Application Note

F28P65 + DP83826 EtherCAT Link Down Fault Troubleshooting Methods and Case

Yunjing Wang, Zane Wei *North West China FAE team*

ABSTRACT

This document was translated from a simplified Chinese source. (ZHCAG57)

EtherCAT networks are known for their high real-time performance, deterministic nature, and flexible topology redundancy. Therefore, EtherCAT applications have strict requirements on the establishment speed and stability of their links. This article provides a systematic, in-depth, and efficient troubleshooting guide for the Link Down fault in EtherCAT applications. Based on the F28P65+DP83826 EtherCAT application case, it also presents detailed troubleshooting steps and analysis examples to help quickly locate and resolve the Link Down fault, significantly improving the development and debugging efficiency and field reliability of EtherCAT slave devices.

Table of Contents

List of Figures

List of Tables

Revision History

Version	Date	Author	Notes
1.0	Feb 25th, 2026	Yunjing Wang, Zane Wei	First version

1 EtherCAT Network and Slave Structure

An EtherCAT network consists of one EtherCAT master and several slaves, as shown in Figure 1-1. As a data frame passes through each slave, relevant input and output data are extracted or inserted in real time. The entire process eliminates the need to store, parse, and forward the entire frame as in legacy Ethernet, contributing to a delay of only a few nanoseconds, while the high real-time performance of EtherCAT is based on the establishment speed, stability, and fault reaction capability of the links between slaves.

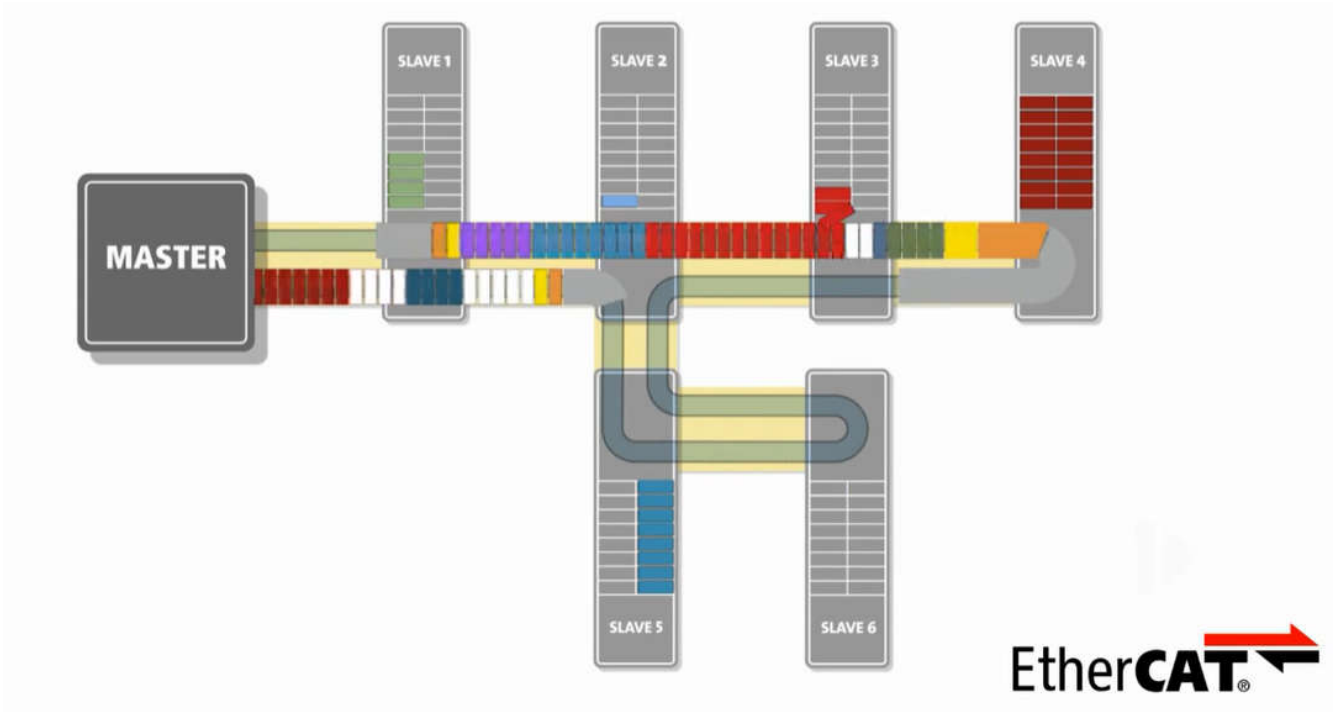

Figure 1-1. EtherCAT Communication Process Diagram

As shown in Figure 1-2, each EtherCAT slave module primarily consists of a slave controller (ESC), a physical layer unit (PHYs, MIIs, RJ45 connectors, and isolation transformers), an external reference clock, and a power supply network. Therefore, the Link Down fault in EtherCAT applications is related to the PHY chip, PHY peripheral design, and ESC configuration within the slave system. This article aims to provide a detailed description of the analysis and troubleshooting methods for the EtherCAT Link Down fault.

