TwinCAT控制系统中

EtherCAT 诊断和错误查找

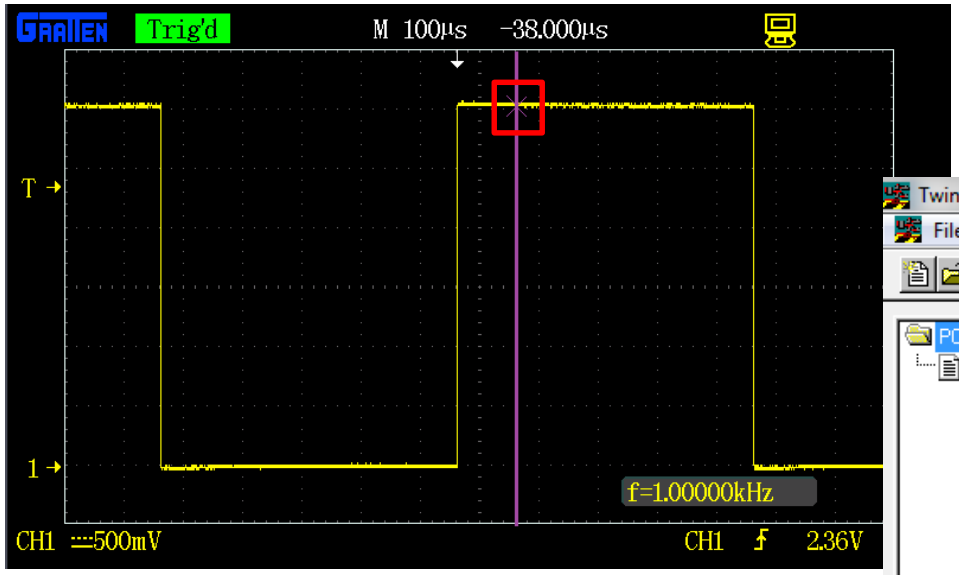
04-Jul-2017

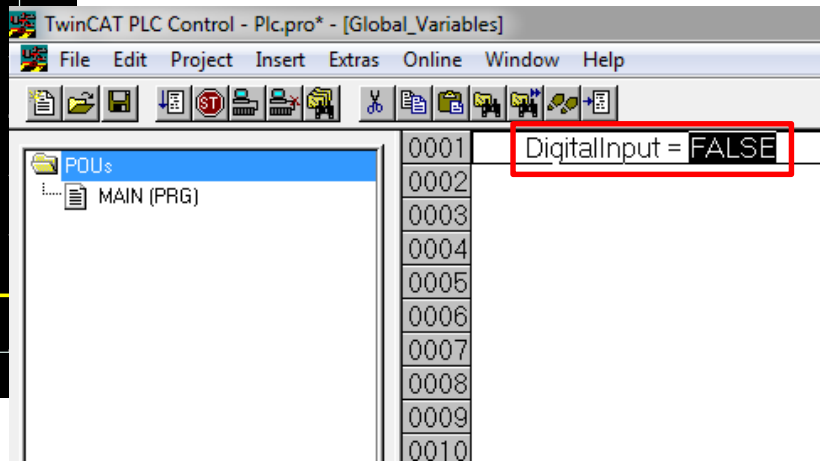
EtherCAT 错误是怎么检测到的?

EtherCAT 通常具有“即扫即用”的特性，扫描网络、启动系统，所有组件就运行起来了。绝大多数情况下不需要设置任何参数。

发现EtherCAT网络的通讯问题，典型的情况是因为以下原因：

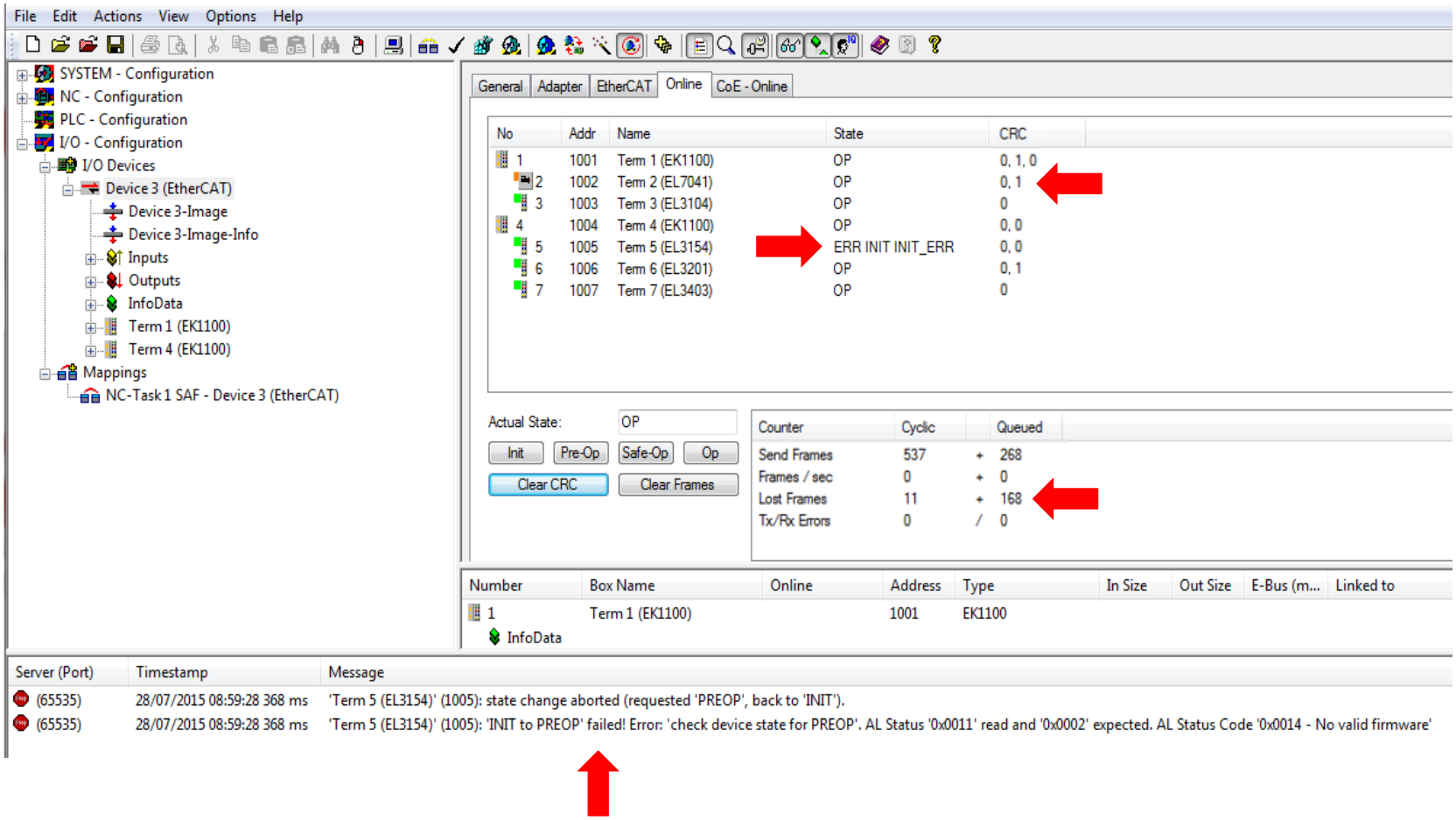
1. 机器停止工作，或者控制项目 (PLC 程序, NC 任务…) 接收到的数据值明显与实测值不一致





EtherCAT 错误是怎么检测到的?

2. 在TwinCAT项目中显示了错误（典型的情况是在Master Online页面或者TwinCAT Logger）



EtherCAT 网络中的错误类型

能影响到EtherCAT网络的错误分为2大类：

1. 硬件错误

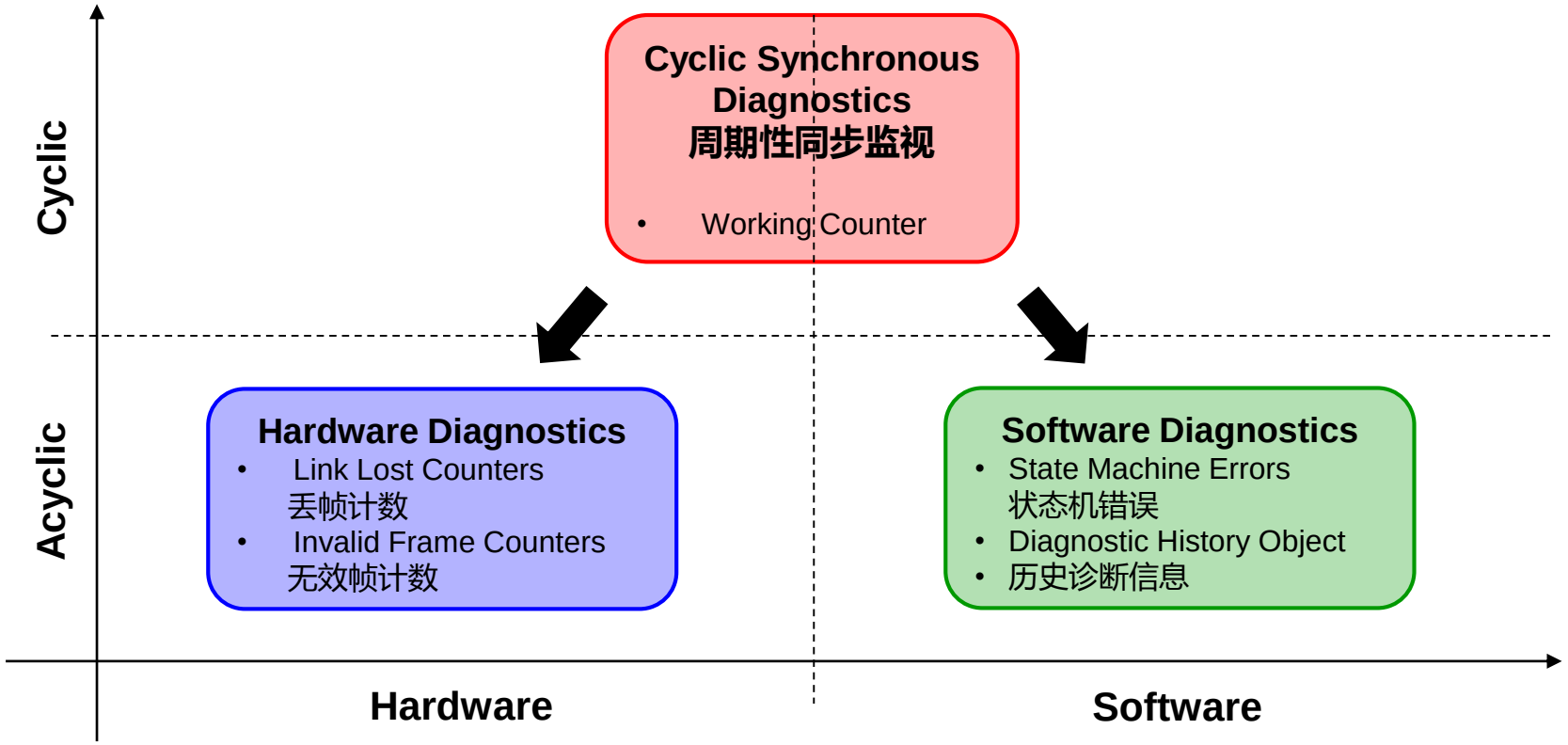
- 物理介质中断：数据帧无法达到所有从站，或者根本不能回到主站。
- 信息破坏： 数据帧到达网络中的所有从站并回到了主站，但有一些内容改变了。

2. 软件错误

- 启动时从站达不到OP状态，因为来自主站的一个或多个初始化的命令被检测到不正确
- 由于在操作过程中检测到错误，一个正常工作的从站突然脱离OP状态

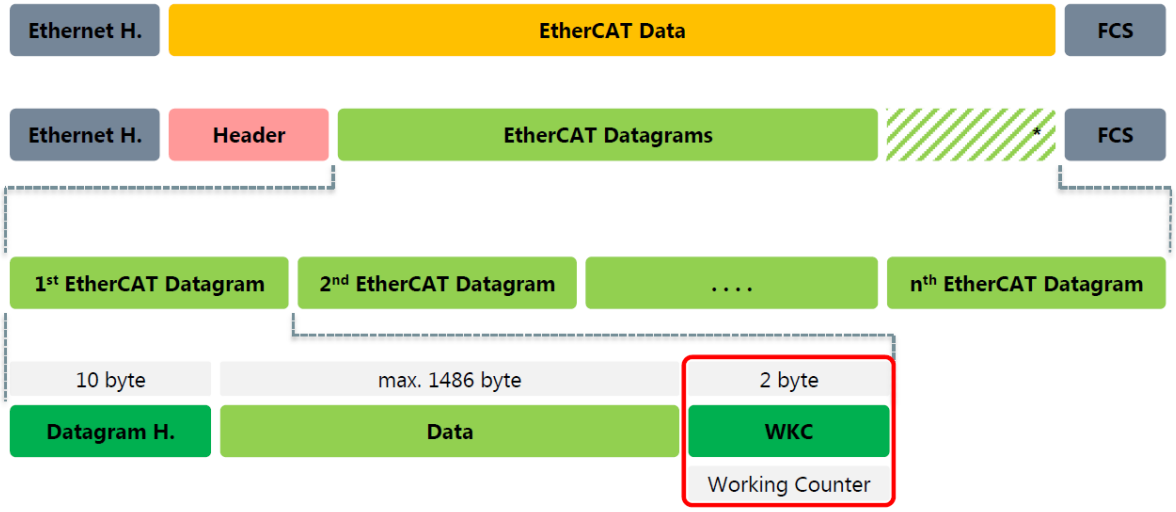
EtherCAT 网络中的错误类型

EtherCAT 分别从硬件和软件层面提供了丰富的**诊断信息，**
诊断信息可以按下图归类：



周期性同步诊断

周期性同步诊断 – Working Counter



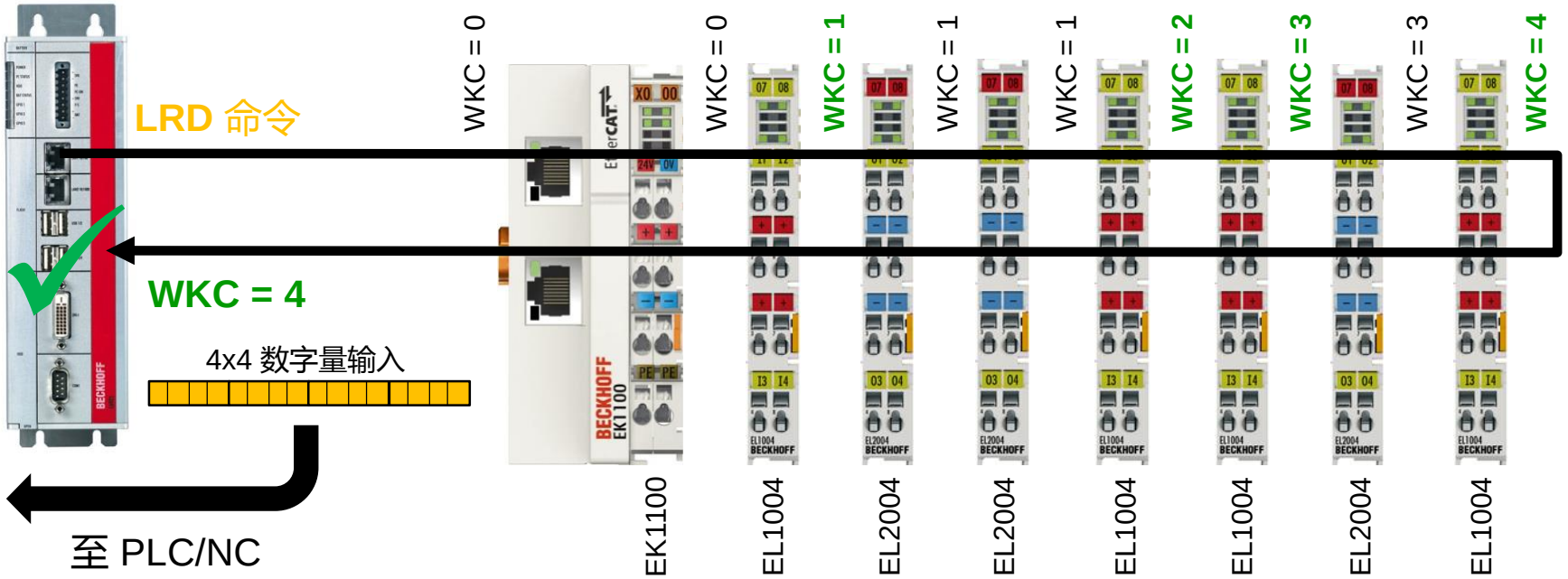
一个EtherCAT数据帧中的每个数据报文都以一个16位的**Working Counter** (WKC)结尾，WKC每经过一个数据报文中指定的从站就会递增，递增的原则如下：

- Read-only 命令(xRD): 如果从站内存可读，则 WKC+1.
- Write-only 命令(xWR): 如果从站内存可写，则 WKC+1.
- Read+Write 命令(xRW): 如果从站内存可读 WKC+1，可写则 WKC+2 (i.e. 如果读写都成功则 WKC+3).

周期性同步诊断 – Working Counter

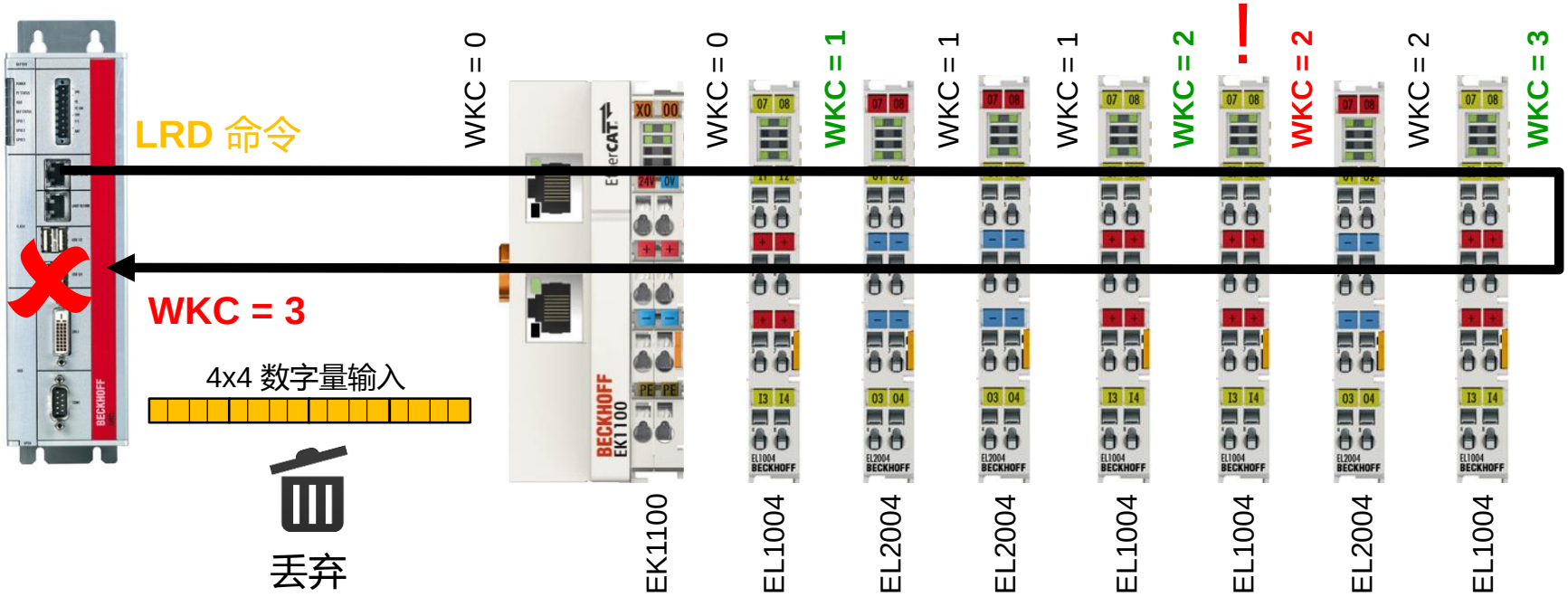
主站检查返回的每个数据报文的WKC值

- 返回主站的数据报文中的WKC 当前值 = 预期值 → **WKC 有效**
  - 数据报文中的输入数据转发给控制程序(PLC, NC, …)



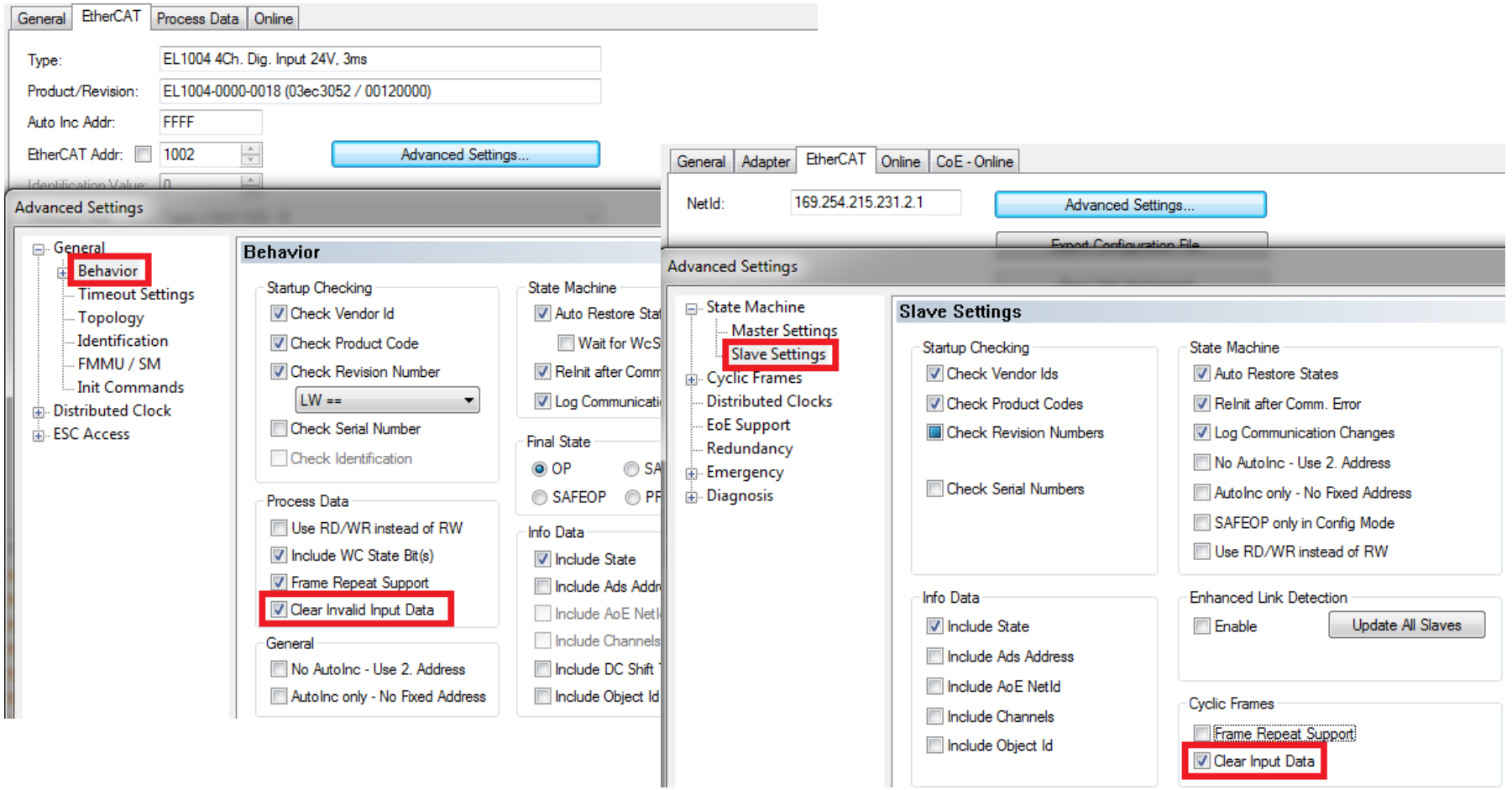
周期性同步诊断 – Working Counter

- 返回主站的数据报文中的WKC 当前值 ≠ 预期值 → **WKC 无效**
  - ➢ 数据报文中的输入数据被丢弃 (PLC/NC 使用旧的数据)



