C/D	0	0000	..

- This photo shows that this PHY has experienced a link drop
  - Reg 0x1[2] is read periodically and is a latch low bit in event of link drop
  - Reading this register will clear the latch and revert to current state

Offs		Dec	Hex	Char
0510	Phy MIO Ctrl/Status	0	0000	..
0512	Phy MIO Address	258	0102	..
0514	Phy MIO Data	30825	7869	ix
0516	MIO access	0	0000	..
0518	MIO port status A/B	0	0000	..

Offs		Dec	Hex	Char
0510	Phy MIO Ctrl/Status	0	0000	..
0512	Phy MIO Address	5378	1502	..
0514	Phy MIO Data	216	00d8	..
0516	MIO access	0	0000	..
0518	MIO port status A/B	0	0000	..
051a	MIO port status C/D	0	0000	..

How to determine which PHY to look at

- The following photos are from Slave 2 in the same chain
- This shows that Slave 2's input PHY had also observed some, but less RX Errors as well as link drop
- This determines that the cause of this issue is the connection between Slave 1 and Slave 2
  - This analysis can be used for longer and more complex chains to isolate for issues

Offs	Dec	Hex	Char
0300 CRC A	65283	ff03	..
0302 CRC B	0	0000	..
0304 CRC C	0	0000	..
0306 CRC D	0	0000	..
0308 Forw. CRC A/B	0	0000	..
030a Forw. CRC C/D	0	0000	..
030c Proc. CRC/PDI Err	0	0000	..
030e	0	0000	..

0510 Phy MIO Ctrl/Status	2	0002
0512 Phy MIO Address	256	0100
0514 Phy MIO Data	30825	7869

Offs	Dec	Hex	Char
0510 Phy MIO Ctrl/Status	2	0002	..
0512 Phy MIO Address	5376	1500	..
0514 Phy MIO Data	3	0003	..

Tips to debug further

- Now that the problem has been identified, some recommendations to probe sensitivities in the chain:
  - Changing slaves around within the chain, or altogether replacing one in an A-B-A swap to isolate if a certain device or combination is causing the issue
  - Changing cables between slaves
  - Ensuring PHYs within Slave components are properly configured for EtherCAT