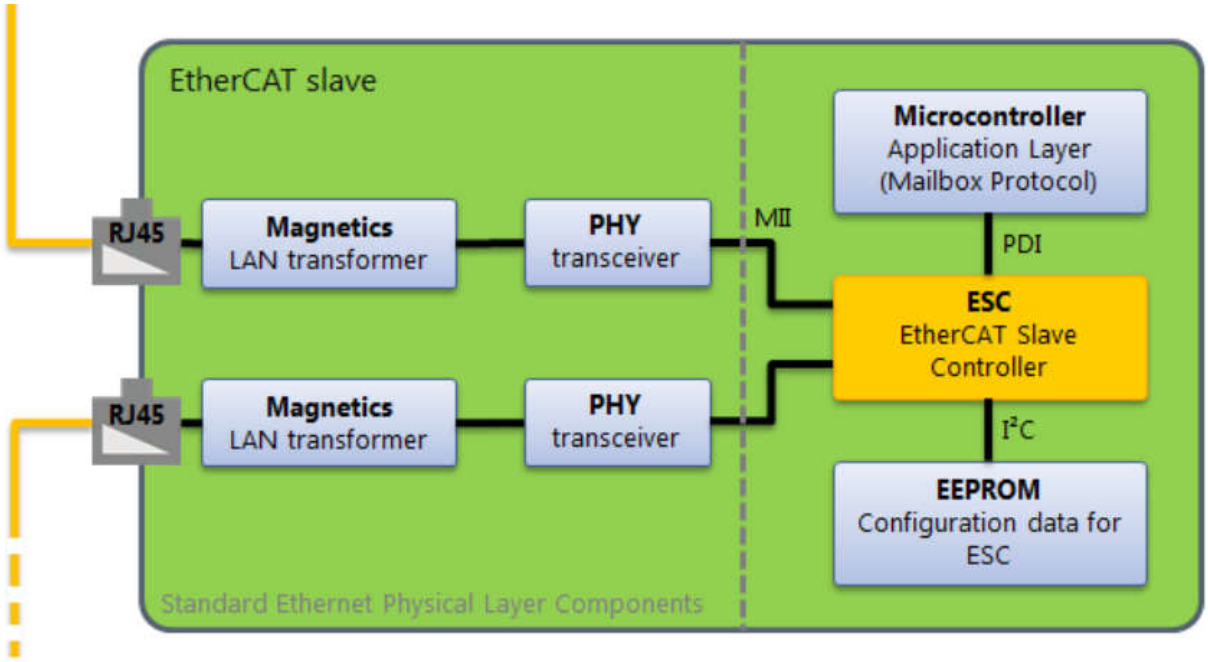

Figure 1-2. Structure of Standard EtherCAT Slave Modules

2 Criteria for EtherCAT Link Multi-Layer Detection

The EtherCAT link diagnosis involves multi-layer detection, and understanding the criteria is the basis for reverse troubleshooting of the Link Down fault in EtherCAT applications. Figure 2-1 illustrates the multi-layer detection structure for the EtherCAT link.