周期性同步诊断 – Working Counter

在最新版的TwinCAT 3.1中，可以把单个从站（左图）或者全部网络从站（右图）配置为：主站把 无效的输入数据全部置 0 而不是废弃之。



周期性同步诊断 – Working Counter

Working Counter 相关信息:

☹ 很少能检测到具体是哪个或者哪些从站没有成功处理数据报文
（数据报文中指定地址的从站没能成功增加数据报文的WKC）

☹ 不会报告错误原因（从站没有成功增加WKC，有可能因为物理上数据帧就没有到达，或者因为它不在OP状态，或者其它可能的原因）

☺ WKC与数据帧同步到达，主站以最快的速度作出故障响应，立即向控制程序或者用户提供一个故障标记位。

☺ WKC的有效标记位通常还需要与硬件或者软件诊断信息同时使用，这些诊断信息可以由主站发送命令读取回来，并帮助定位故障位置和故障原因。

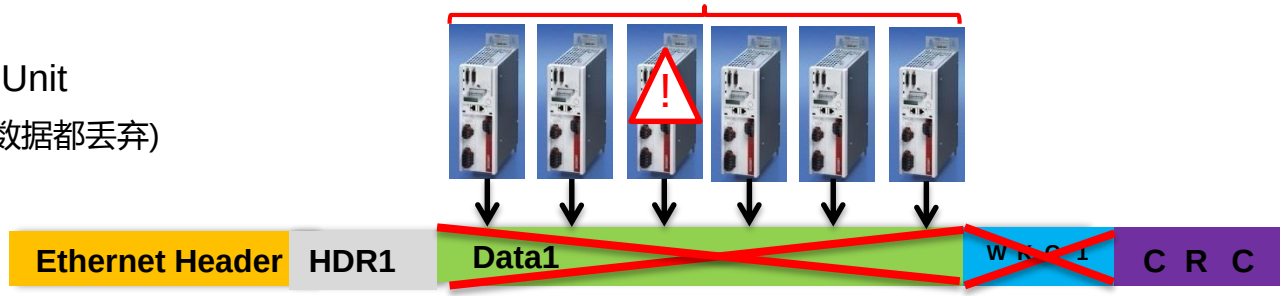
Working Counter 的主要目的是立即检测出通讯错误， (i.e. 为了主站快速做出反应), 而不是为了精准分析通讯错误

Working Counter 和 Sync Units（WKC和同步单元）

Sync Units（同步单元） 把从站分为不同的组，每个组使用不同的数据报文，所以每个同步单元具有独立的 WKC

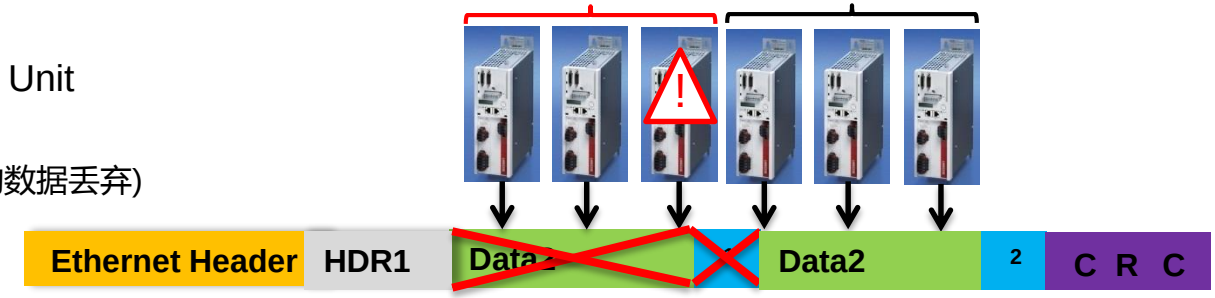
a) 使用 1 个同步单元 Sync Unit

(错误的 WKC → 6个伺的的数据都丢弃)



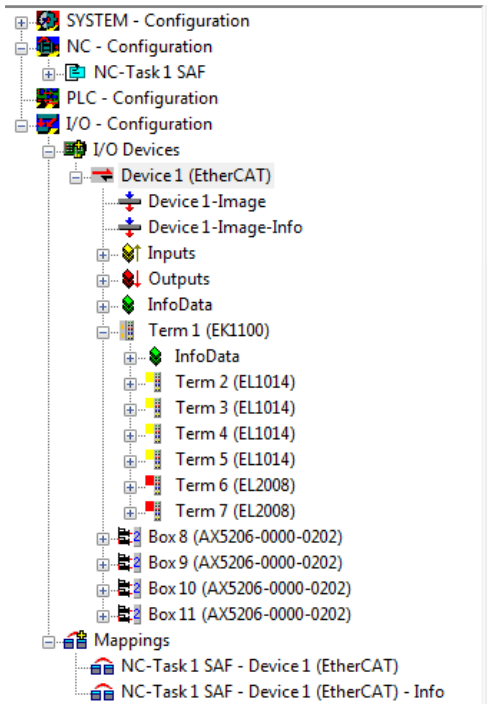
b) 使用 2 个同步单元 Sync Unit

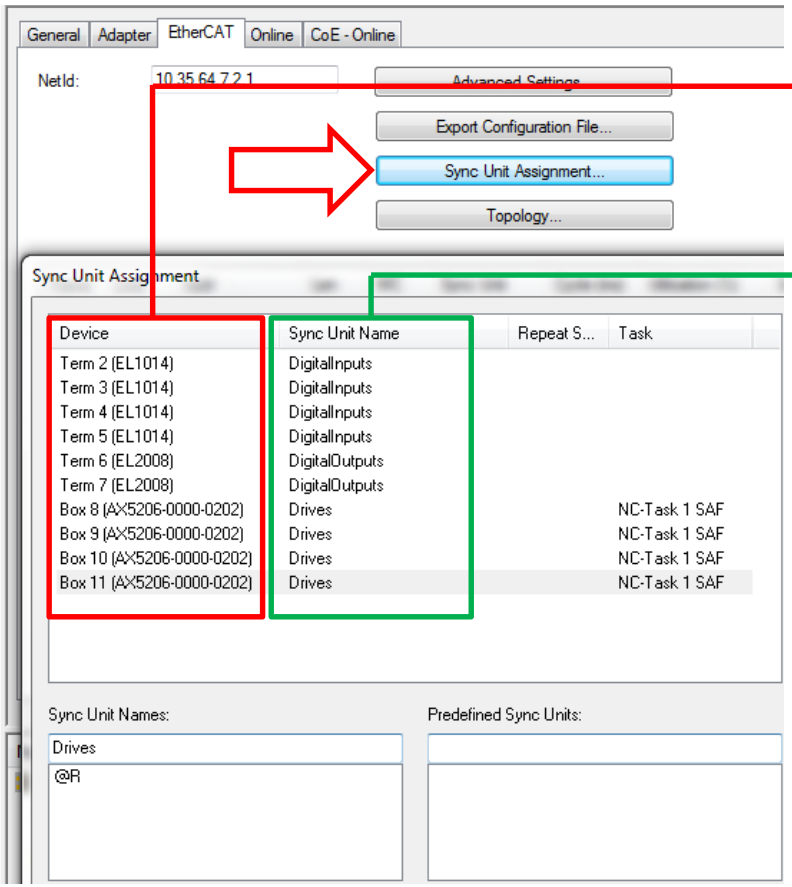
(错误的 WKC → 3个伺的的数据丢弃)



为了**优化**的目的，Sync Unit 的配置是一个**可选**的步骤。关于如何分配同步单元，并没有一个黄金法则，因为这是由项目决定的：通常来讲，工作时紧密相关的从站应该分配到同一个同步单元，而功能相对独立的从站应该放到独立的同步单元。

在 TwinCAT 中指定 Sync Unit





List of slaves.

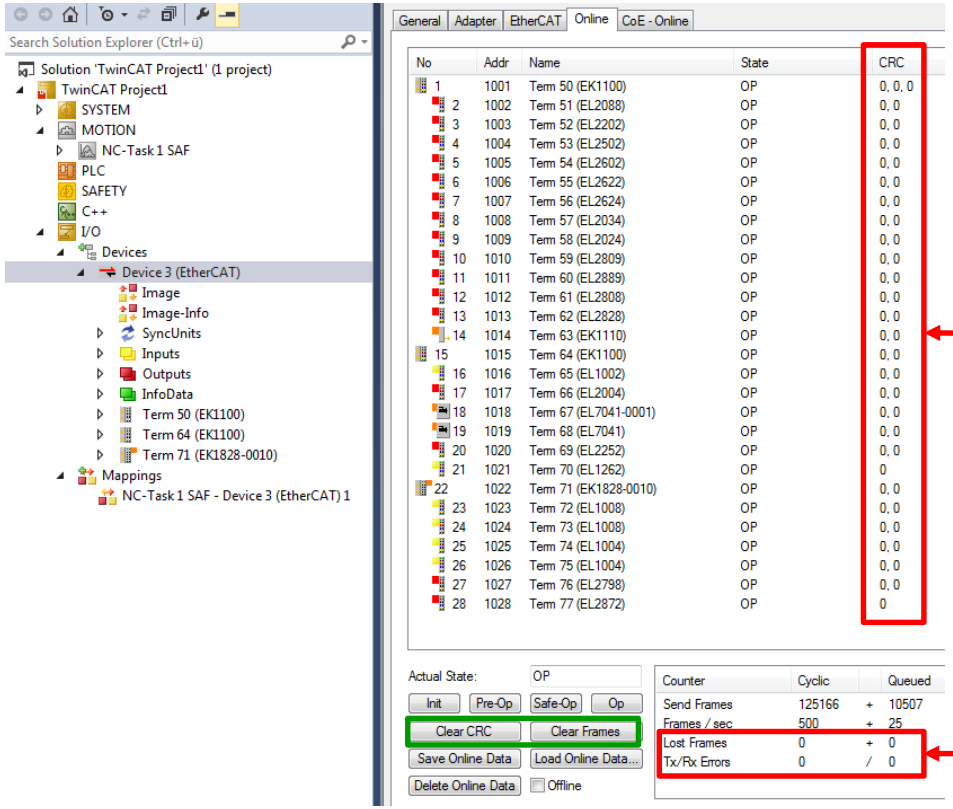
Sync Unit assignment (configurable by user).

TwinCAT为不同的Sync Units（同步单元）创建独立的数据报文，每个报文拥有自己 Working Counter 计数器

Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
0	NOP	0x0000 0x0900	4			2.000			
0	ARMW	0xfff9 0x0910	4			2.000			
0	LRD	0x09000000	1			2.000			
0	LRD	0x01000000	2	4	DigitalInputs	2.000			
0	LWR	0x01000800	2	2	DigitalOutputs	2.000			
0	LRW	0x01001000	80	12	Drives	2.000			
0	BRD	0x0000 0x0130	2	11		2.000	0.87 0.88	195 / 17.52	1

硬件 诊断

硬件状态速览



主站的 **Online** 页面，提供了诊断网络硬件问题的最佳方法

从站硬件错误

主站硬件错误

系统上电时候可能产生硬件错误，并在TwinCAT中累加，而通常这些错误并不代表有问题。所以，查找可能的硬件问题时，建议清除程序启后的所有错误计数，这样才能监视到运行过程中发生的实际错误次数。

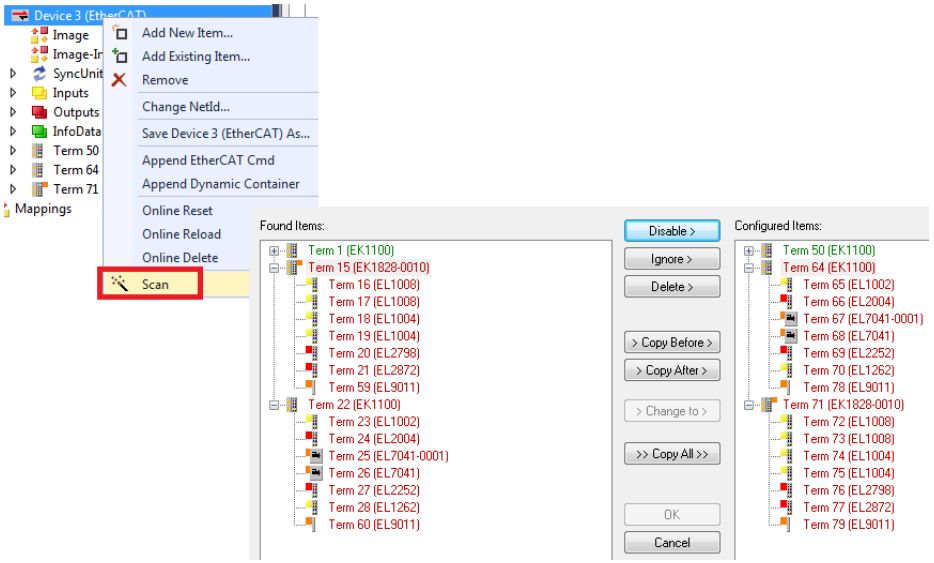
初步检查 – 拓朴错误

VPRS = **V**endor ID, **P**roduct Code, **R**evision Number, **S**erial Number: 网络启动时主站扫描到的拓朴结构与TwinCAT 配置文件的朴拓结构不符。此类错误的原因，可能只是简单地插错网线。

No	Addr	Name	State
1	1001	Term 50 (EK1100)	OP LNK_MIS C
2	1002	Term 51 (EL2088)	OP
3	1003	Term 52 (EL2202)	OP
4	1004	Term 53 (EL2502)	OP
5	1005	Term 54 (EL2602)	OP
6	1006	Term 55 (EL2622)	OP
7	1007	Term 56 (EL2624)	OP
8	1008	Term 57 (EL2034)	OP
9	1009	Term 58 (EL2024)	OP
10	1010	Term 59 (EL2809)	OP
11	1011	Term 60 (EL2889)	OP
12	1012	Term 61 (EL2808)	OP
13	1013	Term 62 (EL2828)	OP
14	1014	Term 63 (EK1110)	OP
15	1015	Term 64 (EK1100)	INIT VPRS INIT_ERR LNK_ADD C
16	1016	Term 65 (EL1002)	INIT VPRS INIT_ERR
17	1017	Term 66 (EL2004)	INIT VPRS INIT_ERR
18	1018	Term 67 (EL7041-0001)	ERR INIT VPRS INIT_ERR
19	1019	Term 68 (EL7041)	ERR INIT VPRS INIT_ERR
20	1020	Term 69 (EL2252)	INIT VPRS INIT_ERR
21	1021	Term 70 (EL1262)	INIT VPRS INIT_ERR
22	1022	Term 71 (EK1828-0010)	INIT NO_COMM
23	1023	Term 72 (EL1008)	INIT NO_COMM
24	1024	Term 73 (EL1008)	INIT NO_COMM
25	1025	Term 74 (EL1004)	INIT NO_COMM
26	1026	Term 75 (EL1004)	INIT NO_COMM
27	1027	Term 76 (EL2798)	INIT NO_COMM
28	1028	Term 77 (EL2872)	INIT NO_COMM

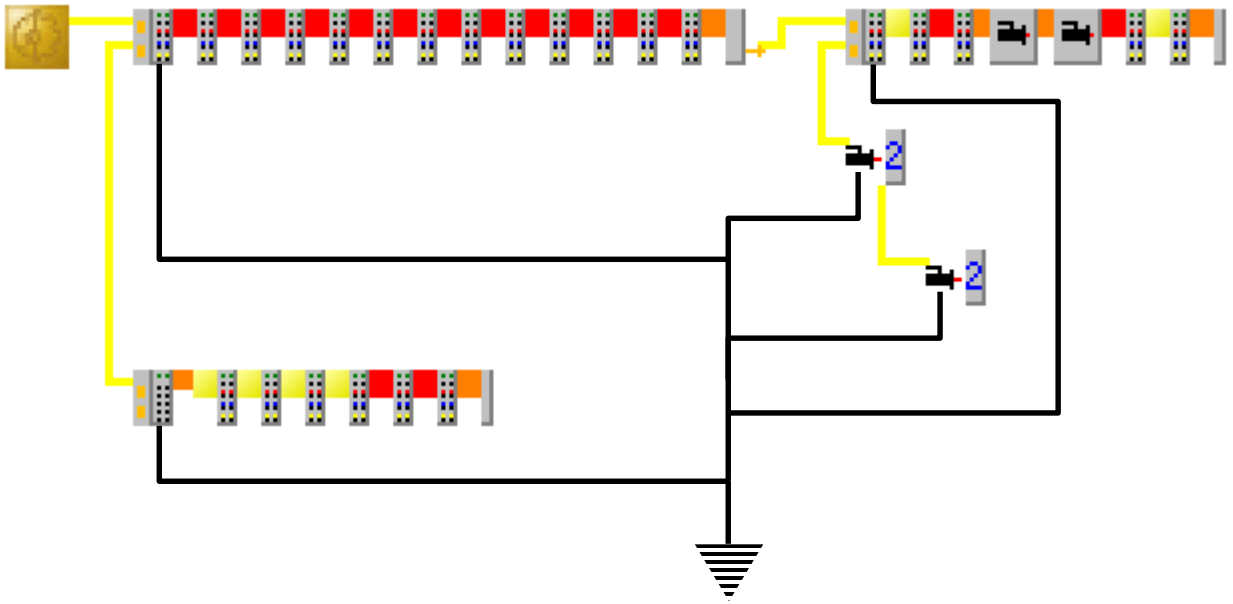
可能的原因是**丢失**网络设备或**错误**的网线**顺序**.

重做网络扫描（在Config模式），比较扫描到的拓朴 (左图) 和当前TwinCAT配置文件的topology (右图)：



初步检查 – 接地和屏蔽

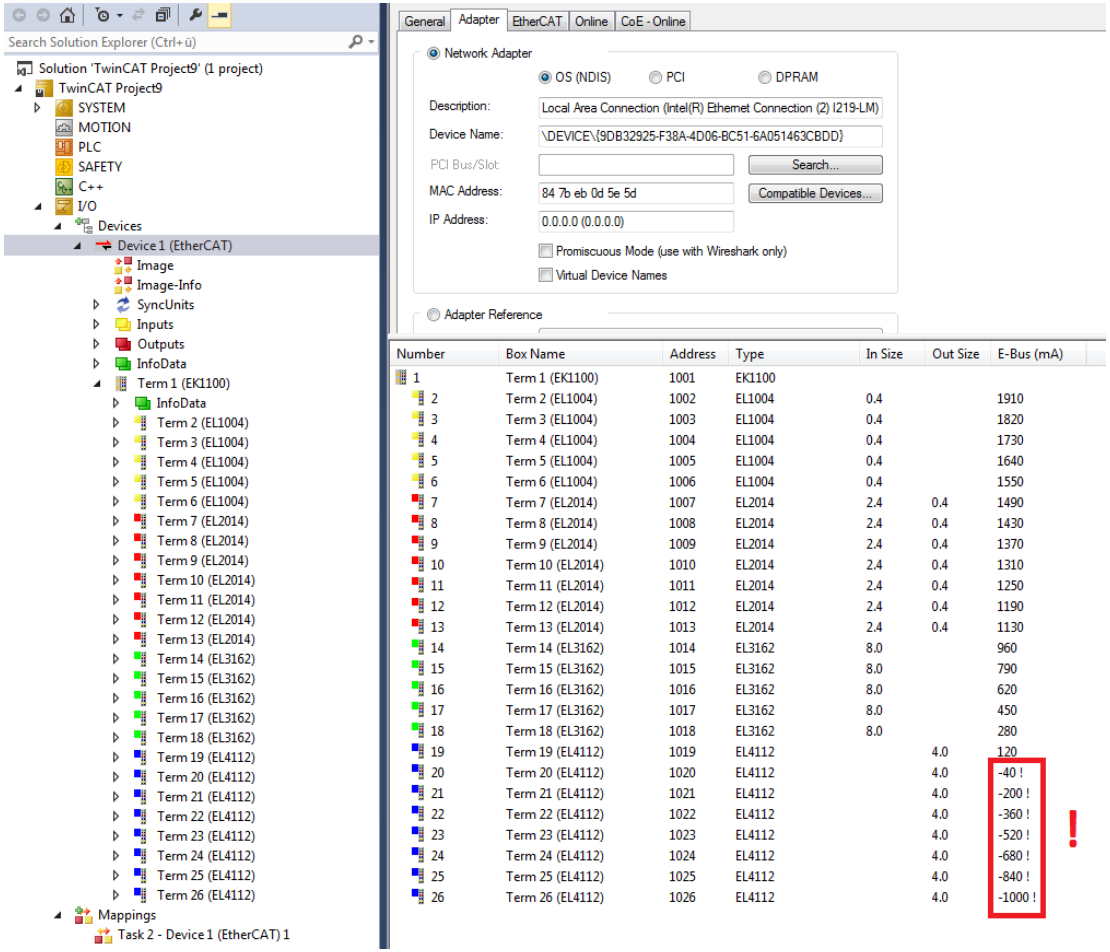
发生硬件错误时，建议必须检查EtherCAT网络设备是否共用同一个**接地**（因为不同接地之间的环流可能导致数据损坏）



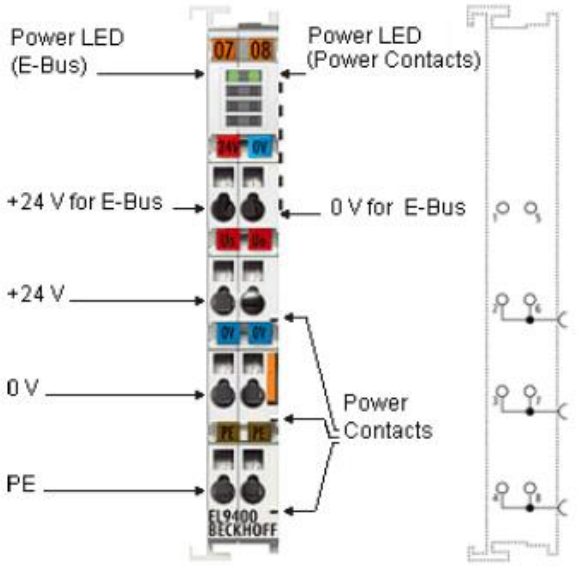
关于EtherCAT网络安装的更完整、全面的描述信息，请参考ETG文档 **ETG.160** “EtherCAT Installation Guidelines”，可以从ETG官网www.ethercat.org.下载