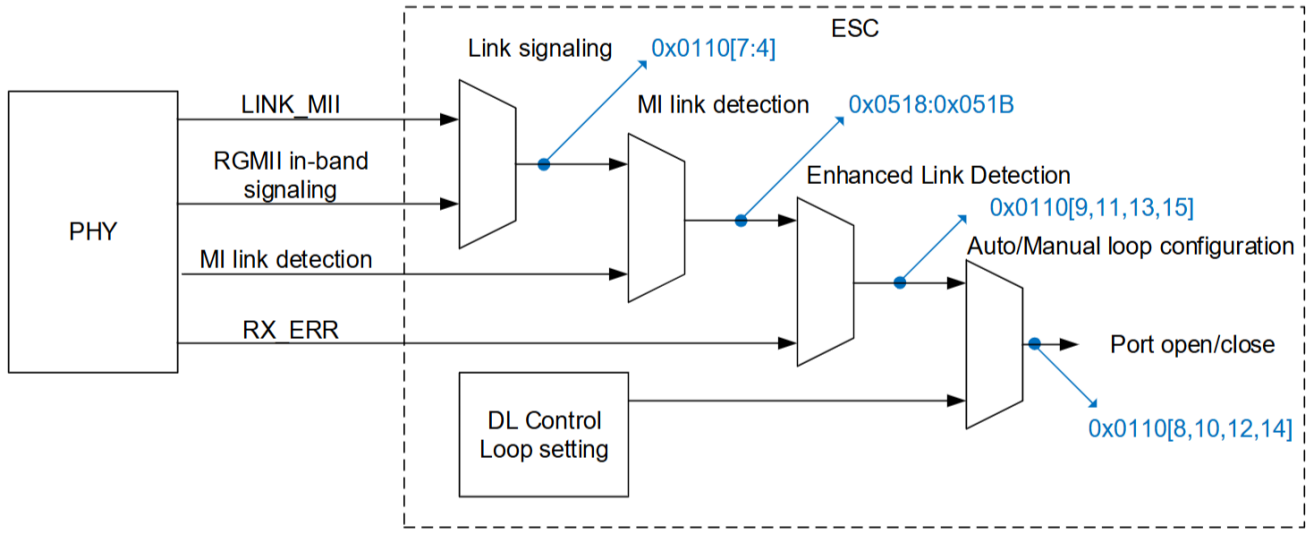

Figure 2-1. Multi-Layer Detection Structure for EtherCAT Link

Layer 1: Physical layer link detection

This is the lowest layer of link detection, where the ESC directly samples the link signal from the PHY: The LINK_MII signal indicates the link status via a dedicated pin provided by the PHY, which drives the LINK_MII signal to an active level after the physical layer link is established. RGMII in-band signaling refers to the transmission of link status instructions via the RXD signal over the RGMII. The LINK_MII signal has higher priority than RGMII in-band signaling.

Layer 2: MI link detection

If the user has enabled the ESC's MI link detection (0x0510 MII Management Control Status register), this layer's detection is performed via the ESC's Management Data Input/Output (MDIO) interface by actively reading the PHY's Basic Status register. The ESC verifies:

1. Proper MDIO communication: Whether read and write operations to the PHY are successful.
2. Completion of PHY auto-negotiation: Whether the Auto-Negotiation Complete bit is 1.
3. Correct link speed and duplex mode: Whether they are set to 100Mbps Full-Duplex.

The results are recorded in the PHY Port Status register (0x0518:0x051B) and affect the final determination of communication status.

Layer 3: Enhanced Link detection

Enhanced Link detection is an active protection mechanism introduced to improve link stability. When Enhanced Link detection is enabled, the ESC monitors the RX_ERR signal from the PHY. When the number of active RX_ERR pulses exceeds 32 within a very short time window (approximately 10µs), the ESC determines that the current link quality is unreliable and performs the following actions:

1. Actively closes the local port loop (even if the physical layer signals are still present).
2. Restarts the peer PHY's auto-negotiation through MDIO to notify the peer of link abnormalities.

Layer 4: Final determination of port status

The "Communication establishment status" (0x0110 Bit 15, 13, 11, 9) and "Port open/closed status" (Bit 14, 12, 10, 8) of the terminal port combine the results from the previous layered detection. Additionally, the port mode

(Auto, Manual, etc.) configured in the **DL Control register (0x0100)** determines how the ESC performs the port open or close operations based on these detection results. **Higher-layer detection results can override the "link good" result of lower-layer detection**. For example, even if the Link_MII signal is active (indicating a physical layer link), the final communication status is asserted to 0 (no communication), and the port is then closed if Enhanced Link detection is enabled and there are a large number of RX_ERR triggers. Alternatively, if the PHY mode configuration does not meet EtherCAT requirements (for example, the copper PHYs on both ends of a link are configured to operate in the Force Full Duplex, 100Base-Tx mode, instead of the Auto-Negotiation mode), the Layer-2 MI link detection will still determine the link as down regardless of the good result of the physical layer LINK_MII detection (indicating a physical layer link).

With an understanding of the multi-layer detection criteria for link status in EtherCAT applications, the user is able to accordingly troubleshoot the EtherCAT Link Down fault using a reverse, layered approach. This approach helps to efficiently identify the root cause of the Link Down fault in EtherCAT applications, which is described in detail in Section 3 of this article.

3 Layered Troubleshooting Methods for EtherCAT Link Down Fault

As described in Section 2, to troubleshoot the Link Down fault in EtherCAT applications, the user can efficiently and clearly identify the root cause by starting from the final outcome based on the multi-layer link detection criteria and working downward layer by layer. This section describes a layered troubleshooting method for the Link Down fault, based on the error registers described in Section 1.2 and the multi-layer EtherCAT link detection criteria described in Section 2.

3.1 ESC DL Port Status Check - Fault Localization

The ESC DL Status register (0x0110:0x0111) and DL Control register (0x0100:0x0103) of the slave help identify the faulty slave and the faulty layer where the faulty slave's link anomaly falls, as shown in Table 2.

The DL Control register is typically configured to Auto mode for all ports, allowing slaves to autonomously open or close their ports depending on the link status. Therefore, if the DL Status register of a non-terminal port shows that the port is closed, it indicates a link status failure for the corresponding slave. Loopback occurs prematurely at the fault location because the current slave link is disconnected.