初步检查 – EBUS 电流

每个 EL 模块都会消耗若干 **EBUS** 电流，硬件故障发生时，建议必须检查每个I/O组的可用Ebus电流。

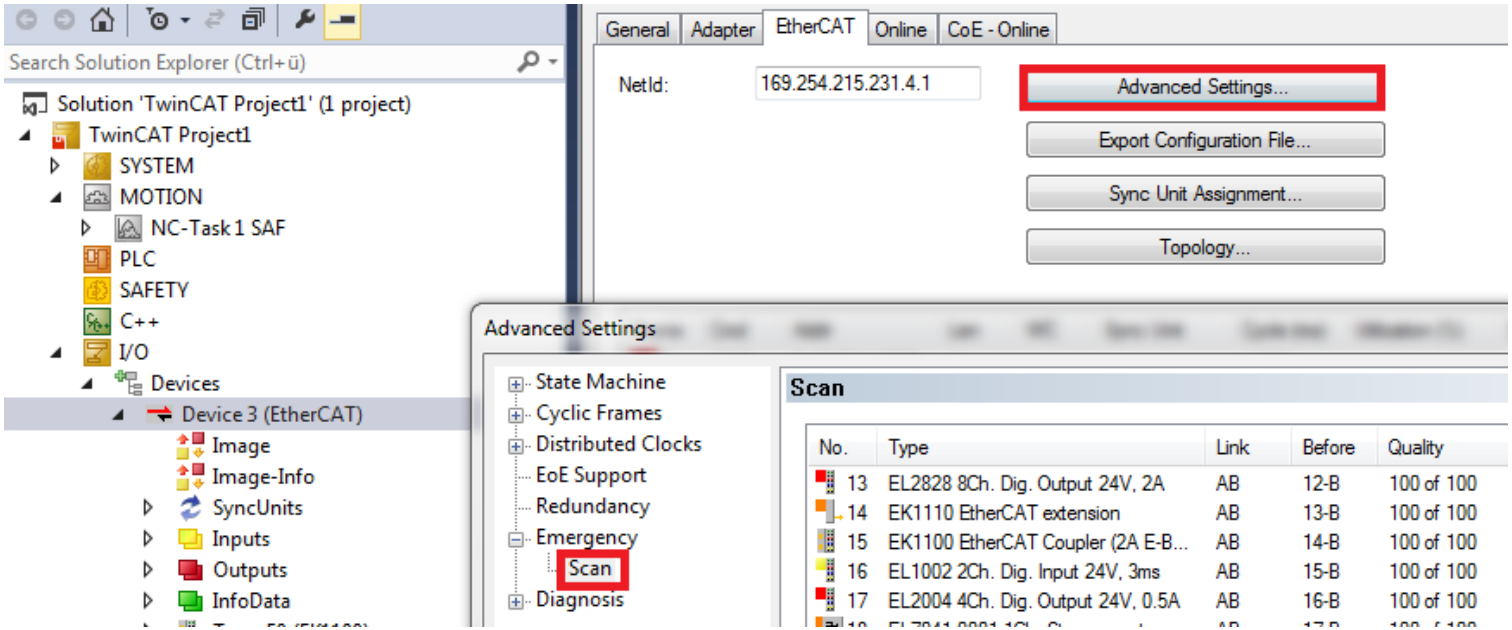


电流消耗达到限值时，应插入一个 **EL94xx** 模块



Emergency Scan（紧急扫描）

Emergency Scan （紧急扫描）操作，可以发送预定数量的探测数据帧，用于快速测试物理连接 (TwinCAT 应处在 **Config Mode**):



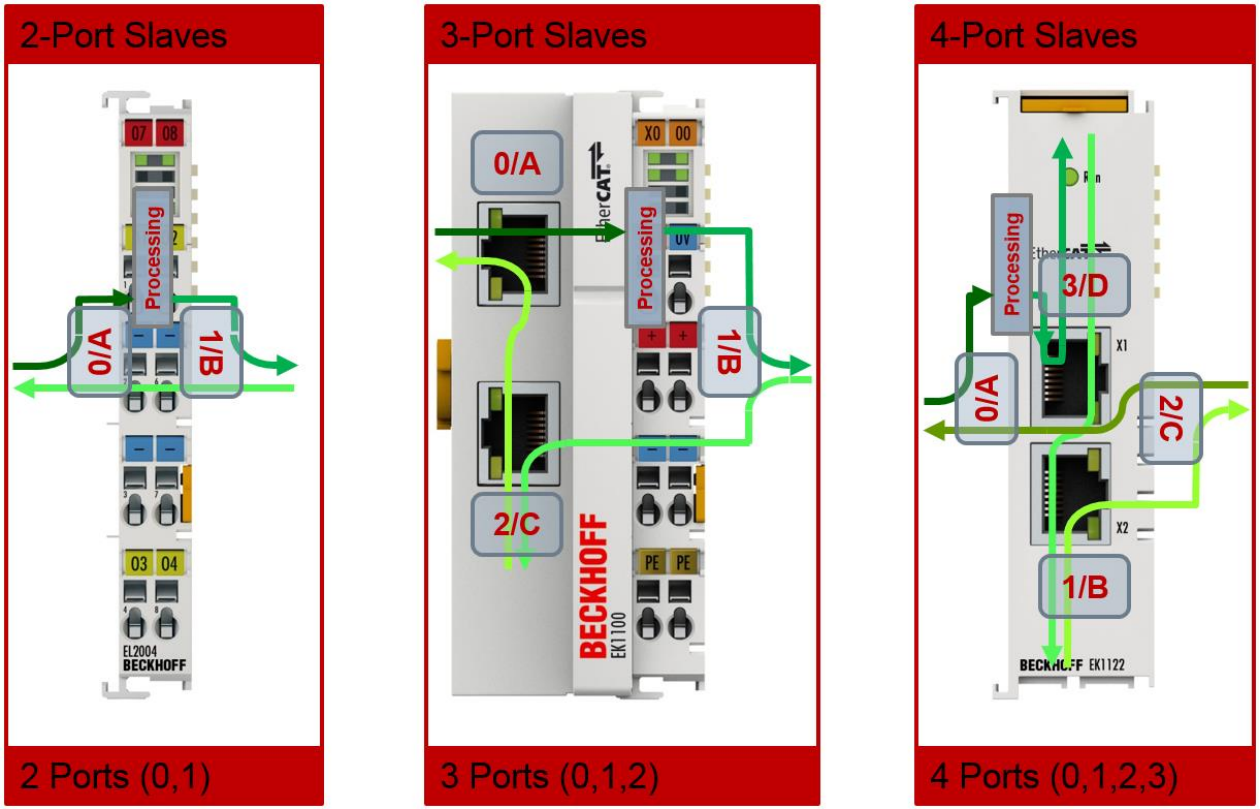
Emergency Scan 能够快速检查EtherCAT网络的永久硬件问题 (设备、电缆或者接头损坏).

用Emergency Scan的方法很难检测到随机的/不定时的干扰，处理这种干扰，应该进行完整的 error counter analysis （错误计数分析）.

EtherCAT 端口的硬件诊断

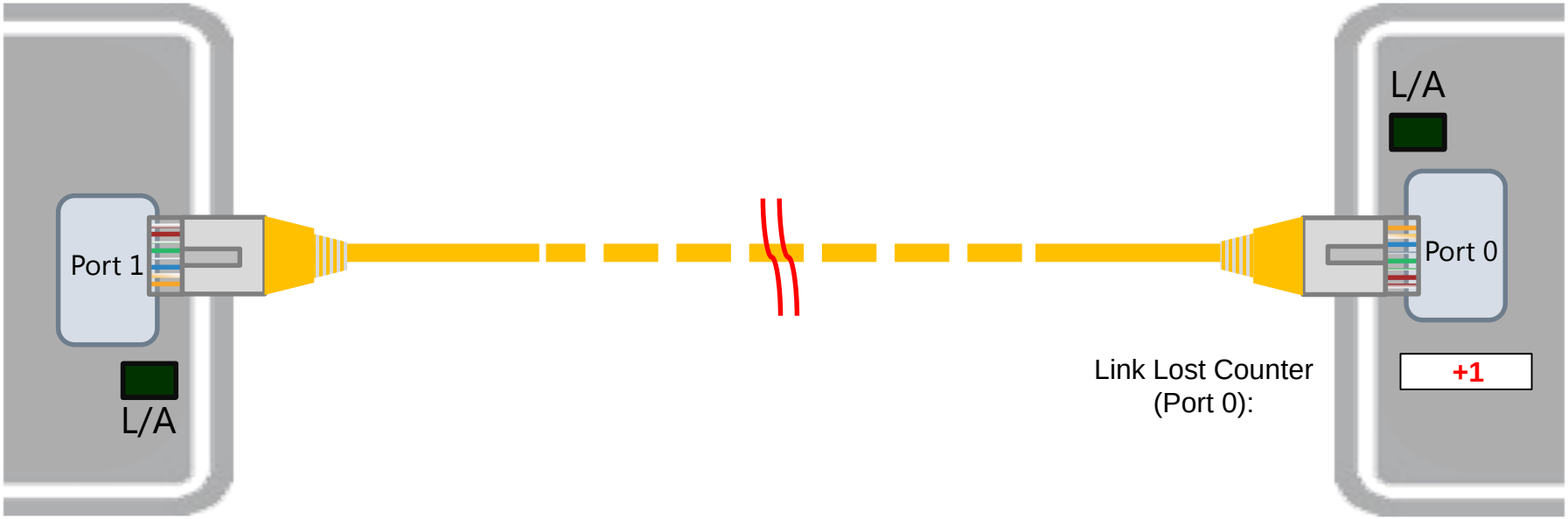
为了进行更深入的分析，从站提供了基于端口的（**port-specific**）硬件错误计数

EtherCAT 定义了端口号 0 to 3 (端口 0 总是输入口)，而TwinCAT 中通常把这些端口依次叫做A A 到 D (A 总是输入口)，两种端口定义方式是等效的：



Link Lost（连接丢失）错误计数器

有时候两个EtherCAT从站之间的物理连接可能会完全中断：信号完全无法到达相邻的从站。



Link loss（连接丢失），最有可能的原因是：

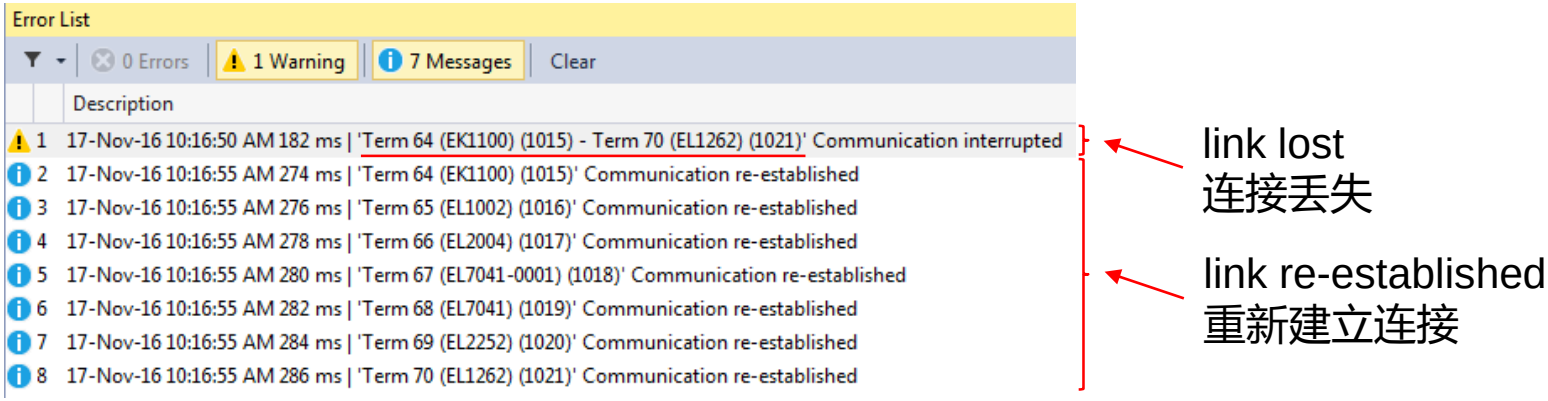
- 电缆或接头损坏（电缆连接），接触弹片不够力或者氧化（ Ebus连接）。
- 一个或者几个从站掉电

Link Lost（连接丢失）错误计数器

一个端口上的物理连接中断，从站就会把该端口的 **Link Lost Counter(连接丢失计数器)** 加 1

Register 内存地址	长度	含 义
0x0310	1 字节	port 0 的连接丢失次数
0x0311	1 字节	port 1 的连接丢失次数
0x0312	1 字节	port 2 的连接丢失次数
0x0313	1 字节	port 3 的连接丢失次数

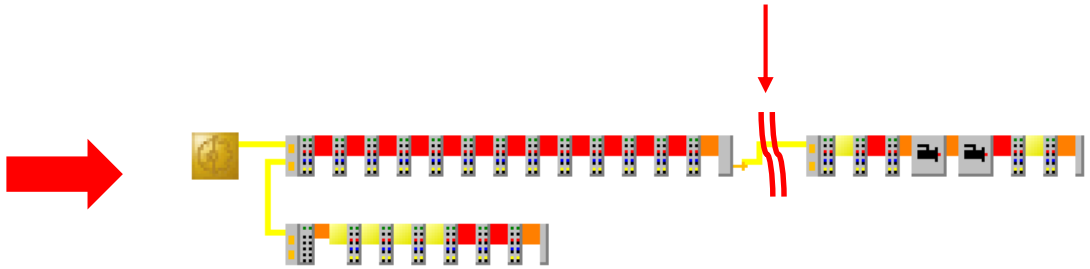
物理连接的状态发生久性或暂时性的改变，在TwinCAT Logger消息区中都会报告（并且保存在 Windows Log 中）



Link/Activity（连接/活动）指示灯

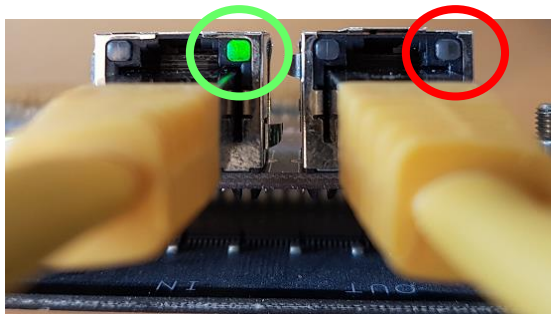
连接的永久中断很容易检查出来，不用分析连接丢失计数器（ Link Lost Counters ），只需要简单查看主站的Online页面：

No	Addr	Name	State
3	1003	Term 52 (EL2202)	OP
4	1004	Term 53 (EL2502)	OP
5	1005	Term 54 (EL2602)	OP
6	1006	Term 55 (EL2622)	OP
7	1007	Term 56 (EL2624)	OP
8	1008	Term 57 (EL2034)	OP
9	1009	Term 58 (EL2024)	OP
10	1010	Term 59 (EL2809)	OP
11	1011	Term 60 (EL2889)	OP
12	1012	Term 61 (EL2808)	OP
13	1013	Term 62 (EL2828)	OP
14	1014	Term 63 (EK1110)	OP LNK_MIS B
15	1015	Term 64 (EK1100)	INIT NO_COMM
16	1016	Term 65 (EL1002)	INIT NO_COMM
17	1017	Term 66 (EL2004)	INIT NO_COMM
18	1018	Term 67 (EL7041-0001)	INIT NO_COMM
19	1019	Term 68 (EL7041)	INIT NO_COMM
20	1020	Term 69 (EL2252)	INIT NO_COMM
21	1021	Term 70 (EL1262)	INIT NO_COMM
22	1022	Term 71 (EK1828-0010)	OP
23	1023	Term 72 (EL1008)	OP
24	1024	Term 73 (EL1008)	OP
25	1025	Term 74 (EL1004)	OP
26	1026	Term 75 (EL1004)	OP
27	1027	Term 76 (EL2798)	OP
28	1028	Term 77 (EL2872)	OP



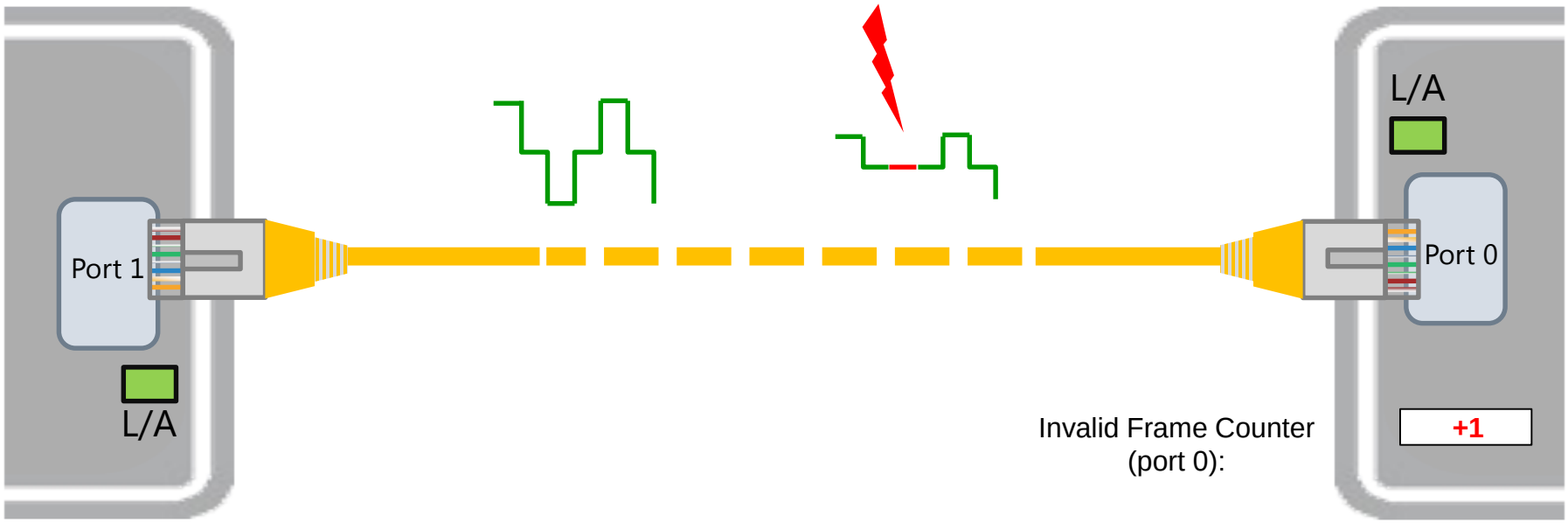
连接永久中断的原因，可能是接头被拔出，电缆严重损坏，或者设备掉电。

用可移动的接头试着接通每个端口，肉眼观察Link/Activity（连接/活动）指示灯，会有助于检查此类故障：



无效帧（Invalid frame）错误计数器

有时候，即使硬件信号到达了相邻的从站，接收到的信号却与最初发送的不一致：



数据包发生损坏，最有可能的原因是：

- 如果该错误计数零星增加，极可能是由于外部 EMC 干扰。
- 如果该错误计数快速且稳步增加，极可能是由于设备损坏。

无效帧（Invalid frame）错误计数

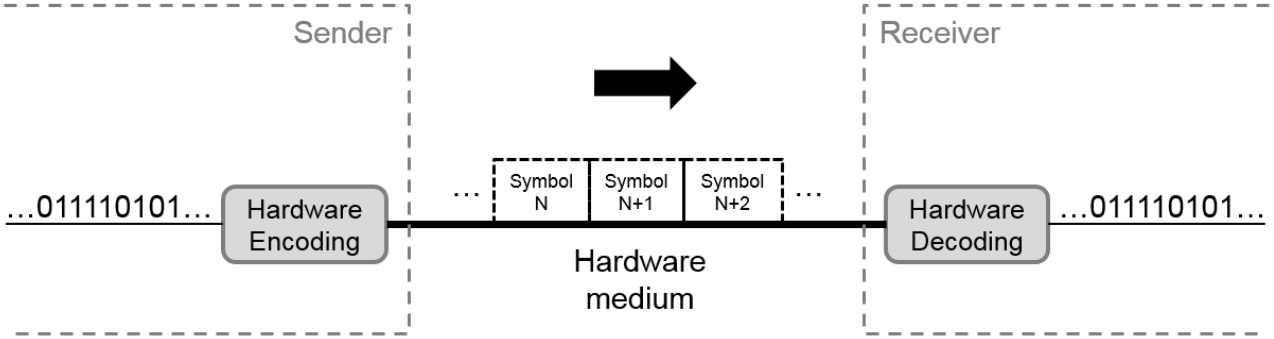
一个端口上的数据帧破坏，从站就会把该端口的 **Invalid Frame Counter(无效帧计数器)** 加 1

Register	Length	Meaning	
0x0300	1 byte	port 0 的CRC校验错误计数器	Invalid Frame Counter port 0 Port 0 的 无效帧计数器
0x0301	1 byte	port 0 的RX （接收）错误计数器	
0x0302	1 byte	port 1 的CRC校验错误计数器	Invalid Frame Counter port 1 Port 1 的 无效帧计数器
0x0303	1 byte	port 1 的RX （接收）错误计数器	
0x0304	1 byte	port 2 的CRC校验错误计数器	Invalid Frame Counter port 2 Port 2 的 无效帧计数器
0x0305	1 byte	port 2 的RX （接收）错误计数器	
0x0306	1 byte	port 3 的CRC校验错误计数器	Invalid Frame Counter port 3 Port 3 的 无效帧计数器
0x0307	1 byte	port 3 的RX （接收）错误计数器	

RX 错误计数器 和 CRC 校验错误计数器，都意味着硬件接收到的数据帧损坏，表征信息非常相近，但是二者分别是在从站架构的不同层面检测到的，其含义也略有不同。

物理介质上的信号传输

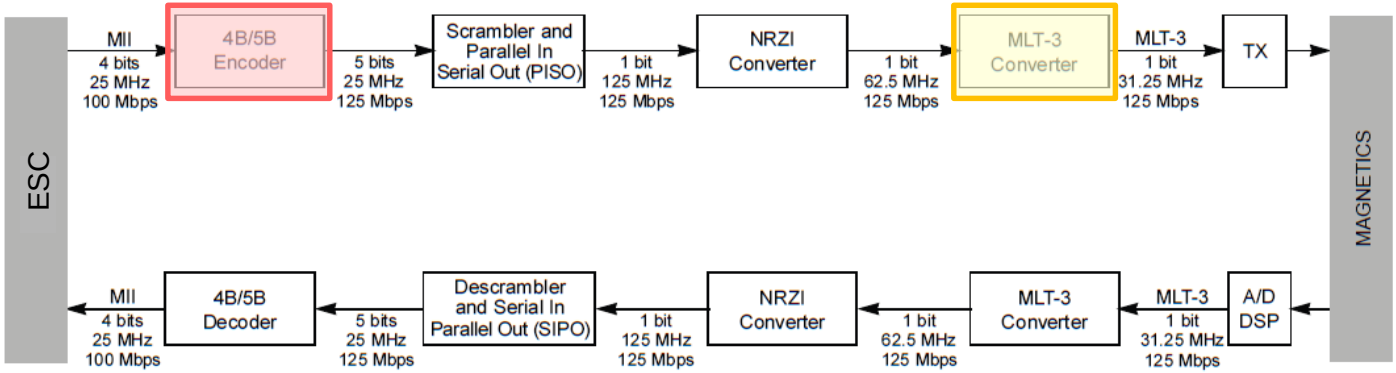
0和1的逻辑序列要在物理介质上传输，需要编码成预定义的电压/电流 电平（或者电平转换）



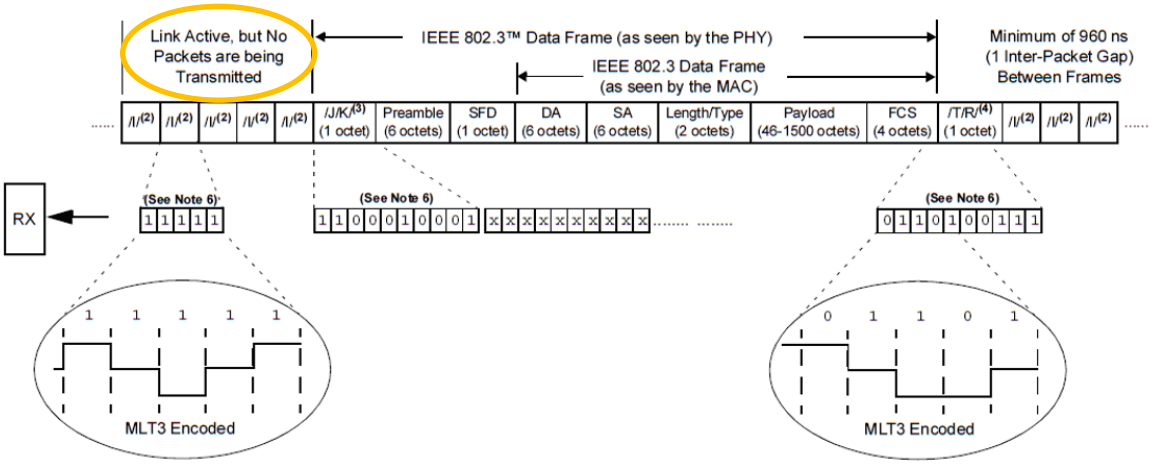
- 电流/电压 值的序列叫做 **symbols（符号）**.
  - 基于特定的硬件编码标准，电流/电压值的可能序列并不是全都具有含义，这样就有了有效符号和无效符号的分别。
  - 物理介质上符号是连续传输的，既在Ethernet数据帧内也在数据帧外（数据帧外的符号序列是为了让接收方及时检测到可能发生的连接丢失错误）

- 通讯由**符号序列**组成
  - 携带着有含义的信息的符号序列，就是**Ethernet 数据帧**。
  - 两个Ethernet数据帧之间传输的符号序列，就是 **数据帧间隔**

深入了解 – 100BASE-TX 硬件编码标准（网线连接）



Control Character	5b symbols	Purpose
JK	11000 10001	Sync, Start delimiter
II	11111 11111	Not Used
TT	01101 01101	FDDI end delimiter
TS	01101 11001	Not Used
IH	11111 00100	SAL
TR	01101 00111	100BASE-TX end delimiter
SR	11001 00111	Not Used
SS	11001 11001	Not Used
HH	00100 00100	HDLC0
HI	00100 11111	HDLC1
HQ	00100 00000	HDLC2
RR	00111 00111	HDLC3
RS	00111 11001	HDLC4
QH	00000 00100	HDLC5
QI	00000 11111	HDLC6
QQ	00000 00000	HDLC7



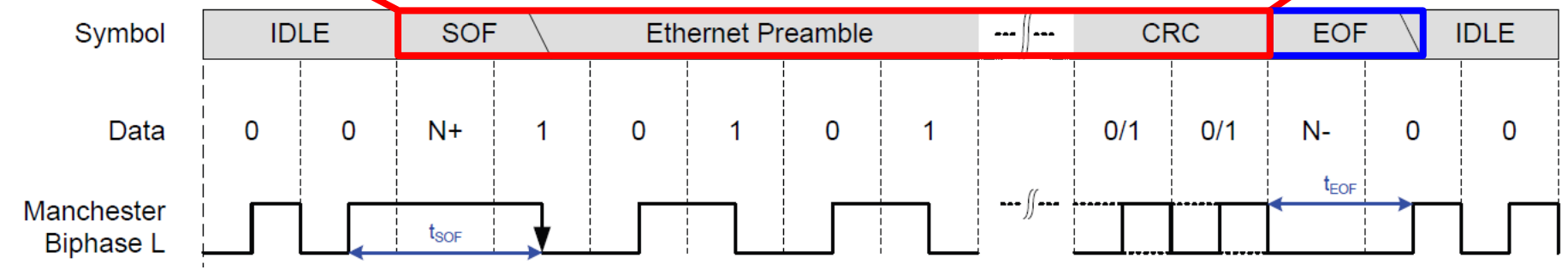
Multi-Layer Transmit 3 Code:

- Logic 0 : 低 → 高 转换
- Logic 1 : 高 → 低 转换

深入了解 – LVDS 硬件编码标准（EBUS连接）

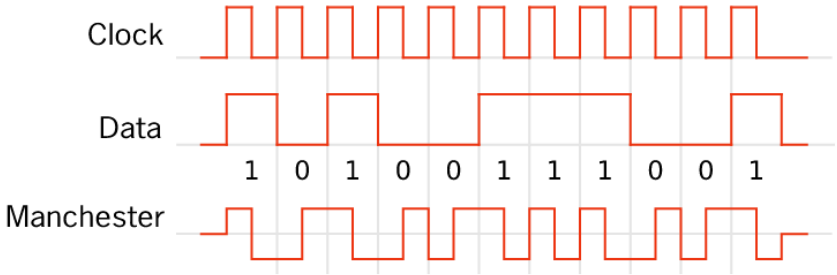
EBUS 是基于 LVDS 的信号，兼容 ANSI/TIA/EIA-644 规范:

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46(42)[b]–1500 octets	4 octets	12 octets
Layer 2 Ethernet frame	← 64–1518(1522) octets →								
Layer 1 Ethernet packet	← 72–1526(1530) octets →								



Biphase-L Mancherster Code:

- Logic 0 : LO → HI transition
- Logic 1 : HI → LO transition



无效帧（Invalid frame）错误计数器

硬件错误分为两种类型，都在无效帧计数器中累计

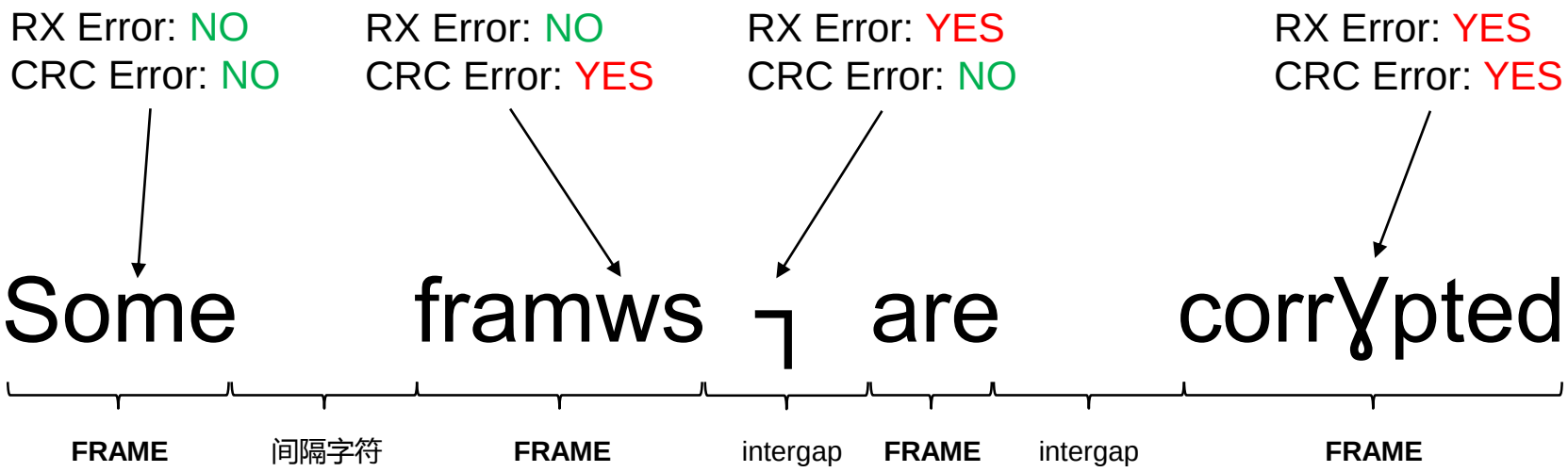
- **RX Errors**:
  - 个别符号错误 (被特定的硬件解码器识别为无效）.
  - 在数据帧内部或者外部都可能发生 (即使是在没有数据帧传输的时候，每个物理接口都会传输 idle 即表示空闲的符号)
  - → **RX Error Counters** (无效帧计数器的高字节)

- **CRC Errors**:
  - 整个数据帧的循环冗余校验（CRC）错误.
  - 在数据帧内发生 (只有Ethernet 数据帧才进行校验).
  - → **CRC Error Counters** (无效帧计数器的低字节)

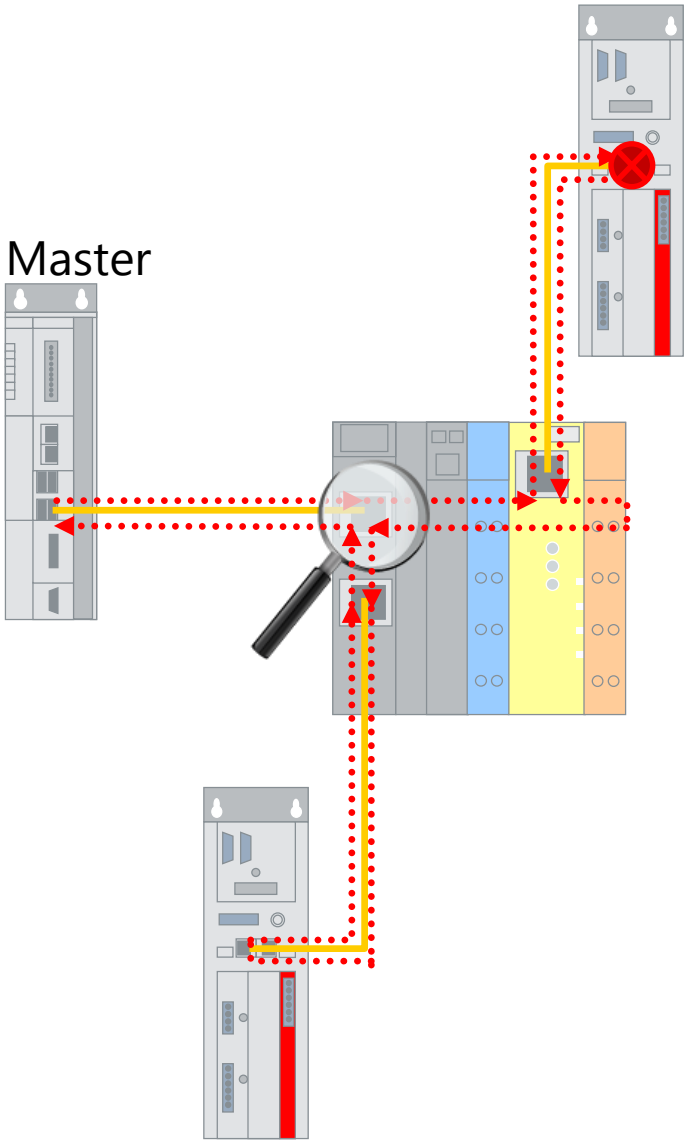
RX 和 CRC 错误的类比

硬件通讯介质上的数据传输可以用标准的语言书写来打个比方：

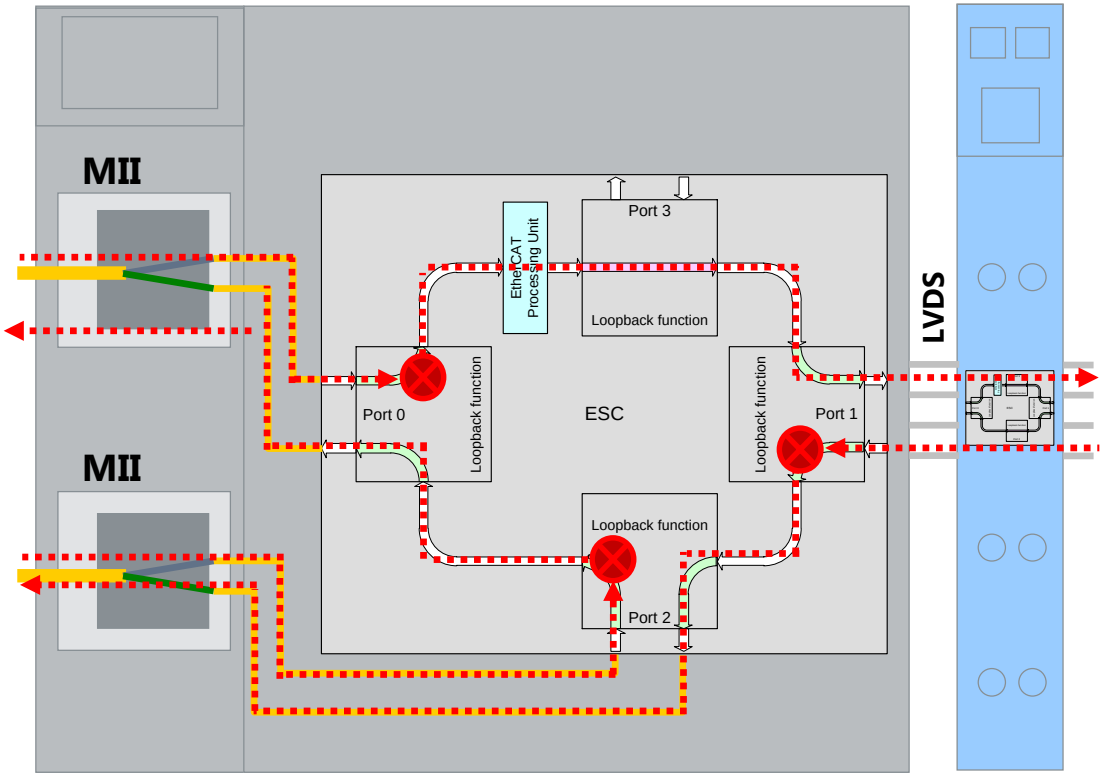
- RX 和 CRC 错误 在解释上具有细微的差别，例如：



CRC 错误 – 什么情况下会检测出来?



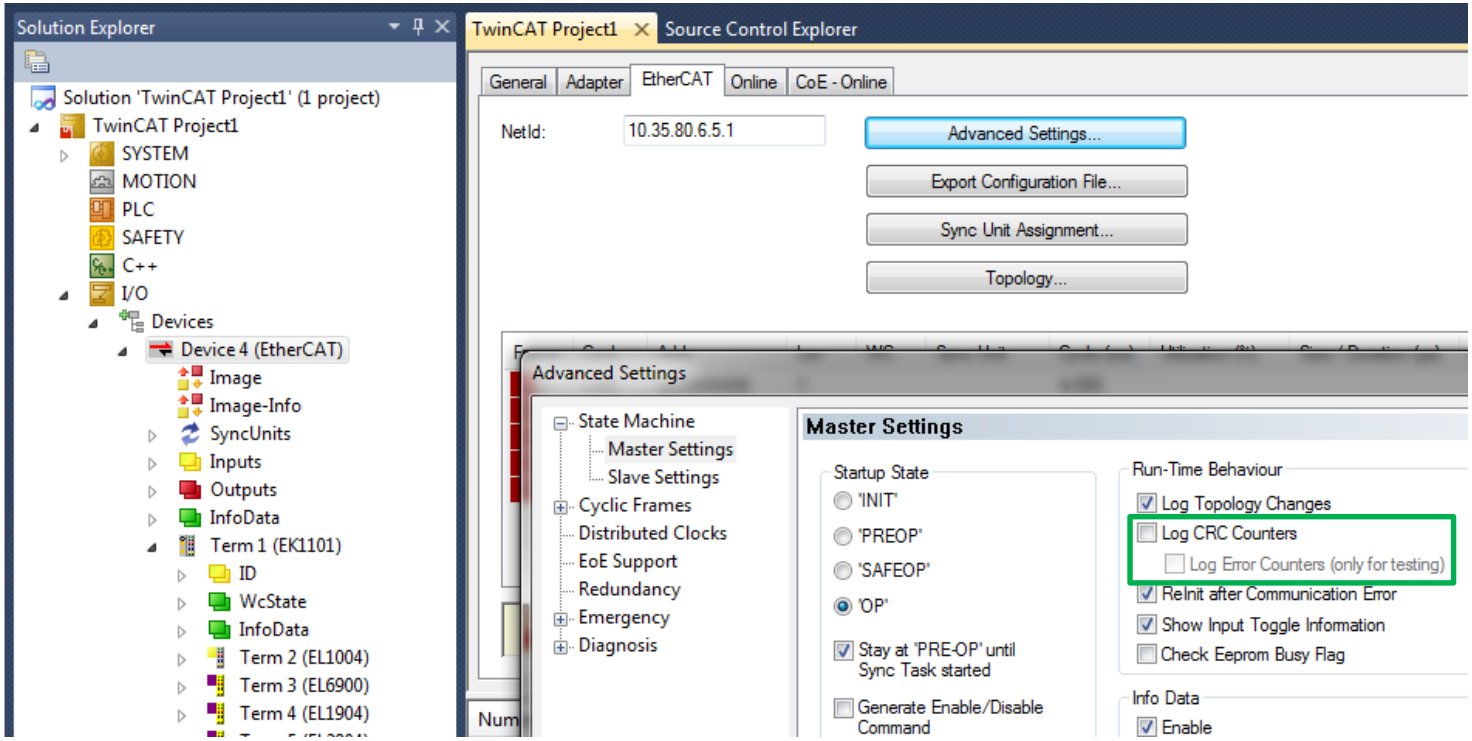
仔细考虑一个特定的CRC错误校验过程是这样的： 从站只在数据帧从外面进入端口时执行CRC校验： port 0 于是检查传播的前进方向的数据帧，而 ports 1, 2 及 3 检查返回方向的数据帧。



怎样正确跟踪硬件错误计数

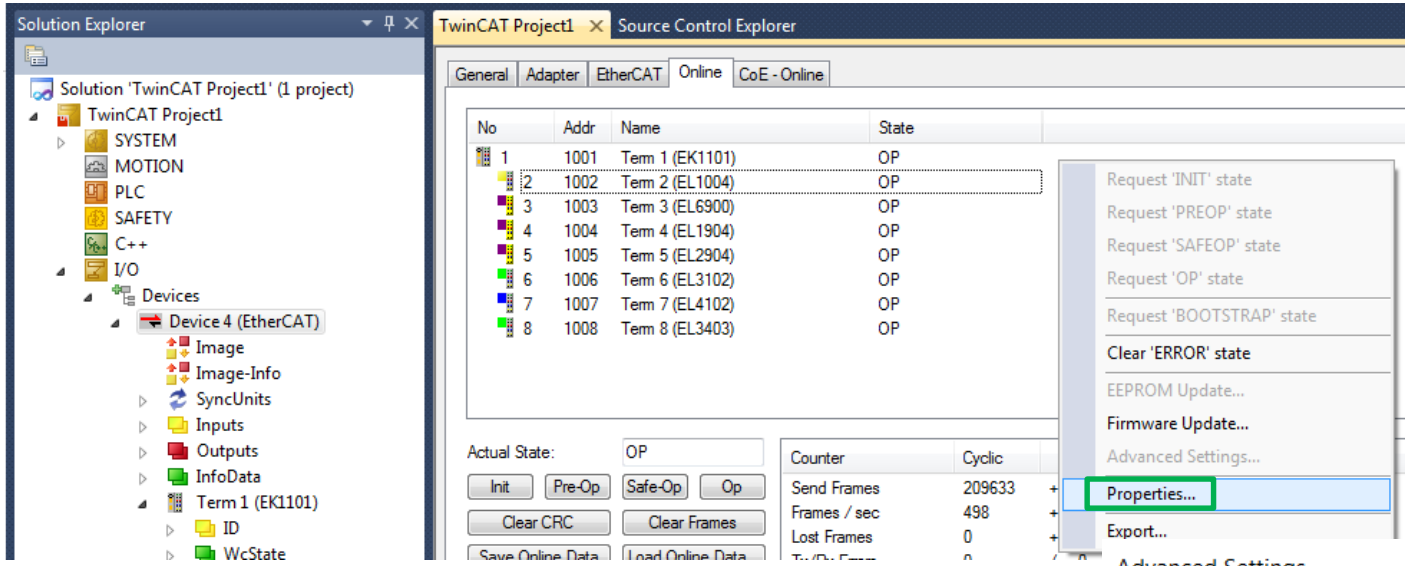
为了追踪硬件错误计数，建议进行以下设置:

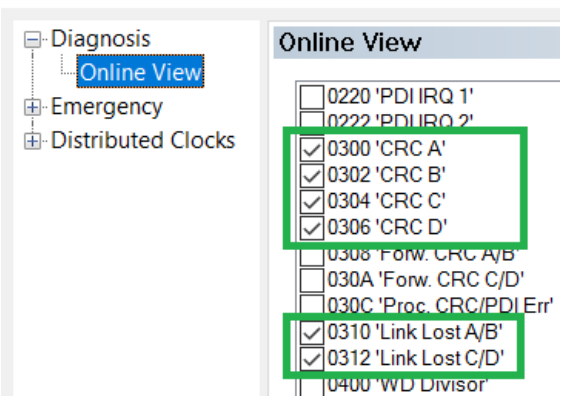
1. 在主站的Advanced Setting中 **不要勾选** “Log CRC Counters” 标记



怎样正确跟踪硬件错误计数

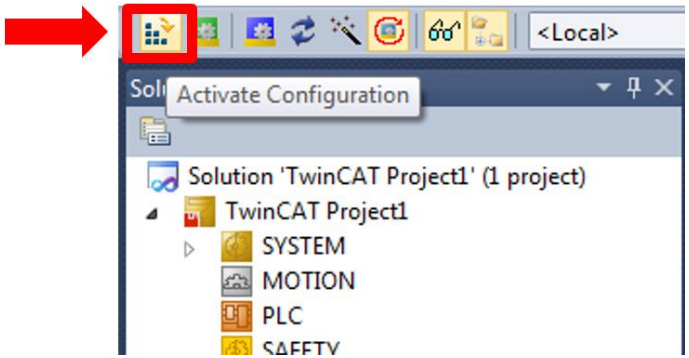
2. 在Online视图中添加注册字 0x0300÷0x030A 和 0x0310÷0x0312