At this point, check the communication status (Bit [15], [13], [11], [9]) and physical layer link status (Bit [7], [6], [5], [4]) of the DL Status register to identify the port and the layer where the faulty slave's link anomaly occurs:

1. **Physical Link Status = 0, Communication Status = 0**: PHY link is abnormal. Follow Section 3.4 to troubleshoot PHY fault factors.
2. **Physical Link Status = 1, Communication Status = 0**: PHY link is OK, but the ESC's upper-layer communication is abnormal. Follow Sections 3.2 and 3.3 to perform Enhanced Link detection or MI link detection.

Table 3-1. ESC DL Registers and Their Fault Indications

ESC Register Address	ESC Register Name	Register Status and Fault Indication
0x0110:0x0111	**DL Status register**	**Link status and loop status.** **• Bit 15, 13, 11, 9 (Port 3, 2, 1, 0): Communication establishment status.** This status is the final result of link detection, which combines the status results from physical link detection, MI link detection, and Enhanced Link detection. 1 = stable communication, 0 = no communication. **• Bit 14, 12, 10, 8: Port open or closed status.** 1 = port closed, 0 = port open. **If a non-terminal port is closed, it indicates an abnormal link condition for this slave.** **• Bit 7, 6, 5, 4: Physical layer link status.** Directly reflects the physical layer link status. 1 = PHY link OK, 0 = PHY link abnormal. **• Bit 2: Enhanced Link detection enable status.**

Table 3-1. ESC DL Registers and Their Fault Indications (continued)

ESC Register Address	ESC Register Name	Register Status and Fault Indication
0x0100:0x0103	**DL Control register**	**Port control mode configuration**. • Configure each port to operate in one of the following modes: **Auto** (automatically open or close), **Auto Close** (automatically close. A confirmation from the master is required beforehand), **Manual Open** (force open), **Manual Close** (force close). During troubleshooting, verify that the mode is as expected to avoid incorrectly configuring a non-terminal port to operate in Manual Close mode.

3.2 Enhanced Link Configuration Check

If the troubleshooting described in Section 3.1 reveals that the physical link status is good but the communication status is abnormal, first check the Enhanced Link detection: Read the ESC Configuration register (0x0141) to verify whether Enhanced Link detection is enabled. If it is enabled, check the RX error counter of the faulty port (0x0301/3/5/7). If the counter value is not 0 or is increasing, it generally indicates that excessive PHY RX_ERRs are causing abnormal Enhanced Link detection. Follow Section 3.4 to troubleshoot physical layer issues. Alternatively, temporarily disable Enhanced Link detection to observe whether the link recovers stability. If it recovers, focus on identifying the cause of PHY RX_ERRs; otherwise, troubleshoot MI detection-related content according to Section 3.3.

Table 3-2. ESC Configuration & RX_ERR Registers and Their Fault Indications

ESC Register Address	ESC Register Name	Register Status and Fault Indication
0x0141	**ESC Configuration register**	Bit [7:4] indicates whether Enhanced Link detection is enabled for the corresponding Ports 3–0
0x0301/3/5/7	**RX ERR register**	Bit [15:8] displays the RX_ERR count, indicating the number of PHY RX_ERRs

3.3 MI Link Configuration Check

If the Enhanced Link detection described in Section 3.2 shows no anomalies, it is necessary to check the MI link detection status. Check the 0x0510 register to verify whether MI link detection is enabled for the corresponding port. If it is enabled, the 0x0518:0x051B register for the faulty port displays the following diagnostic information.

Bit 3 = 1 (read error): Indicates that the ESC is unable to access the PHY via MDIO. Check PHY address configuration: Check the ESC PHY Address register (0x0512) and compare the displayed value with the PHY's actual address (set by strap pin or register) to verify whether they match.

Bit 1 = 0 (MI link abnormal): Indicates that the ESC is able to read the PHY, but the link status reported by the PHY is not EtherCAT compliant. In this case, the PHY BMSR Status register must be directly read via MDIO to check if the auto-negotiation result, speed, and duplex mode are correct (auto-negotiation must be enabled, and 100Mbps full duplex broadcasting is configured).

3.4 Physical Layer Link Anomaly Troubleshooting

If the ESC register displays 0 for the physical link status (physical layer Link Down), it indicates that the PHY and its peripheral circuitry are causing physical layer link anomalies, which requires hardware and software troubleshooting of the PHY and its peripheral circuitry.

If a physical layer Link Down fault occurs during system power cycling, it is necessary to first check power source quality and PHY power-up timing. A power source that does not meet device requirements (abnormal voltage or excessive ripple) and abnormal power-up and reset timing will result in various possible PHY faults, such as Link Down, unreadable registers, unpredictable error counts, and so on. Therefore, in any case, it is necessary to first verify the power source and power cycle. This operation ensures that the PHY is properly initialized into normal operating mode and avoids confusion between issues caused by the power source or power cycling timing and other suspected faults; otherwise, fault localization might be disrupted.

Reading the critical PHY status information and error register information via MDIO can help to quickly determine the troubleshooting direction. Using the DP83826 as an example, Table 3 lists the key registers for PHY link faults and their fault indications.

Table 3-3. DP83826 PHY's Critical Registers and Their Fault Indications

Register Address	Register Name	Fault Indication
0x0001	**Basic Status**	Check the link status and auto-negotiation capability or status. If the auto-negotiation capability is abnormal, check the 0x00h Basic Control configuration or hardware Strap for any exceptions.
0x0010 **0x0019 bit [15]**	**PHY Status** **Auto MDIX**	Check Auto-MDIX enable, Duplex status, and Speed status EtherCAT requirements: Auto-nego, 100Mbps Full-duplex, Auto-MDIX
0x0467–0468	**Strap Status**	Check whether the PHY hardware Strap configuration matches the specifications outlined in the *DP83826 EtherCAT Application Note* A PHY address configuration error indicates an ESC status register read error, as described in Section 3.3; A PHY operating mode configuration error indicates an ESC status register MI Link exception, as described in Section 3.3.
0x0017	**Bit [5] RMII Mode** **Bit [2:3] FIFO ERR**	Both the DP83826 PHY and ESC require a stable clock, and their reference clocks must share the same source. Clock issues cause data errors at the MII or within the PHY's internal FIFO. A FIFO error typically indicates a CLK fault. Check whether the clocks share the same source and check the 25MHz reference clock for skew (<25ppm) and jitter (<250ps).
0x0015	**RX Error Counter**	The counter for data errors received by the PHY from the MDI, in conjunction with the ESC RX_ERR counter, is used to verify that the external MDI circuitry is not abnormal.
0x0218	**MSE Link Quality**	This register reflects the physical link signal quality. If MSE > 0x33B, troubleshoot peripheral MDI circuits, including the selection and quality of transformers, RJ45 connectors, and network cables, as well as the circuit design.

The F28P65 real-time microcontroller reads the registers of the Ethernet PHY via the Management Data Input/Output (MDIO) module. PHY register management is an important part of Ethernet communication initialization, link status monitoring, and fault troubleshooting. The ESC subsystem within the F28P65 integrates an Ethernet MAC layer containing an MDIO module, also known as the Management Interface (MII) or Serial Management Interface (SMI). This interface is IEEE 802.3 compliant and is used to configure and manage PHY devices connected to the serial management bus.

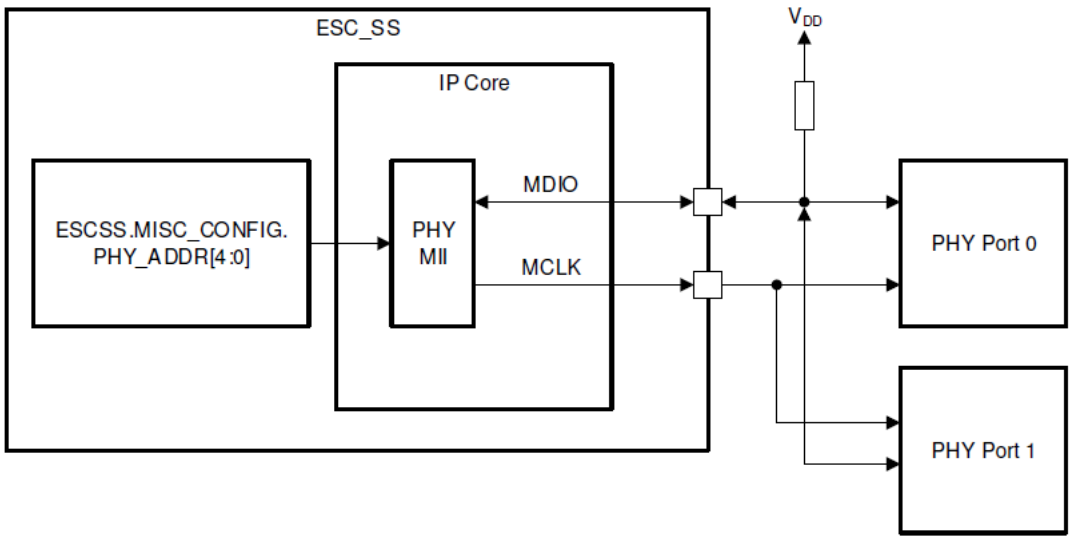

Figure 3-1. F28P65 MDIO Interface

The ESC typically handles Ethernet PHY operations by combining a logical port with a PHY address offset. The PHY address over Ethernet must be equivalent to the logical port number; therefore, PHY addresses 0 and 1 are used. The PHY address offset allows for shifting within any consecutive address range. The ESC module expects PHY address 0 of logical port 0 plus a logical address offset, which can be selected in ESCSS_MISC_CONFIG.PHY_ADDR[4:0].