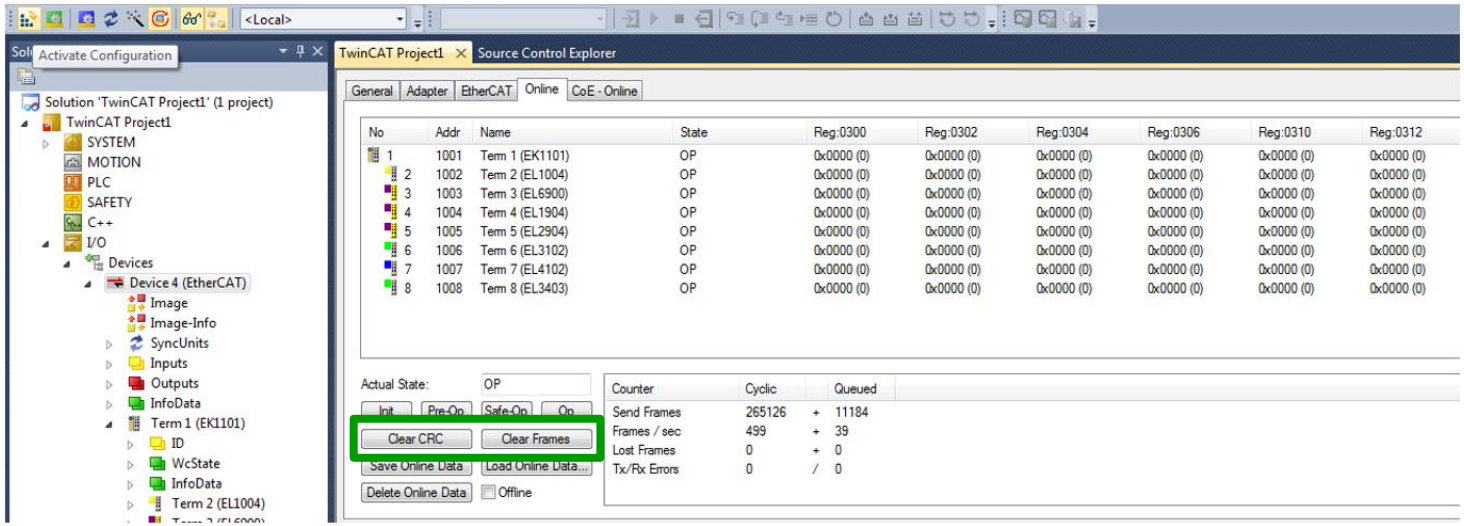


怎样正确跟踪硬件错误计数

3. 激活并重启 TwinCAT 配置:

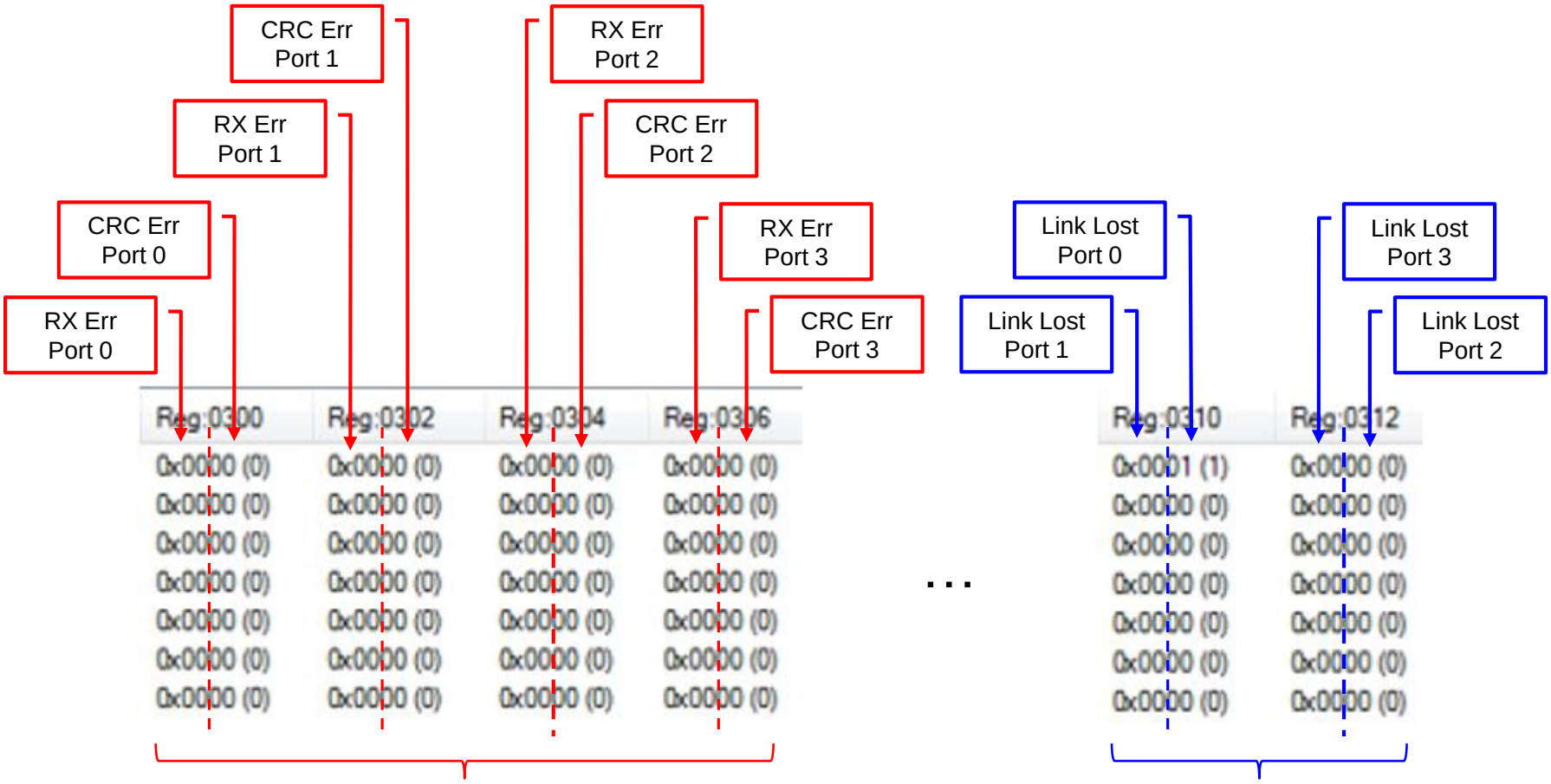


4. 重启TwinCAT后，清空所有错误计数:



怎样正确跟踪硬件错误计数

TwinCAT 中， EtherCAT主站的 Online 页面总是以 **word-oriented** 的方式显示计数值

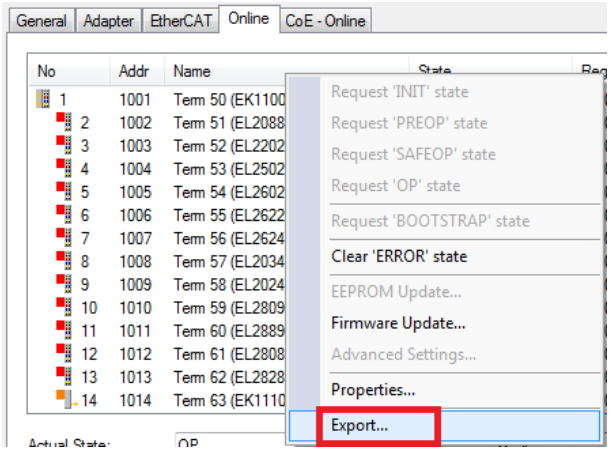


Invalid Frame Counters
无效数据帧计数器

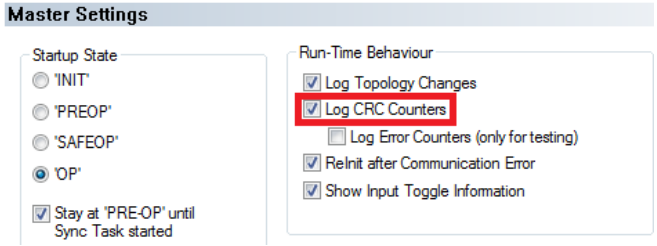
Link Lost Counters
连接丢失计数器

怎样正确跟踪硬件错误计数

3. 等待，直到错误补检测到：捕捉到的错误次数越多越好（关闭项目或者重启TwinCAT就会清除计数，而最小化项目或者切换窗口则不会清除计数). 记录的值可以导出，以供进一步分析。



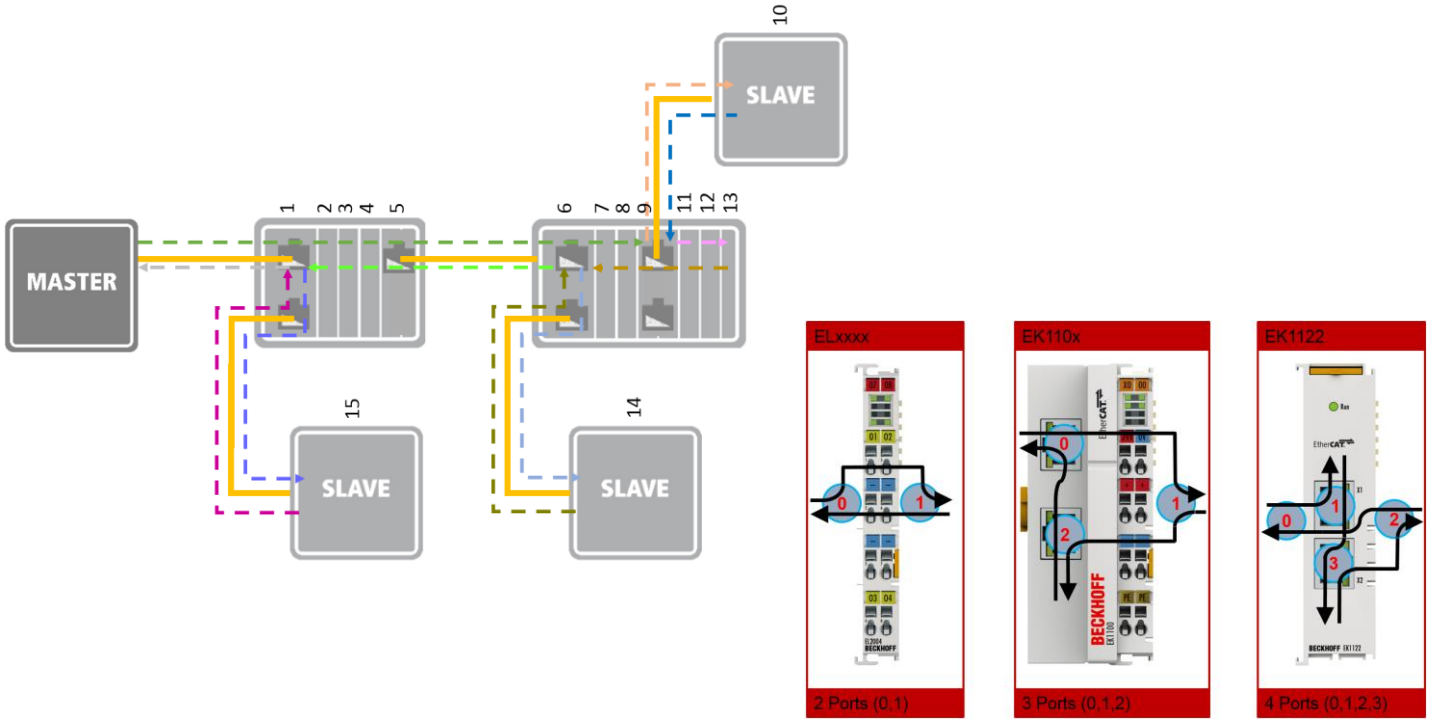
错误计数采集完成后，记得应把 “Log CRC Counters” 重新勾选上。(TwinCAT 需要重启).



怎样定位一个错误?

为了从显示的CRC错误计数分析出网络中的故障点分布情况，建议:

1. 逻辑上沿着网络数据帧传播的方向确定从站端口执行CRC校验的顺序(根据前述信息 previous information):



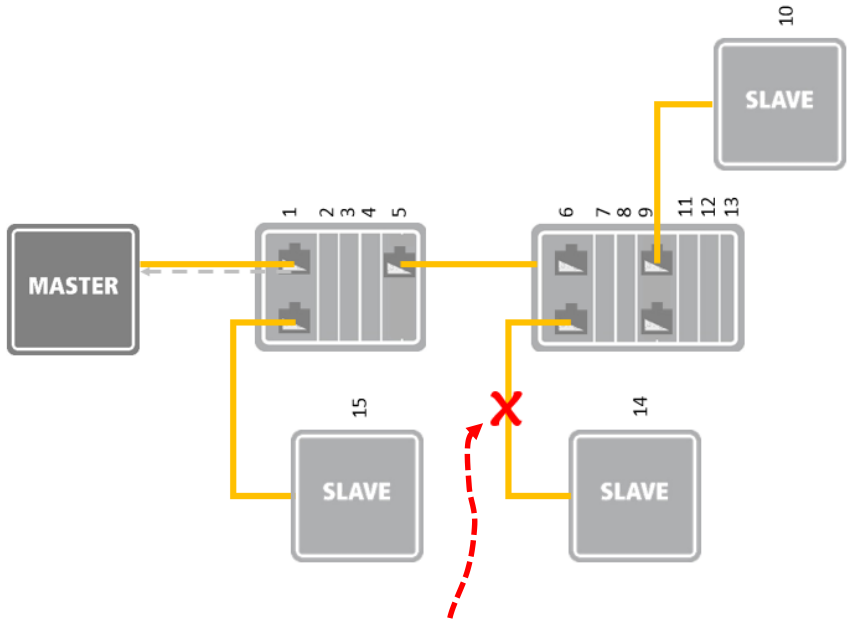
CRC checked by	
slave 1	port 0
slave 2	port 0
slave 3	port 0
slave 4	port 0
slave 5	port 0
slave 6	port 0
slave 7	port 0
slave 8	port 0
slave 9	port 0
slave 10	port 0
slave 9	port 3
slave 11	port 0
slave 12	port 0
slave 13	port 0
slave 12	port 1
slave 11	port 1
slave 9	port 1
slave 8	port 1
slave 7	port 1
slave 6	port 1
slave 14	port 0
slave 6	port 2
slave 5	port 1
slave 4	port 1
slave 3	port 1
slave 2	port 1
slave 1	port 1
slave 15	port 0
slave 1	port 2

怎样定位一个错误?

2. 检查第一个报告错误计数器不为零的端口:

	CRC port 0	CRC port 1	CRC port 2	CRC port 3
slave 1	0x00	0x00	0x00	
slave 2	0x00	0x00		
slave 3	0x00	0x13		
slave 4	0x00	0x00		
slave 5	0x00	0x13	0x13	
slave 6	0x00	0x00		
slave 7	0x00	0x00		
slave 8	0x00	0x00		
slave 9	0x00	0x00		0x00
slave 10	0x00			
slave 11	0x00	0x00		
slave 12	0x00	0x00		
slave 13	0x00			
slave 14	0x0A			
slave 15	0x13			

← 在 TwinCAT 中查看错误 (对比前述的单向度的清单 previous slide)



报告无效错误计数器不为零的第一个端口 → 极有可能就是故障点.

在故障点采取什么措施?

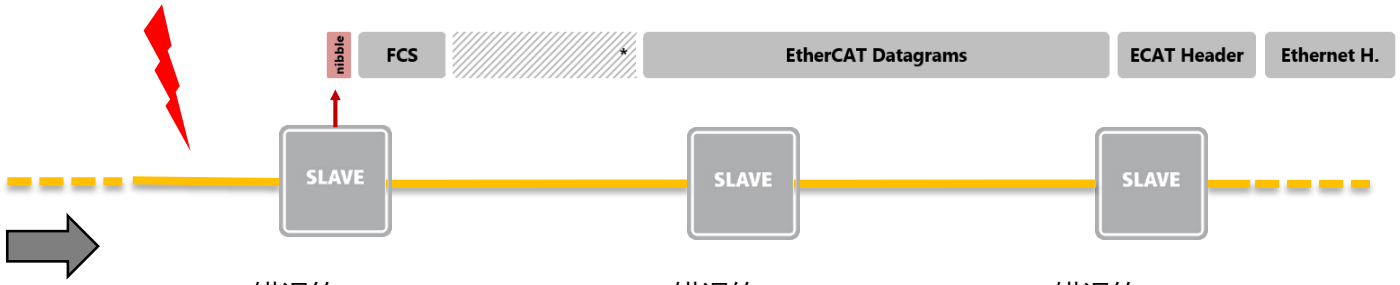
3. 在上页定位到故障点 previously located 后，应进行以下操作：
   - 检查与上个从站之间的连接电缆：
     - EtherCAT网线的走向是否靠近电源线或者干扰源?
     - 是否自制电缆的接头做工很差?
     - 电缆的屏蔽层是否接地良好?
   - 检查故障点的站和前一个从站：
     - 是否供电不足 (供给本站的电流太小, 例如：EBUS 电流)?
     - 两个从站的接地点不是等电位？
   - 试着替换该从站和前一个从站，或者交换二者的顺序，以查看错误是跟随某个从站还是出现在原先的位置。

对于 EMC 干扰, 应该会导致 RX 和 CRC 错误都有增加 (即使二者增加的机率可能不同), 因为外部干扰不可能与通讯同步，于是在Ethernet 数据帧内部和外部都会破坏数据。

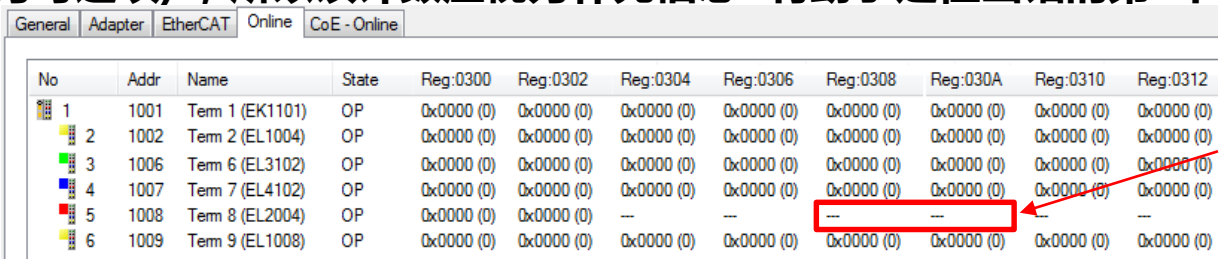
完全不平衡的计数值 （RX错误很多，而没有CRC错误，或者很CRC错误而没有RX错误）可能意味着其中一个从站的内部硬件错误。

深入了解 - CRC 错误的转发机制

部分从站还额外支持 **转发CRC错误计数器（Forwarded CRC Error Counters）**：第1个检查出CRC错误的从站，其CRC 错误计数器（Register 0x0300）增加，第2个及之后再检查出CRC错误的从站则增加其转发CRC错误计数器（Register 0x0308）而CRC 错误计数器保持不变。



并非所有从站都支持 转发CRC错误计数器（Forwarded CRC Error Counters） 功能 (该特性为可选项)，所以该计数应视为补充信息: 有助于定位出错的第1个设备，但不是关键指标。

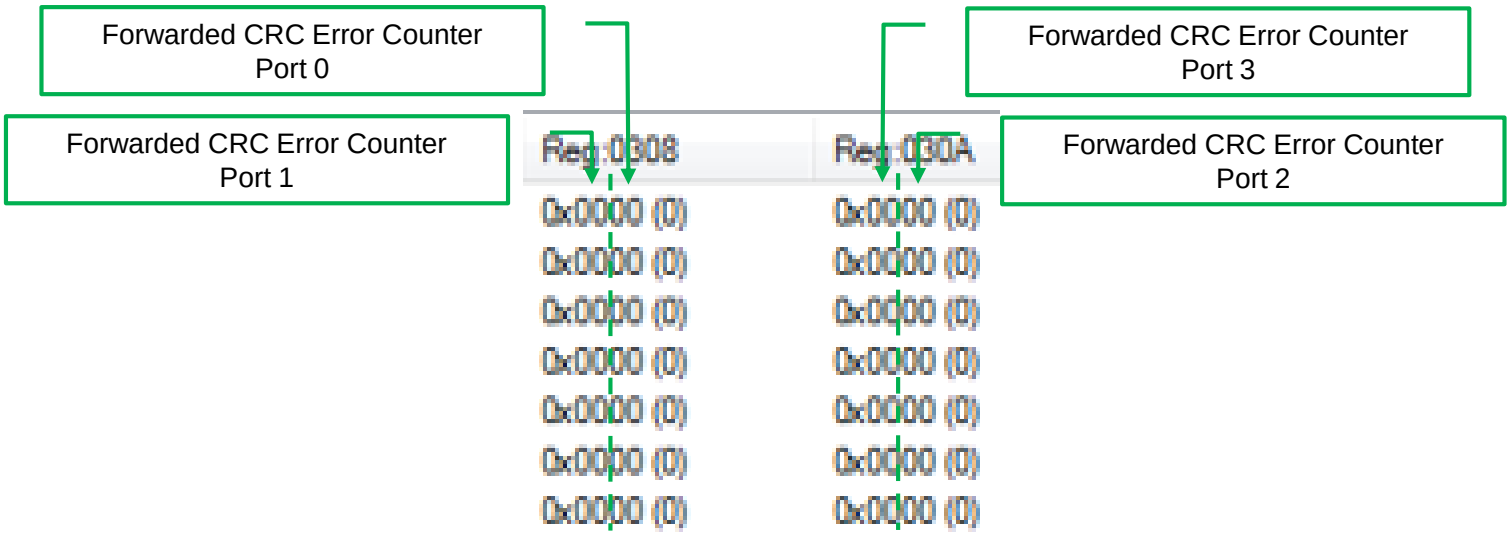


如果 从站不支持Forwarded CRC Error Counters功能，TwinCAT中就如此显示

深入了解 – 转发CRC错误的计数器

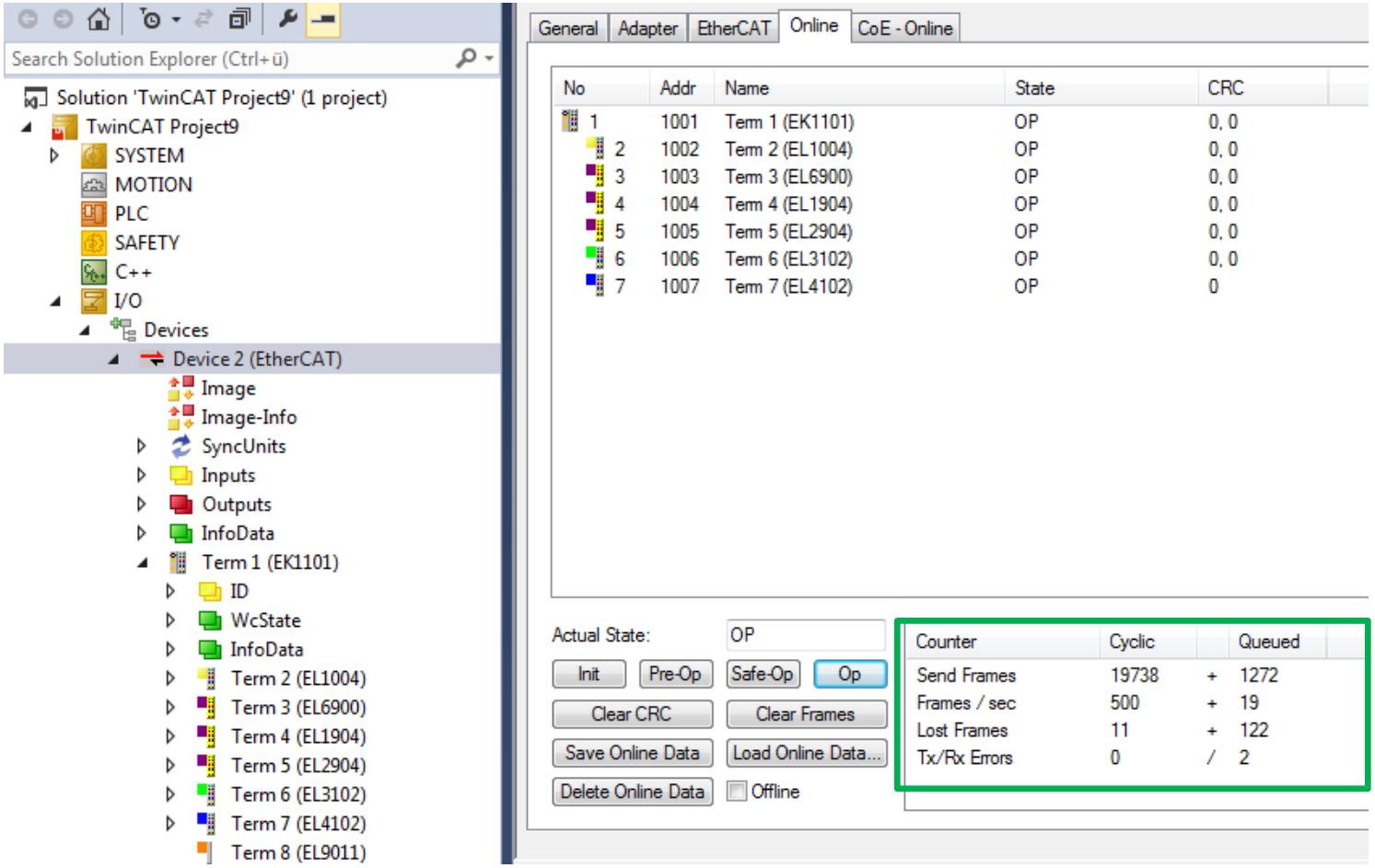
TwinCAT 中， EtherCAT主站的 Online 页面总是以 **word-oriented** 的方式显示计数值，所以转发CRC错误计数器（ Forwarded CRC Error Counter ）的说明如下：