The following are register-related configurations for the MII. Accessing PHY registers via the MDIO interface primarily involves operating the following registers:

Register Address	Length (Byte)	Description
0x0510:0x0511	2	MII Management Control/Status
0x0512	1	PHY Address
0x0513	1	PHY Register Address
0x0514:0x0515	2	PHY Data
0x0516	1	MII Management ECAT Access State
0x0517	1	MII Management PDI Access State
0x0518:0x051B	4	PHY Port Status

Figure 3-2. MII Registers

If bit 0 of the MII Management PDI Access Status register 0x0517 is not asserted, then the EtherCAT master controls the MI. The EtherCAT master can block PDI control over the MI and force the PDI to release its control over the MI. After power-up, the PDI can take over control of the MI without any master action.

MI read and write timing:

1. Transfer control of the MI to the PDI
2. Write the PHY address to the PHY Address register
3. Write the number of the PHY register to be accessed (0-31) to the PHY Register Address register
4. Write-only instructions: Write the data to be written to the PHY Data register
5. Issue instructions by writing to the control register:
   a. For read instructions, write 1 to the Instruction Register bit 0x0510[8]
   b. For write instructions, write 1 to the Write Enable bit 0x510[0] and write 1 to the Instruction Register bit 0x510[9] simultaneously. Both bits must be written within the same frame
6. Read-only instructions: The read data can be accessed from the PHY Data register. Refer to the code below for specific information:

```
/* MII Management and PHY Addressing defines */
#define ESC_MII_CTRL_STATUS_1_OFFSET         0x0510
#define ESC_MII_CTRL_STATUS_2_OFFSET         0x0511
#define ESC_PHY_ADDRESS_OFFSET               0x0512
#define ESC_PHY_REG_ADDRESS_OFFSET           0x0513
#define ESC_PHY_DATA_OFFSET                  0x0514
#define ESC_MII_ECAT_ACCESS_OFFSET           0x0516
#define ESC_MII_PDI_ACCESS_OFFSET            0x0517
#define PHY_REG_READ_CMD                     0x01
#define PHY_REG_WRITE_CMD                    0x02
#define PHY0_ADDRESS                         0x00
#define PHY1_ADDRESS                         0x80
#define PHY_REG_ADDRESS                      0x0A
```

```
HW_EscWriteByte(0x01,ESC_MII_PDI_ACCESS_OFFSET); // Give PDI access to MII management
HW_EscWriteByte(PHY_REG_ADDRESS,ESC_PHY_REG_ADDRESS_OFFSET); // Set PHY register to read/write
HW_EscWriteByte(PHY1_ADDRESS,ESC_MII_CTRL_STATUS_1_OFFSET); // Read PHY reg
HW_EscWriteByte(PHY_REG_READ_CMD,ESC_MII_CTRL_STATUS_2_OFFSET); // Read PHY reg
HW_EscWriteWord(0x0102,ESC_PHY_DATA_OFFSET); // Set the value to write to register
HW_EscWriteByte(PHY_REG_WRITE_CMD,ESC_MII_CTRL_STATUS_2_OFFSET); // Write PHY reg
HW_EscWriteWord(0x00,ESC_PHY_DATA_OFFSET); // Set the value to write to register, clearing to 0x00
HW_EscWriteByte(0x01,ESC_MII_CTRL_STATUS_2_OFFSET); // Read PHY reg
```

4 F28P65 + DP83826 EtherCAT Link Down Troubleshooting Case

1. **Fault symptom**: Five EtherCAT slaves were tested in a daisy-chain communication configuration, using F28P65 ESC and DP83826 PHY chips. During power cycling tests, the EtherCAT Link Down fault sporadically occurred, causing a premature loopback at the faulty slave and resulting in system communication failures.
2. **Troubleshooting process**: Based on the analysis methods described in Section 3, the troubleshooting process and localization process are illustrated in Figure 6.