Register (注册字)	长度	含　义
0x0308	1 byte	port 0 的转发CRC错误次数（Forwarded CRC Error Counter ）
0x0309	1 byte	port 1 的转发CRC错误次数（Forwarded CRC Error Counter ）
0x030A	1 byte	port 2 的转发CRC错误次数（Forwarded CRC Error Counter ）
0x030B	1 byte	port 3 的转发CRC错误次数（Forwarded CRC Error Counter ）



主站错误计数

此外，EtherCAT **主站** 也提供硬件错误 **计数**：



主站错误计数

EtherCAT 主站支持两种不同的错误计数器：

- **Rx Errors**: 与从站侧的 RX 错误计数器类似，由网卡统计符号错误的次数 (发生在数据帧内和数据帧外的错误).
- **Lost 数据帧**: 由于EtherCAT闭环在某处中断未能返回主站的数据帧和返回了但CRC校验错误的数据帧，TwinCAT 都视作丢失的“lost“。在TwinCAT中，如果丢失一个数据帧，则该数据帧包含的所有数据报文（ Datagram，即Sync Unit同步单元）都会报Working Counter 错误，显示为所有从站的WcState为Invalid。

分别考虑不同的数据帧类型：

- **Cyclic**: 周期性数据帧 发送要交换的过程数据（ Process Data ）：

Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id	Flags
0	LRD	0x09000000	1			4.000				
0	LRW	0x01000000	16	9	<default>	4.000				
0	LWR	0x01000800	4	1	<default>	4.000				
0	LRD	0x01001000	11	3	<default>	4.000				
0	BRD	0x0000 0x0130	2	7		4.000	0.27 0.27	110 / 10.72	0	

- **Queued**: 排队等候的数据帧，包括所有非周期性的数据帧：

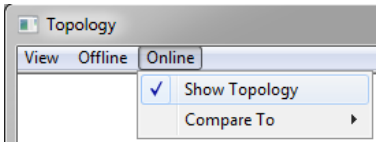
  邮箱 Mailbox,

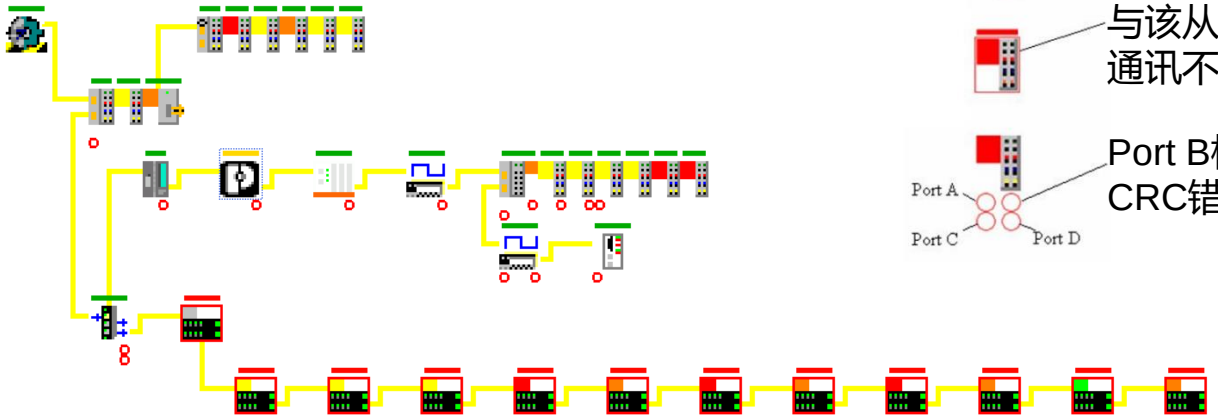
  状态机 State Machine,

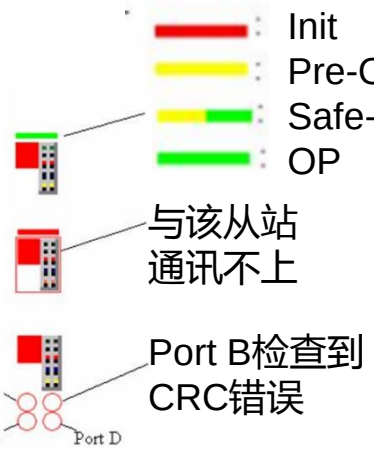
  注册字访问 Register access

在线查看拓扑结构

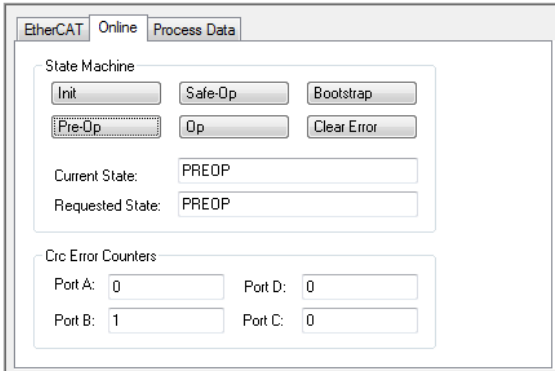
大部分硬件诊断信息，可以在 **Topology View** 界面监视







Topology View 是个 ActiveX 插件，可以导出到任何独立于TwinCAT 开发环境的 Windows 应用。



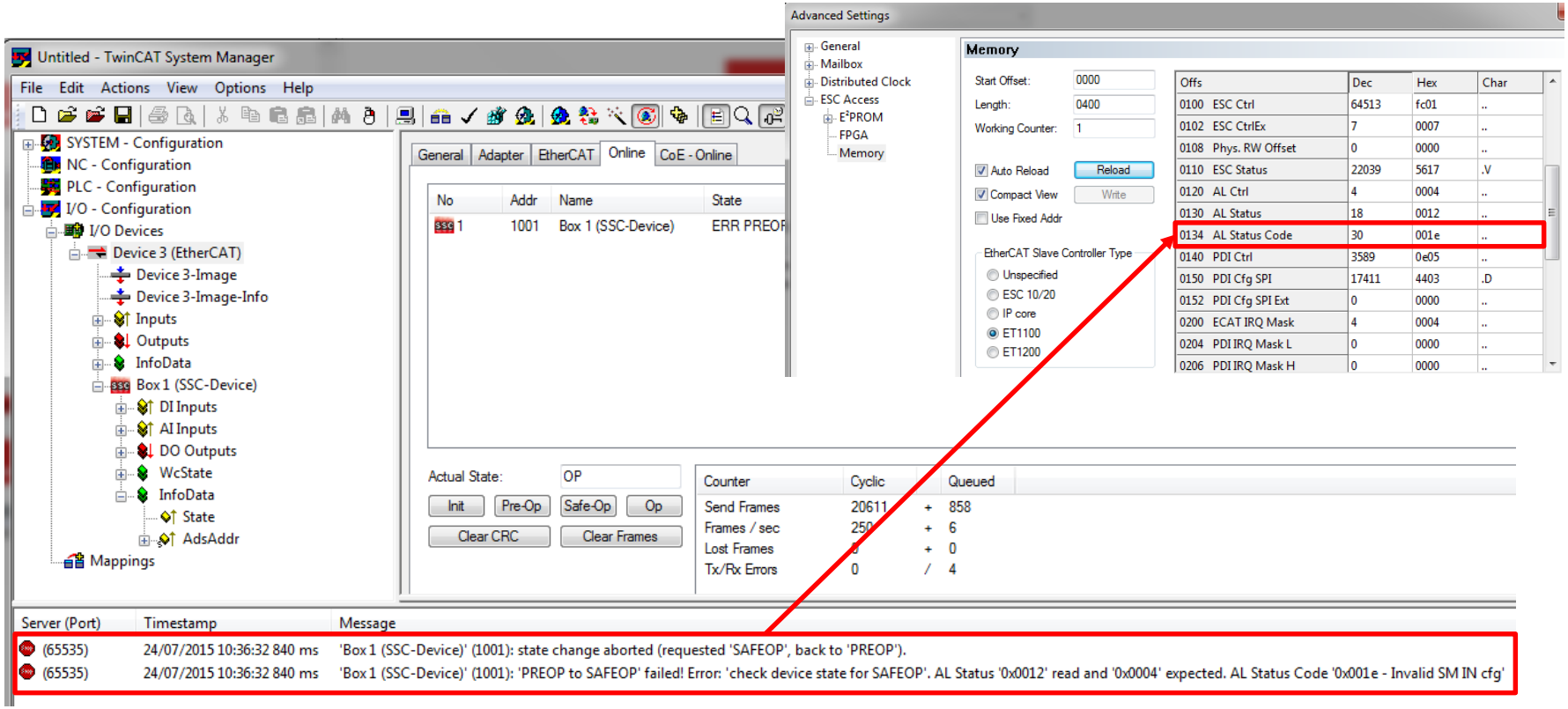
Software Diagnostics

软件诊断

软件层面的错误 – EtherCAT 状态机

软件层面的诊断信息，通常与EtherCAT 状态机操作有关: 一旦从站不能按主站要求进入正确的状态， 就会在从站的注册字（Register 0x0134）中报告AL状态代码（**AL Status Code**）.

在 TwinCAT Logger 中也会显示状态机错误:

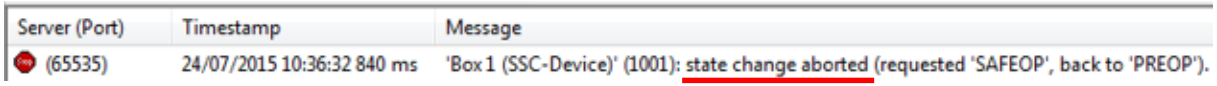


EtherCAT 状态机错误的类型

EtherCAT 状态机错误可以归为**2个大类**:

- **初始化错误** (启动时从站不能进入OP状态): 状态机切换过程中，主站根据 ESI 文件的内容向从站发送初始化命令. 如果从站检测到一个或者几个 start-up 参数无效，就会拒绝相应的状态机切换。

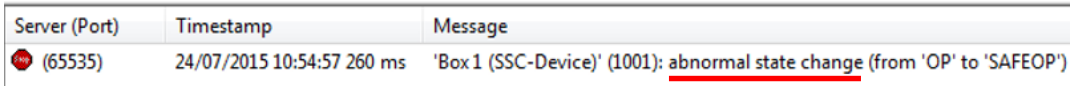
  典型 的初始化错误:



  此时 Register 0x0134 的值显示为：
  - 0x0003 : Invalid Device Setup (BK1xxx上配置的KL模块顺序不正确)
  - 0x001D : Invalid Output Configuration (配置的输出过程数据无效)
  - 0x001E : Invalid Input Configuration (配置的输入过程数据无效)
  - 0x0035 : Invalid Sync Cycle Time (在 DC 模式中设置的Cycle Time不支持)

- **运行时错误** (从站退出OP到更低级的状态)： 配置正确的从站成功进入OP状态以后，在运行过程中检测到错误，并因此执行切换到更低级的状态。

  典型的运行时错误:



  此时 Register 0x0134 的值显示为：
  - 0x001A : Synchronization error (网络的抖动导致从站的同步丢失)
  - 0x001B : Sync manager watchdog (从站未接收到周期性数据的时间，持续超过了watchdog时间)
  - 0x002C : Fatal SYNC error (ESC 再未收到 SYNC 硬件中断)

发生初始化错误 – 怎么办?

从站的 ESI 文件应包含TwinCAT在网络中正确配置该从站所需要的全部信息。如果使用从ESI读取的默认设置来激活配置 （而不做任何手动修改），从站应该能进入OP状态而不报错。

如果发生了初始化错误:

1. 确保 ESI 文件复制到了 包含所有从站描述文件的TwinCAT统一路径（ESI 文件中的Product Code 和 Revision Number 与CoE对象 0x1018 中显示的信息相匹配）。
2. 检查从站的默认设置是否改动过，如果改过就删除该从站再在TwinCAT配置中手动添加。(以恢复默认设置)

3. 对模块化的从站(For modular slaves) ，检查“Slots” 页面配置的模块与实际连接的模块在型号、数量、顺序上是否严格一致。

4. 对具有分布时钟同步功能的从站设备(For DC-Synchronous devices)，检查主站的Jitter或者Sync Shift Time的设置是否会妨碍从站正确实现DC同步。
   如果是这种情况 ，只是检查并不能解决问题，必须联系从站的制造商。

发生运行时错误 – 怎么办?

从站成功进入OP状态以后，在运行时不会无故退出OP状态。

如果发生了运行时错误：

1. 如果从站发生了 watchdog 错误，检查TwinCAT 软件任务 (PLC 程序, NC 任务, …)是否正确运行，因为在TwinCAT中总是由上述任务触发数据帧的周期性发送和接收。

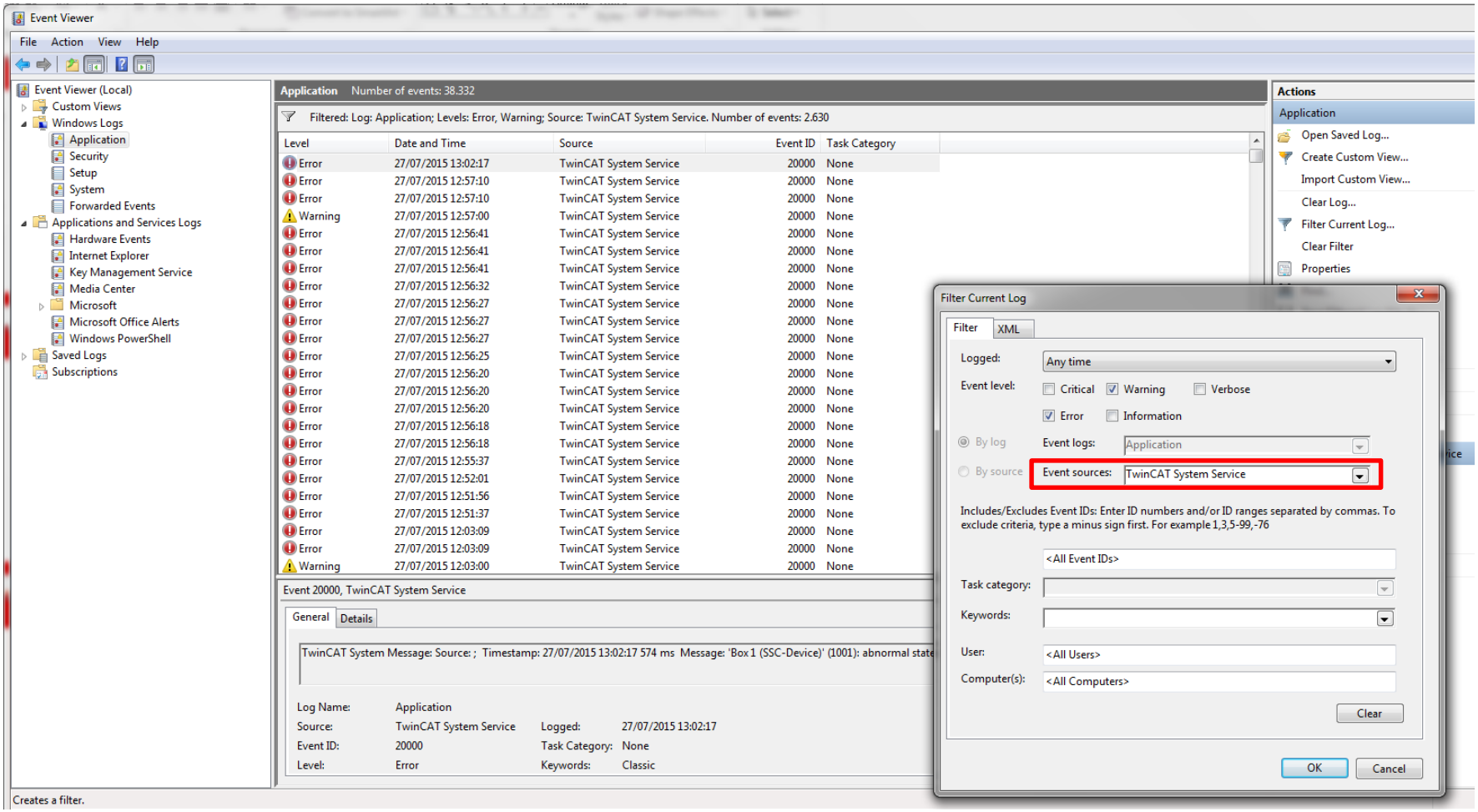
2. 检查主站设备的Jitter表现（抖动范围）是否能证明发生了同步丢失。(最大的Jitter超过通讯周期的20 - 30%时，很容易发生同步错误).

3. 检查是否发生了硬件错误，比如物理连接丢失，这种错误可能间接导致Watchdog动作，或者同步丢失（参考“物理层问题的硬件诊断 ”).

如果是这种情况 ，只是检查并不能解决问题，使用 Wireshark 进行追踪会是个有用的办法。

保存和导出软件错误

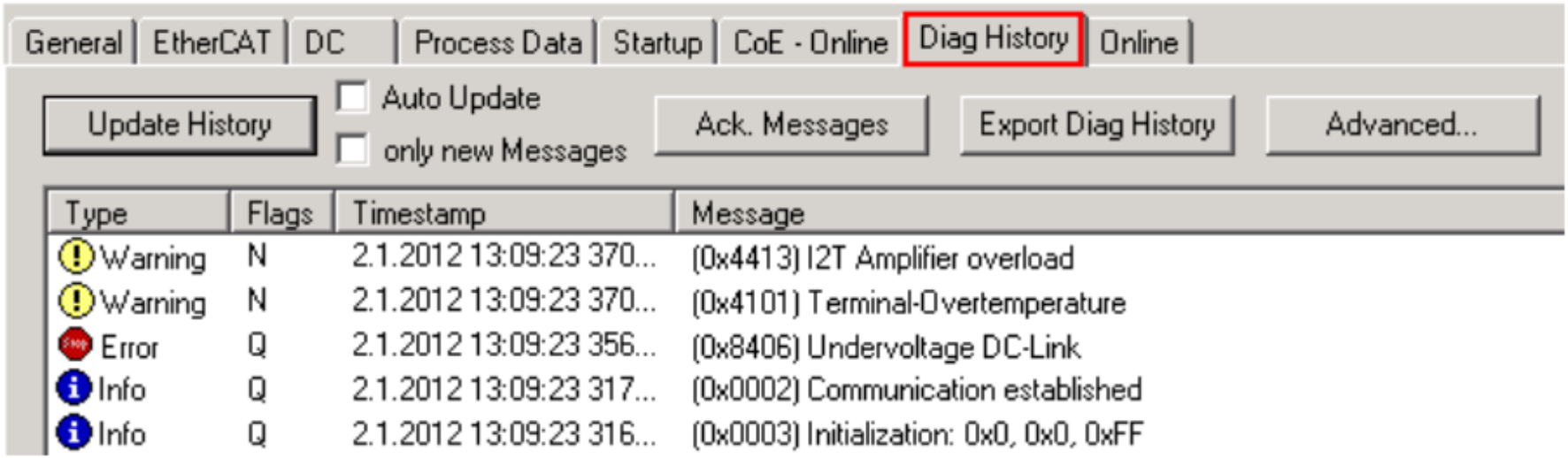
状态机错误保存在 **Windows Log** 中，可以导出，以便在即使TwinCAT Logger已经关闭或者不可用的时候，还可以进一步分析:



历史诊断信息（Diagnose History Object）

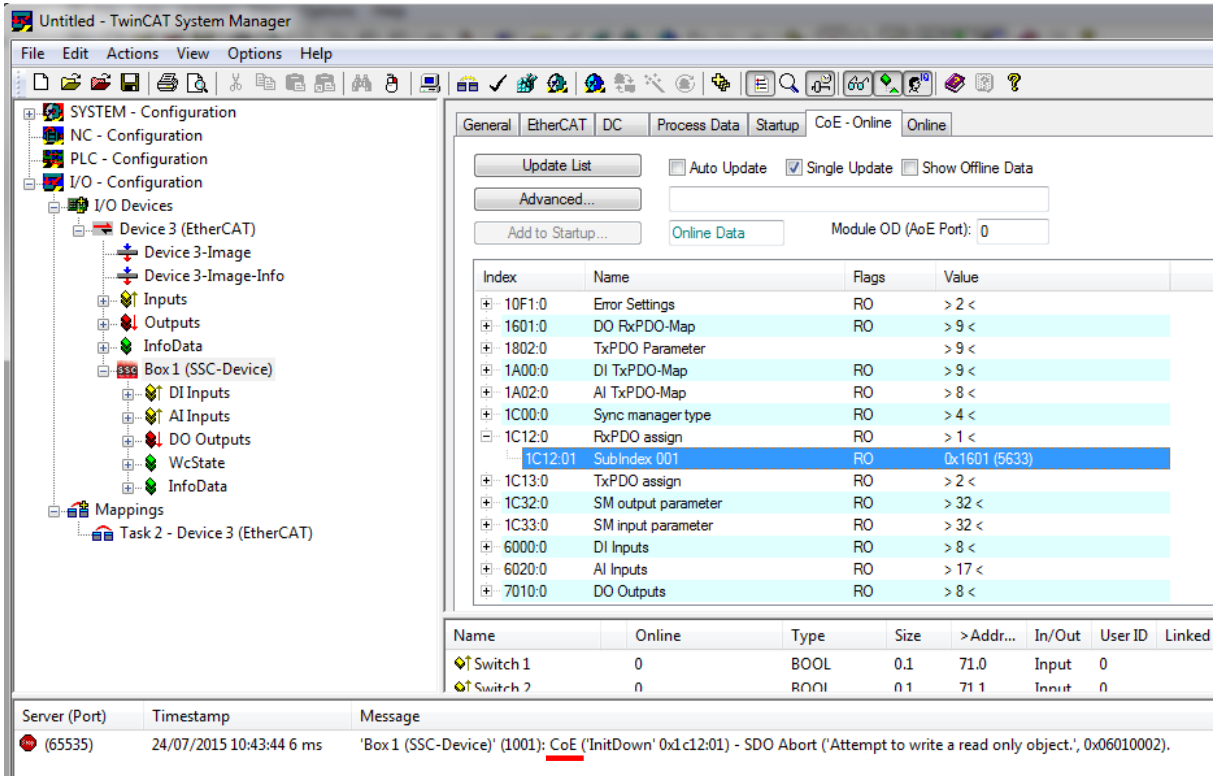
对于所有应用相关的错误，部分 CoE从站设备支持“历史诊断信息”功能 （Diagnosis History Object ）**Register 0x10F3.**

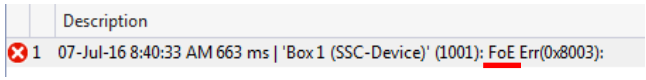
如果从站支持该功能，TwinCAT 就会额外显示一个页面 “Diag History”：



邮箱协议错误（Mailbox Protocol Errors）

邮箱协议错误（ Mailbox protocol errors ）是一个指定类的一般性软件错误，它不是厂家自定义的，不影响 EtherCAT 状态机，不会阻止也不会导致意外的状态切换。仅当执行某个特定的邮箱协议所禁止的动作时，才会发生邮箱协议错误。





发生了邮箱协议错误（Mailbox Protocol Errors ）– 怎么办?