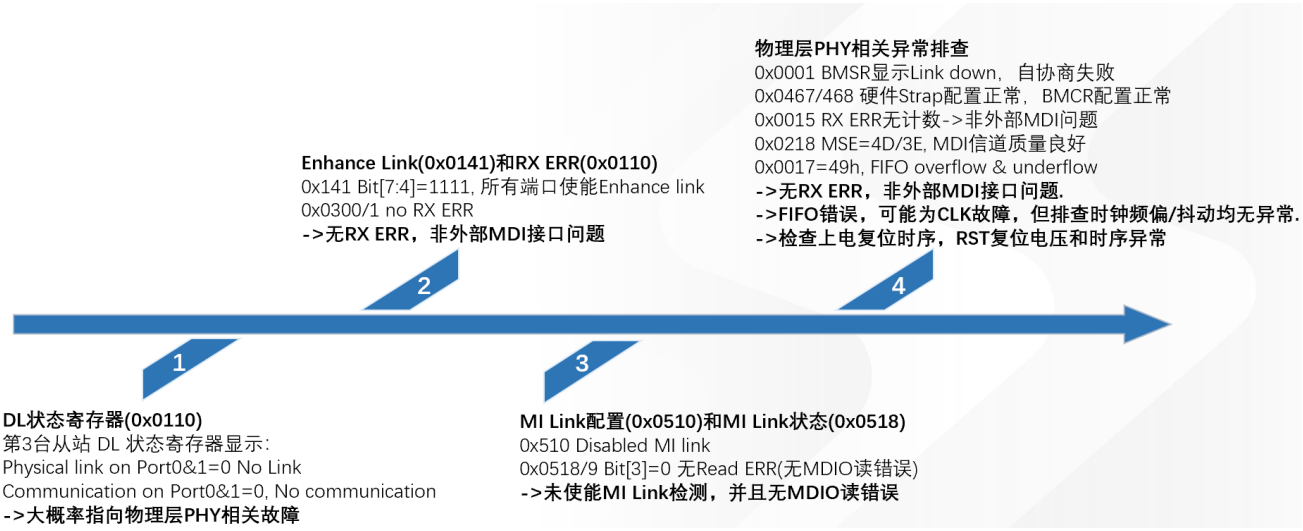

Figure 4-1. Troubleshooting Process and Localization Process Diagrams

3. **Cause of fault**:
   a. The external pull-down resistor for the DP83826 RST pin is high to 4.7kΩ, which is generally sufficient for effective pull-down. However, the internal structure of the device pin must be considered. There is a 10kΩ pull-up resistor inside the DP83826 RST pin and also a 10kΩ pull-up resistor inside the F28P65 GPIO (enabled during application initialization). Therefore, both internal pull-up resistors are connected to 3.3V. When connected in parallel with an external 4.7kΩ pull-down resistor, they create a voltage divider that results in an RST voltage of 1.6V, causing an abnormal RST reset voltage (> 0.8V $V_{IL\ 3V3}$) during power-up.
   b. DP83826 RST timing violation: RST is released before the XI clock settling time of 100us has expired.

Figure 4-2 shows the abnormal RST voltage and RST timing for causes a) and b), respectively.

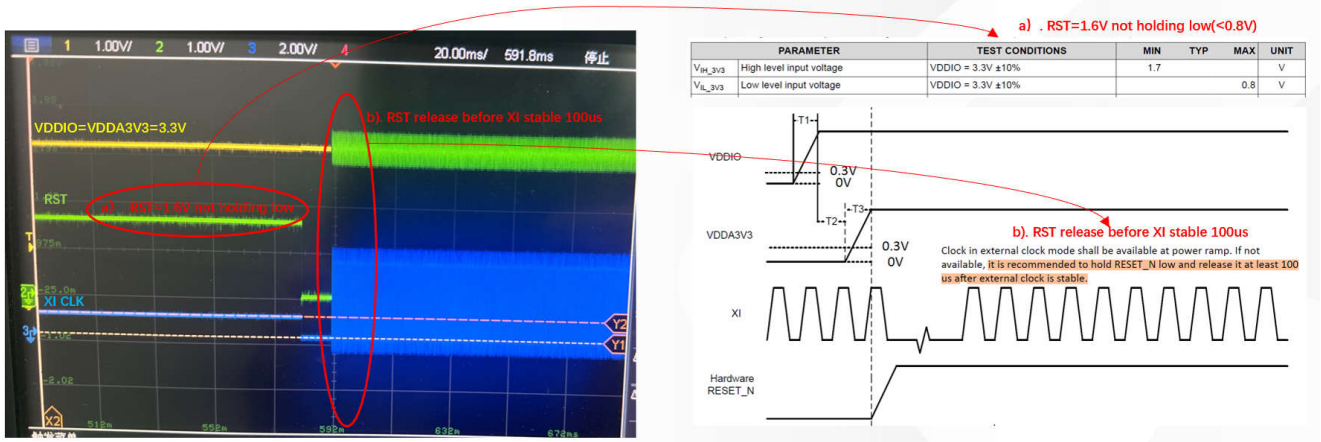

Figure 4-2. Abnormal RST Voltage and RST Timing

Optimize the RST external pull-down resistor to 1kΩ and increase the release delay to release RST until 100us after the XI CLK stabilizes (approximately 10ms in the following figure. See the difference ΔX between the RST release time and XI power-up time shown in the waveforms). After meeting the timing requirements specified in the DP83826 manual, apply the RST voltage and reset timing shown in Figure 4-3. Then the Link Down fault can be resolved.