根据特定的邮箱协议，可能的错误原因是：

- **CoE**
  - 主站试图读写从站的对象字典（Object Dictionary）中并不存在的对象
  - 主站试图对从站的某个“只读” 对象执行“写入”操作。
  - 主站试图以“完全访问”的方式访问从站的某个对象，而从站并不支持该功能。

- **FoE**
  - 文件名错误 (例如, 缺少扩展名*.xxx).
  - 从站需要密码, 但未被正确设置
  - 文件大小超出了从站能接受的范围
  - 从站不在 Bootstrap 状态

- **EoE**
  - 主站试图把从站的 IP-Add（Internal Use）ress 配置为末位为0的值（比如“x.y.z.0”），而EoE从站的Tcp/Ip 协议栈拒绝此设置

在TwinCAT PLC 中编写 EtherCAT 诊断程序

在PLC程序中系统地诊断 EtherCAT 状态

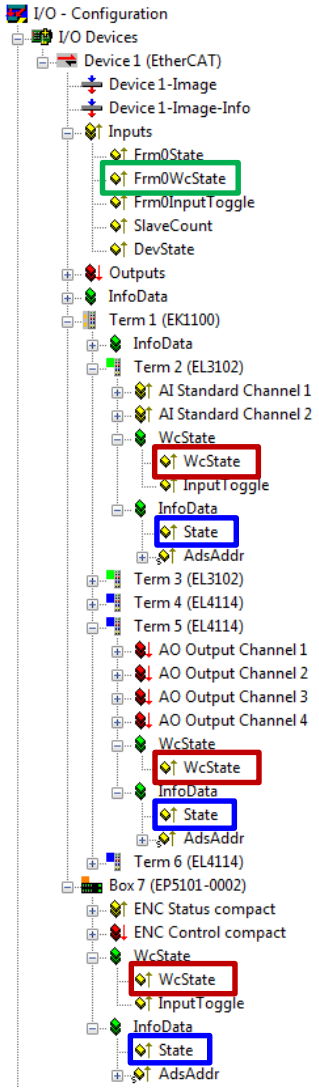
对于EtherCAT 网络，TwinCAT 自动提供了大量诊断信息，可以在PLC程序中用于检测总线通讯的错误，以便自动做出响应并报告给用户。

用户总是倾向于在PLC程序中实现最少的EtherCAT诊断，因为这样有助于通讯错误发生时节约大量的操作时间。

TwinCAT PLC 程序可以混合使用 2 种不同的诊断信息：

- **周期性信息**：默认包含在EtherCAT网络的周期性过程映像数据中的 输入数据，可以直接映射到相应的PLC输入变量 (AT %I*).
- **非周期性信息**：PLC程序调用默认库文件(TcEtherCAT.lib)中指定的功能块，可以获取这些非周期性的EtherCAT诊断信息 。

TwinCAT 中的周期性诊断信息



EtherCAT-related diagnostic information provided cyclically by TwinCAT:

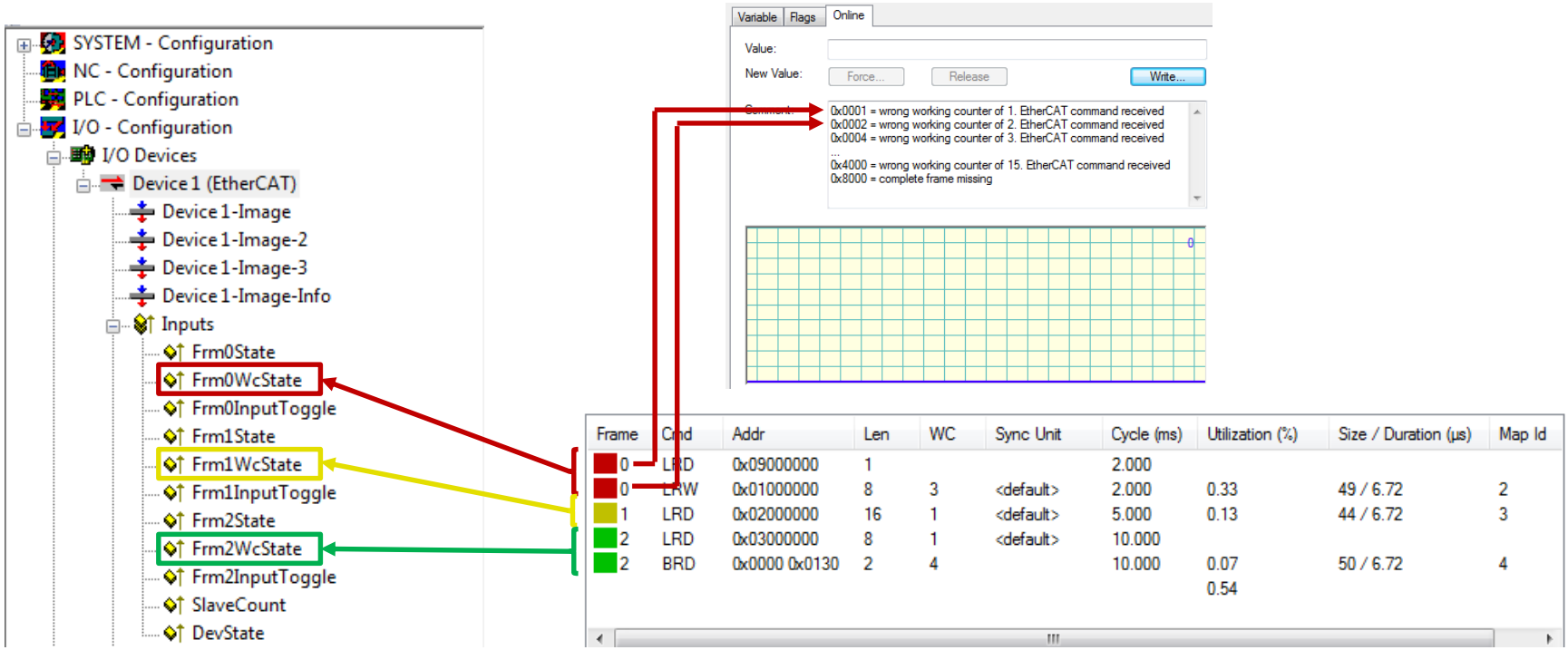
- FrmXWcState (WORD, 1 variable per frame)
- WcState (BOOL, 1 variable per slave)
- State (WORD, 1 variable per slave)

At least the WcState and State variables of all slaves should be linked to the PLC program in order to enable minimal diagnostics in the control application.

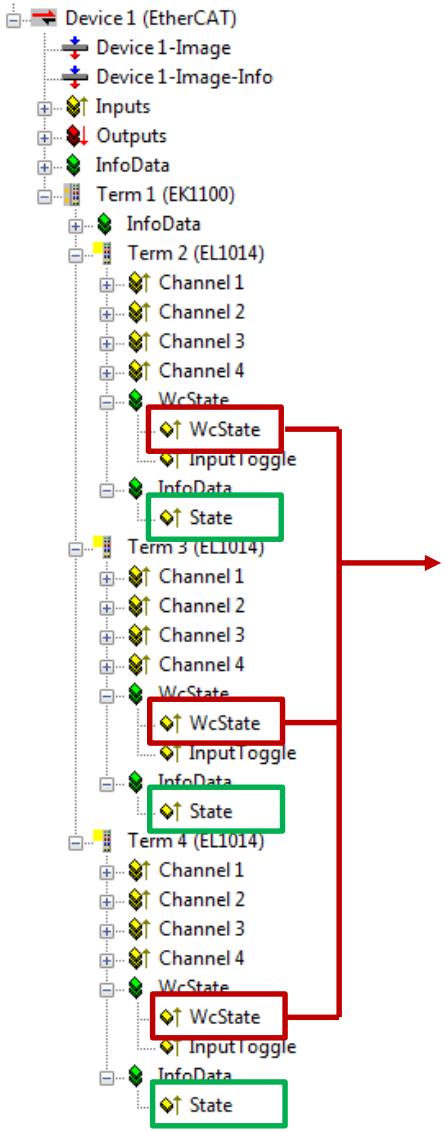
周期性信息 - FrmXWcState 变量

Allows the PLC to check the Working Counter information for each Datagram:

- Each configured Frame has its own 16-bit **FrmXWcState** variable
- Each **bit** in the variable corresponds to a specific Datagram within the Frame
- The bit is set if the corresponding Datagram has a wrong Working Counter



周期性信息 - WcState 变量



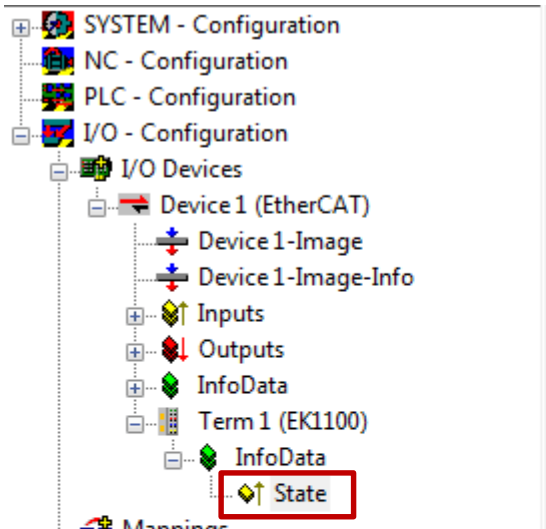
- Boolean variable in the process image of each slave.
- If a datagram has a wrong Working Counter value, WcState variable is set for **all** the slaves Add (Internal Use) ressed by that datagram.
- A slave with WcState = 1, therefore, is not necessarily the cause of the problem. Add (Internal Use) itional information will be provided by the 16-bit **State** variable of slaves with WcState = 1.

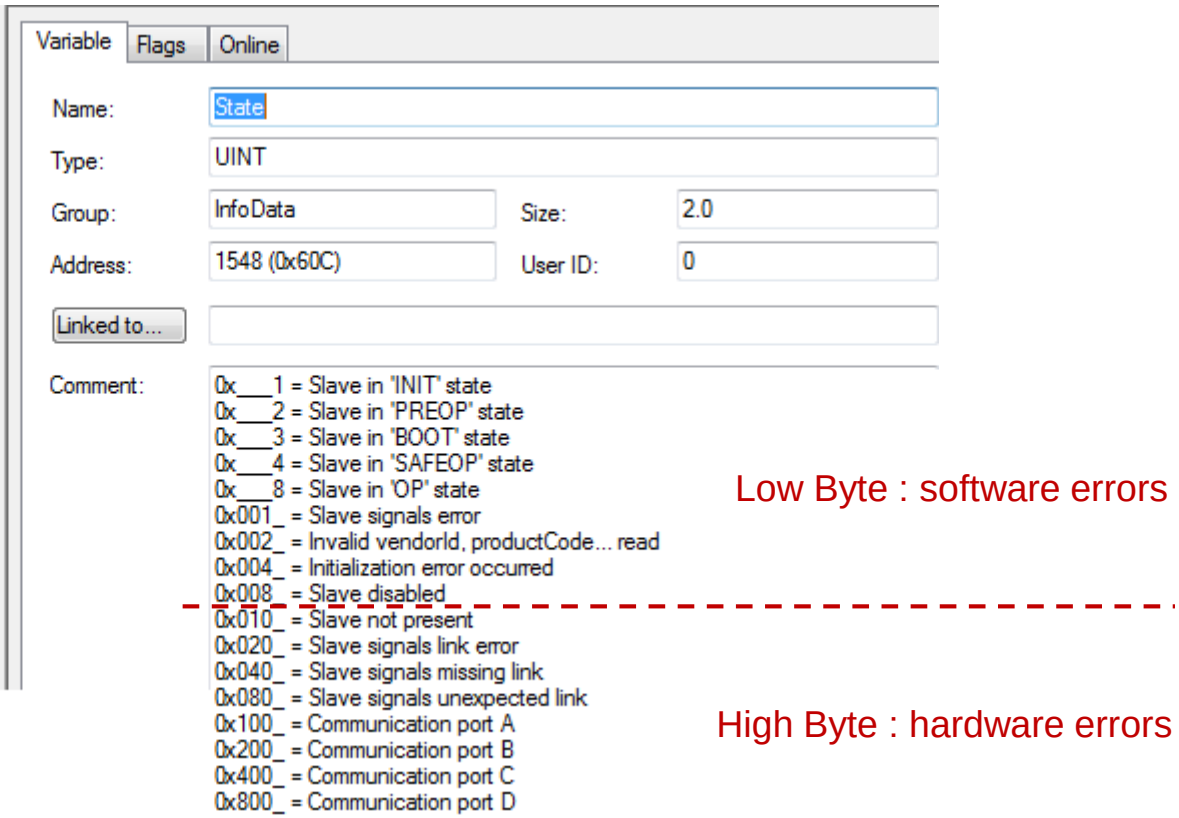
Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
0	LRD	0x01000000	2	3	<default>	2.000			
0	BRD	0x0000 0x0130	2	4		2.000	0.33 0.34	44 / 6.72	0

(Example: in case of wrong Working Counter for the LRD datagram, all Slaves Add (Internal Use) ressed by this datagram will set WcState = 1. Reading the State variable of these slaves the master application can investigate which slave is responsible for the error).

周期性信息 - State 变量

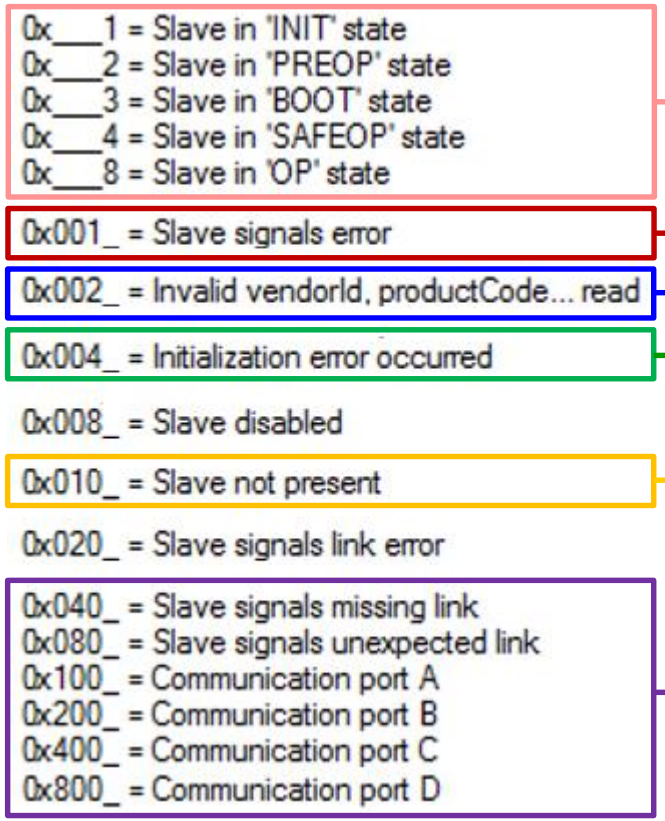
Through this variable the EtherCAT Master summarizes the diagnostic information collected from the network, cyclically reports several error conditions:





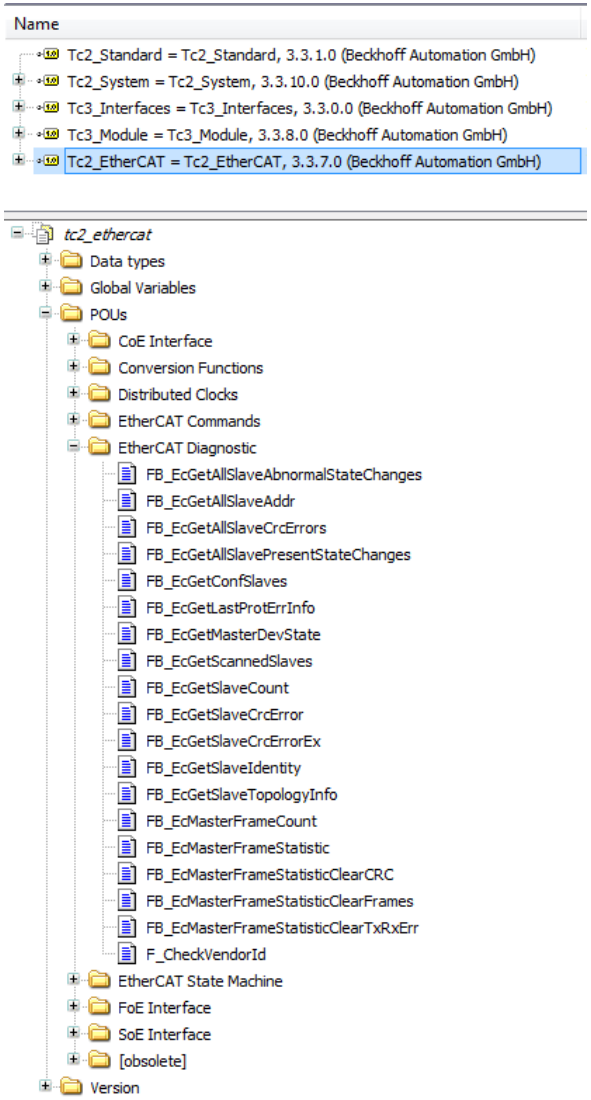
周期性信息 – State 变量

Examples of error diagnostic information reported by the State variable are:



- Current state of the slave (also without error)
- An error in the EtherCAT State Machine occurred and the AL Status Code is != 0
- Corresponds to VPRS error
- The error occurred during slave initialization
- Corresponds to NO_COMM error
- Corresponds to LINK_Add (Internal Use) X and LINK_MIS X

非周期性信息 – TcEtherCAT PLC Library

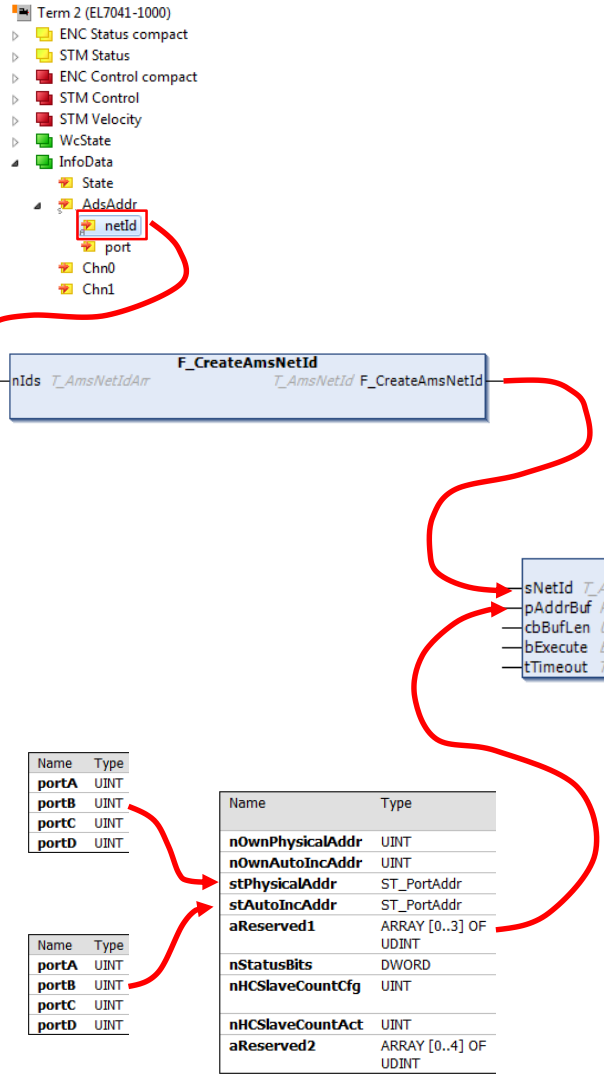


The **TcEtherCAT.lib** library (available free-of-charge with both the default installation of TC2 and TC3) provides function-blocks which enable to acyclically diagnose the EtherCAT network

- Frame statistics
- CRC statistics
- Slave Identity
- Number and list of configured Slaves
- Number and list of found Slaves

The library basically allows to automatically read from the PLC program all the information which is online visible in the TwinCAT project.