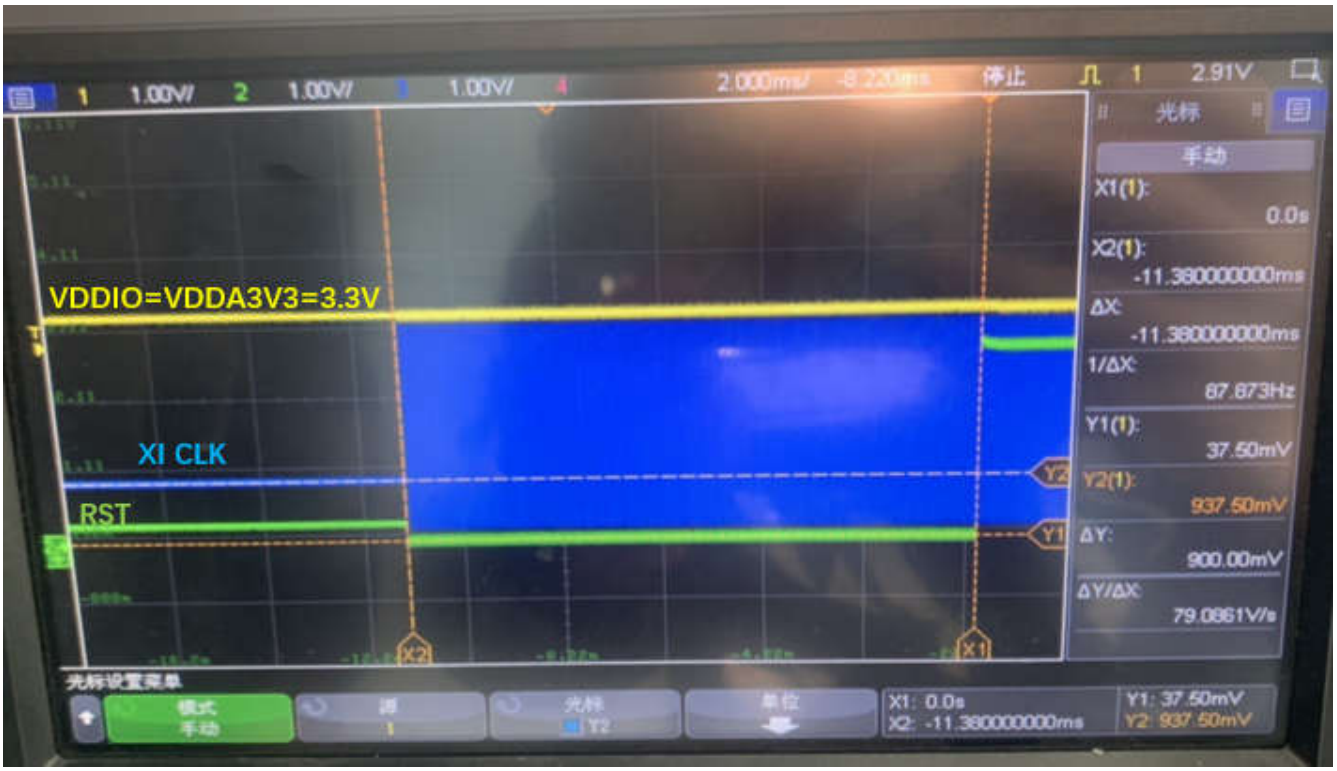

Figure 4-3. Optimized RST Voltage and Timing

This case also verifies that abnormal timing and voltage can cause various possible faults to the PHY, such as Link Down, unreadable registers, unexpected error counts, and so forth, as described in Section 3.4. Therefore, it is necessary to verify the power source and power cycling timing before troubleshooting the PHY, avoiding confusion between issues caused by the power source or power cycling timing and other suspected faults; otherwise, fault localization might be disrupted.

5 Summary

Stable and reliable link status is critical to an EtherCAT application, and this article provides a detailed description of troubleshooting methods for the Link Down fault in EtherCAT applications. Using the F28P65 + DP83826 EtherCAT implementation as an example, this article describes a detailed troubleshooting process, along with fault indications of critical register information, and illustrates a real-world fault analysis case to share detailed debugging guidance and analysis methods for the Link Down fault in EtherCAT applications.

6 References

1. EtherCAT_Diagnosis_For_Developers
2. ethercat_esc_datasheet_sec1_technology_2i3
3. ethercat_esc_datasheet_sec2_registers_3i0
4. ethercat_et1100_datasheet_v2i1
5. F28P65 TRM
6. DP83826 Troubleshooting Guidance
7. DP83826 Data sheet
8. How and Why to Use the DP83826 for EtherCAT® Applications (Rev. C)

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025