非周期性信息 – TcEtherCAT PLC Library

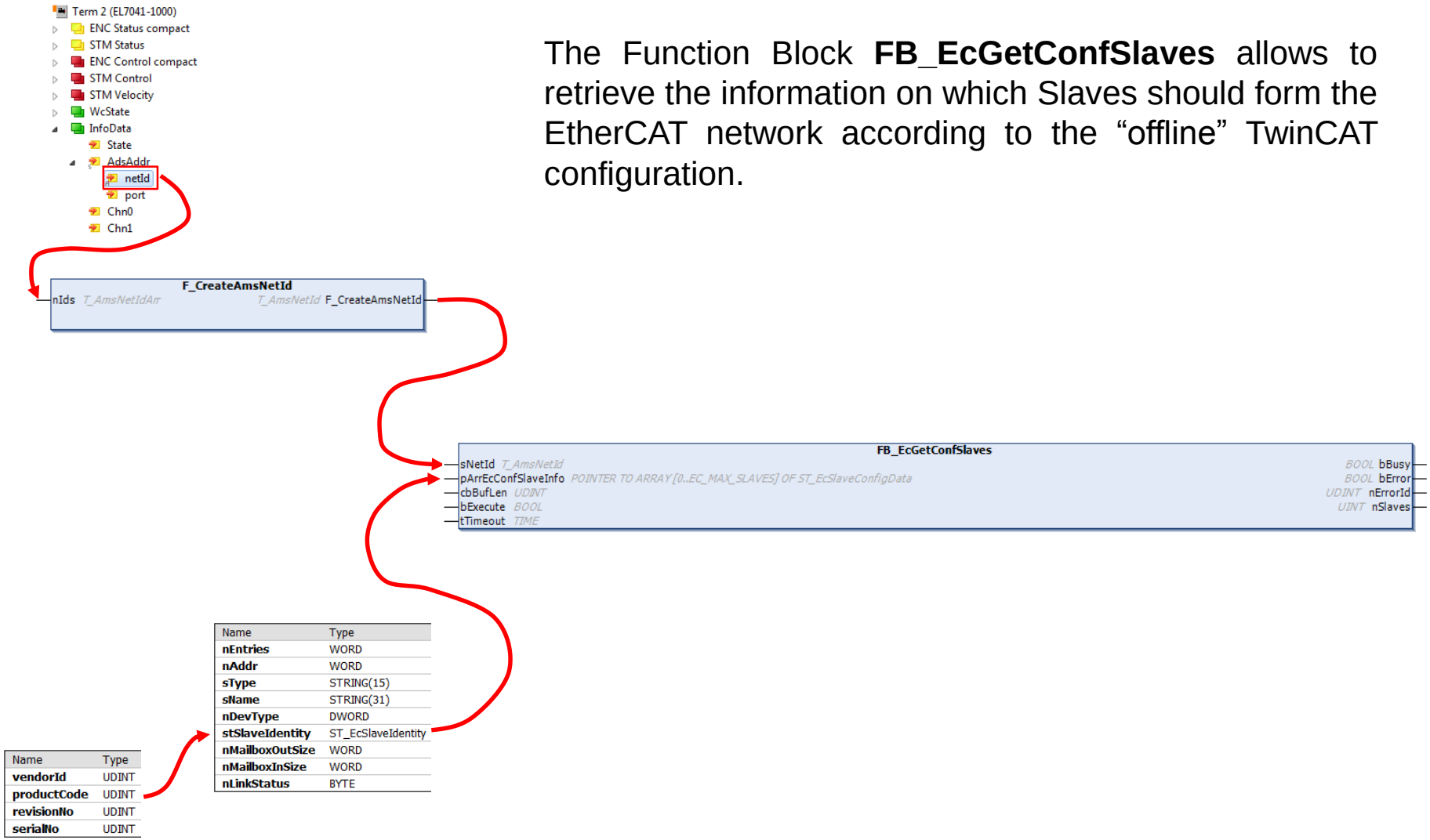


The Function Block **FB_EcGetSlaveTopologyInfo** returns information on how the ports of different slaves are connected to each other, and therefore the structure of the (configured) network topology.

(If Hot Connect groups are present in the configuration, they will be listed at the and of the ARRAY).

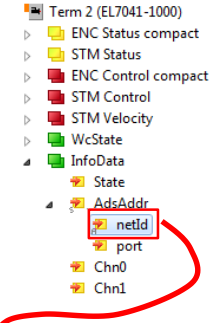
- **nOwnPhysicalAdd (Internal Use) r**: Physical Add (Internal Use) ress of Slave N
- **nOwnAutoIncAdd (Internal Use) r**: Auto Increment Add (Internal Use) ress of Slave N
- **stPhysicalAdd (Internal Use) r**: structure with the Physical Add (Internal Use) ress of Slaves connected to the different Ports of Slave N
- **stAutoIncAdd (Internal Use) r**: structure with the Auto Increment Add (Internal Use) ress of Slaves connected to the different Ports of Slave N

非周期性信息 – TcEtherCAT PLC Library

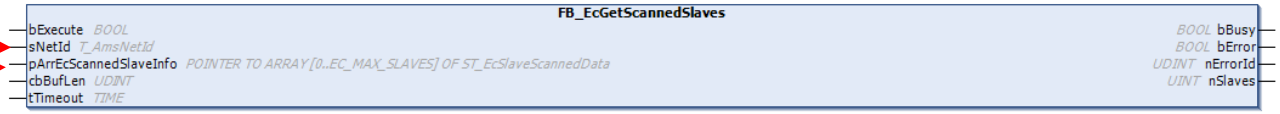


The Function Block **FB_EcGetConfSlaves** allows to retrieve the information on which Slaves should form the EtherCAT network according to the “offline” TwinCAT configuration.

非周期性信息 – TcEtherCAT PLC Library

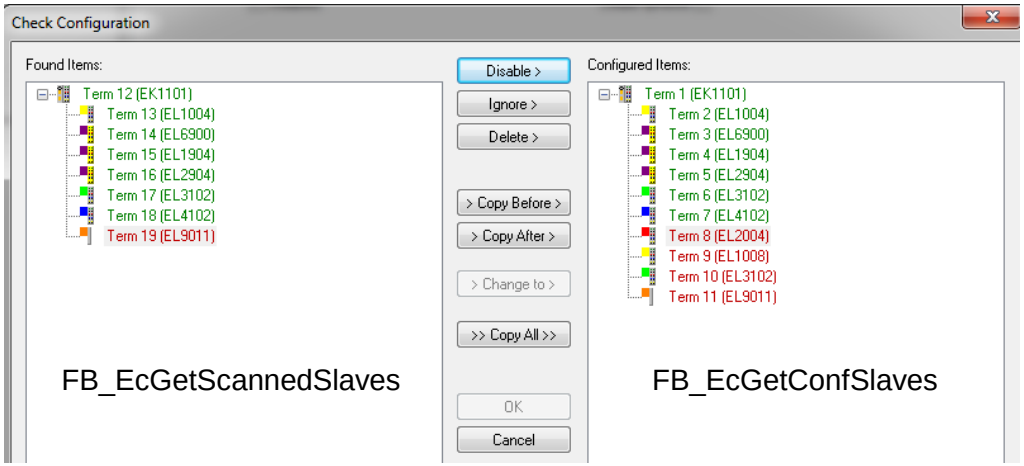


The Function Block **FB_EcGetScannedSlaves** allows to retrieve the information on which Slaves are actually detected “online” on the network. This information can be compared to the “offline” expected configuration retrieved with *FB_EcGetConfSlaves*.



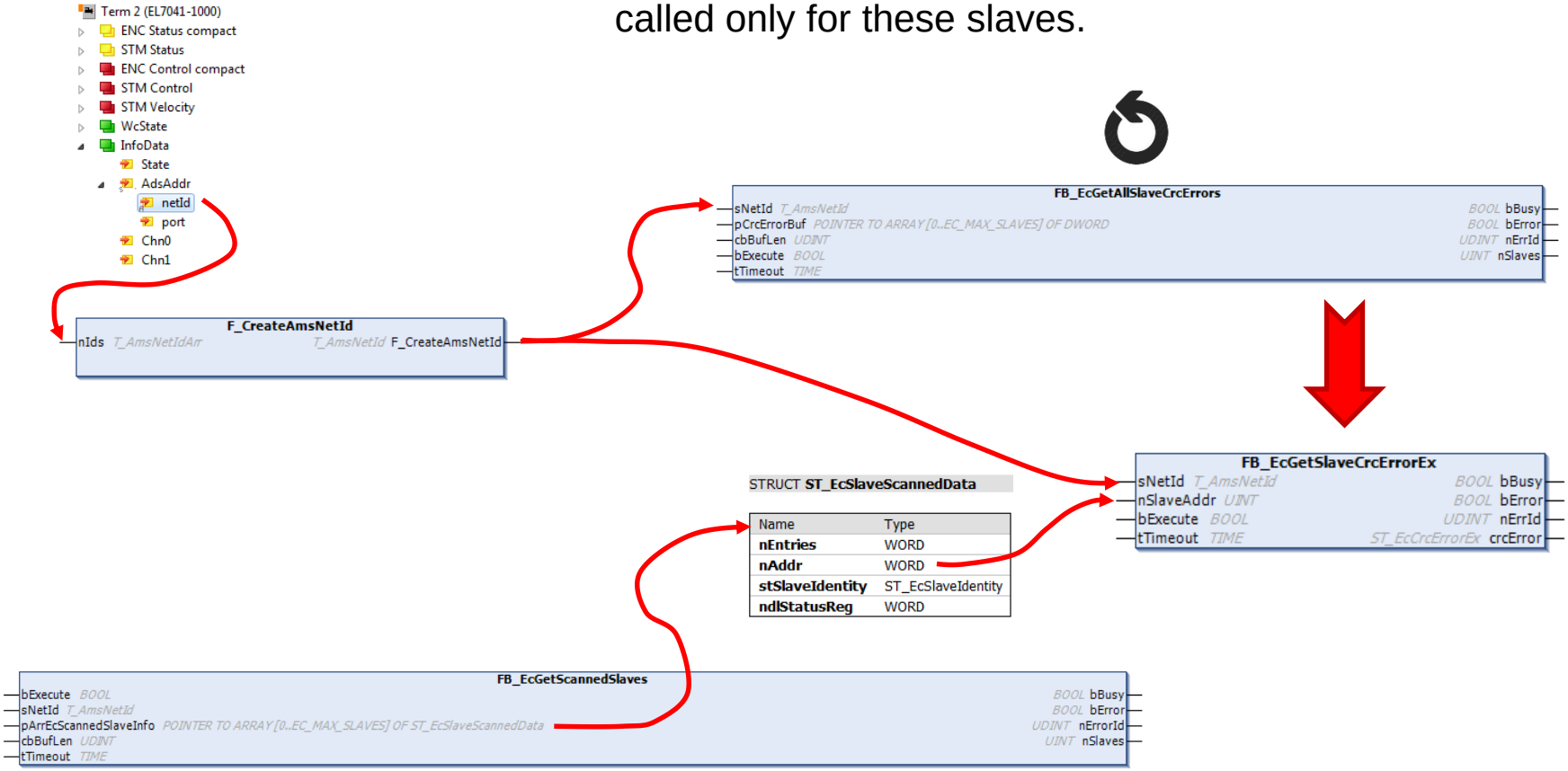
Name	Type
nEntries	WORD
nAddr	WORD
stSlaveIdentity	ST_EcSlaveIdentity
ndlStatusReg	WORD

Name	Type
vendorId	UDINT
productCode	UDINT
revisionNo	UDINT
serialNo	UDINT

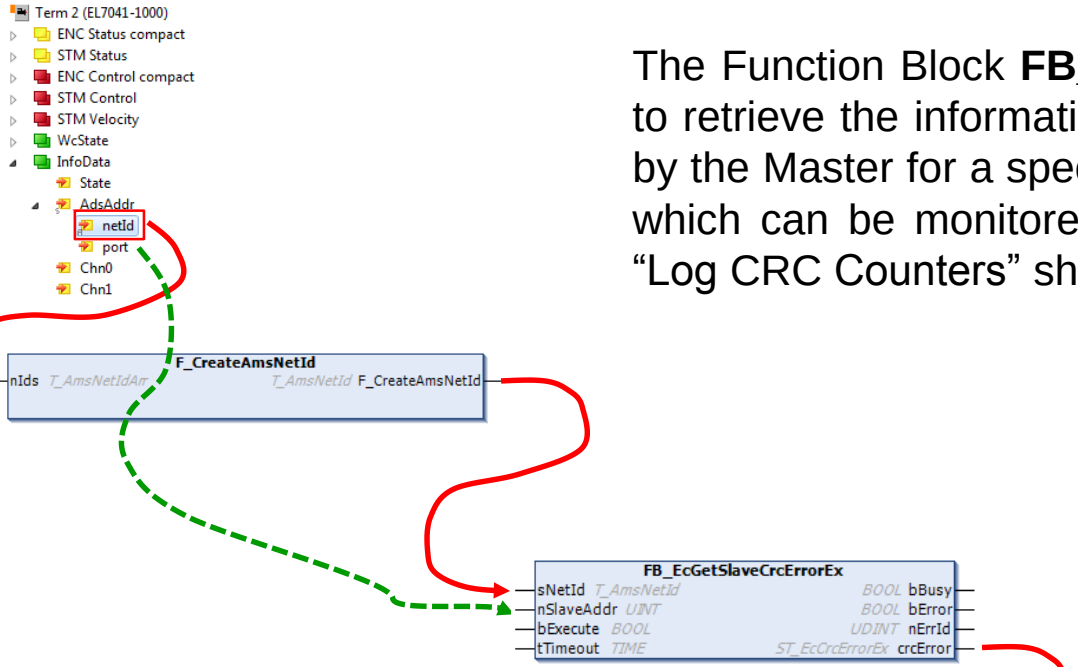


非周期性信息 – TcEtherCAT PLC Library

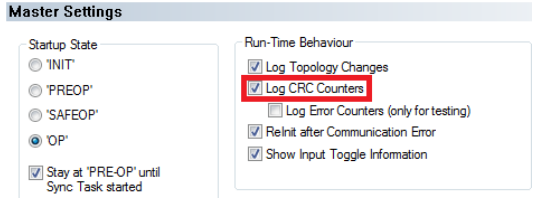
The Function Block **FB_EcGetAllSlaveCrcErrors** can be periodically called in order to determine if one or more slaves report CRC errors. In this case, the Function Block **FB_EcGetSlaveCrcErrorEx** can be called only for these slaves.

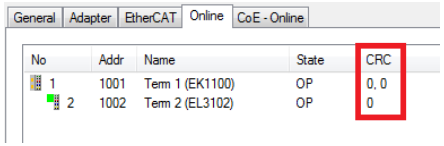


非周期性信息 – TcEtherCAT PLC Library

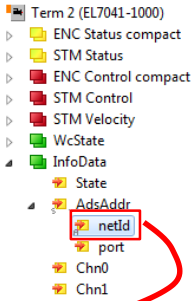


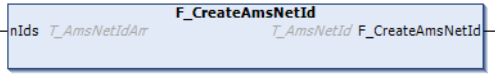
The Function Block **FB_EcGetSlaveCrcErrorEx** allows to retrieve the information about the CRC errors logged by the Master for a specific Slave (the same information which can be monitored in the TwinCAT project). Flag “Log CRC Counters” shall be active!





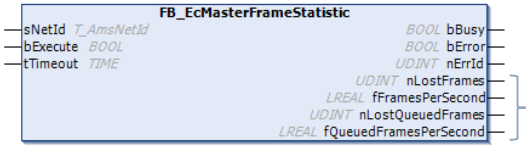
非周期性信息 – TcEtherCAT PLC Library

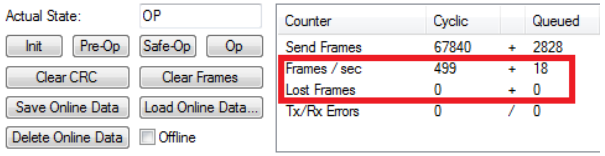




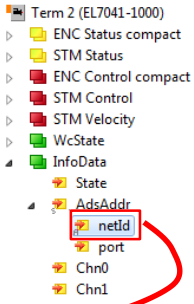
The Function Block **FB_EcMasterFrameStatistics** allows to retrieve the information about Lost Frame detected by the Master (the same information which can be monitored in the TwinCAT project).

获取主站的丢失数据帧信息

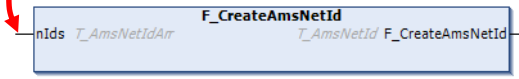




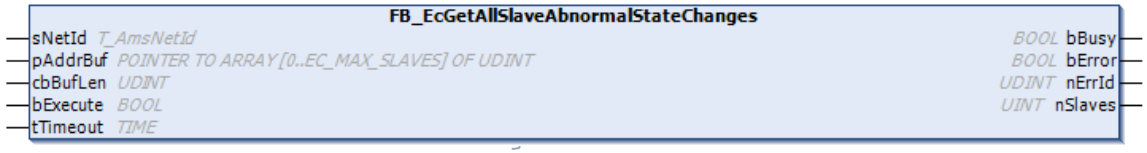
非周期性信息 – TcEtherCAT PLC Library



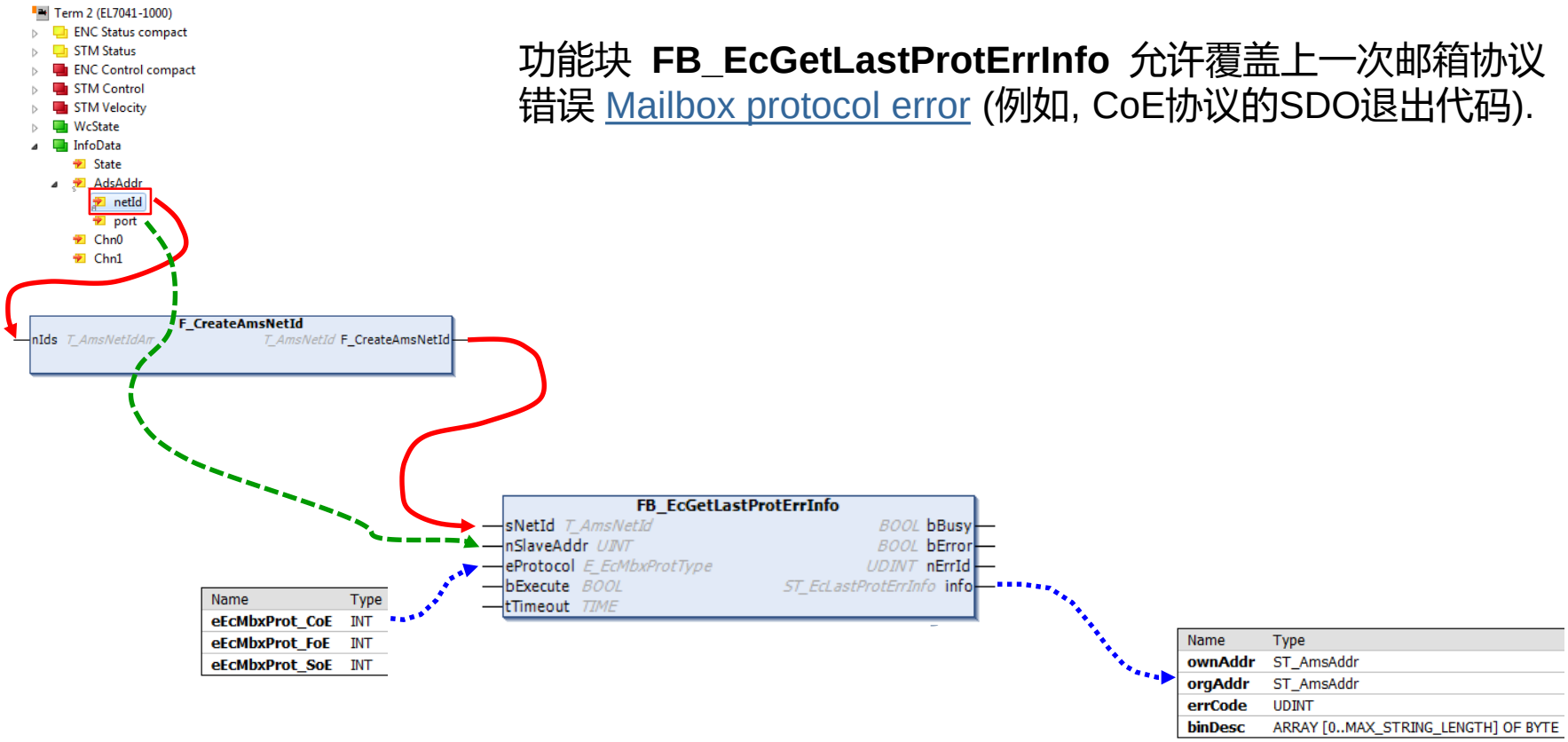
The Function Block **FB_EcGetAllSlaveAbnormalStateChanges** counts the number of unexpected state transitions from OP to SafeOP for each slave (the specific AL Status Code associated to each unexpected transition is not reported).



获取所有从站的异常状态切换



非周期性信息 – TcEtherCAT PLC Library



功能块 **FB_EcGetLastProtErrInfo** 允许覆盖上一次邮箱协议错误 Mailbox protocol error (例如, CoE协议的SDO退出代码).

The next error-free Mailbox access to the same slave deletes the error memory!
对同一从站的下一次正确邮箱访问会清除此前记录错误信息的内存区。

附件:

使用 Wireshark 进行诊断

如果软件诊断还不够 – 使用 Wireshark

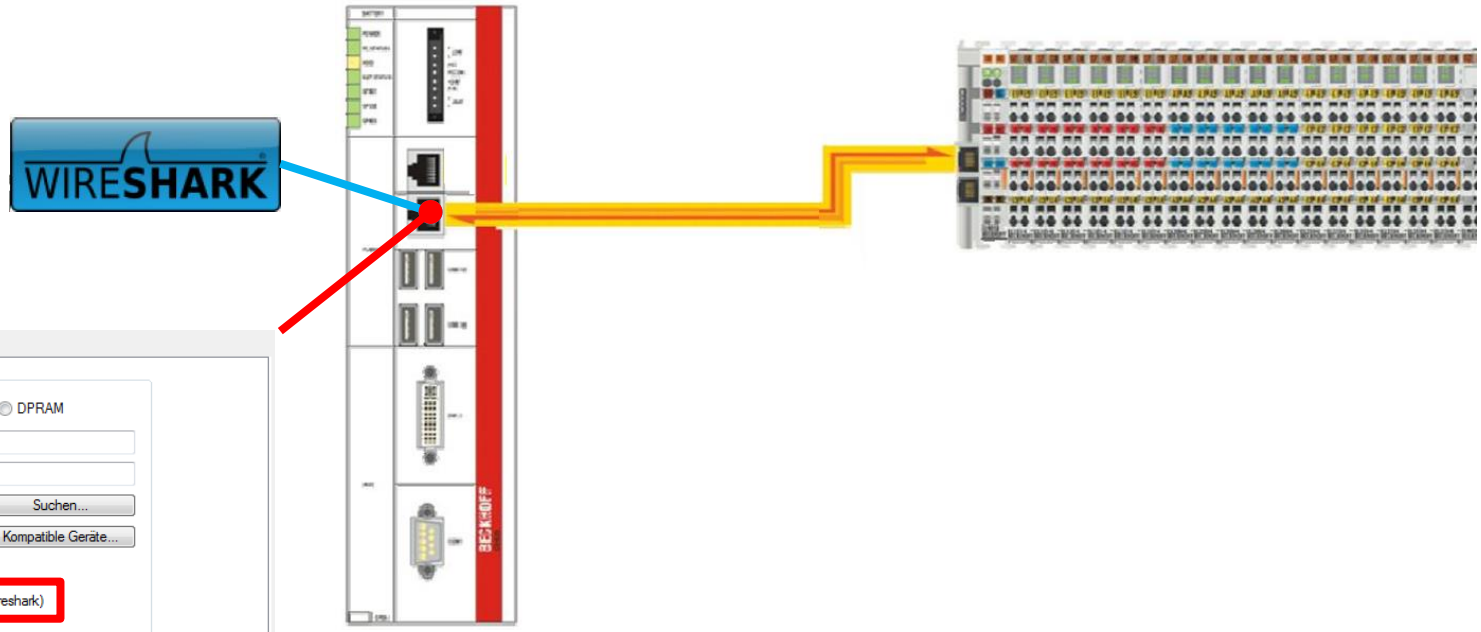
Wireshark 并不是用以取代TwinCAT中的诊断功能 。对于用户来说，应当在使用了其它诊断方法都无法找到故障点的时候，才使用这个“**终极武器**”。

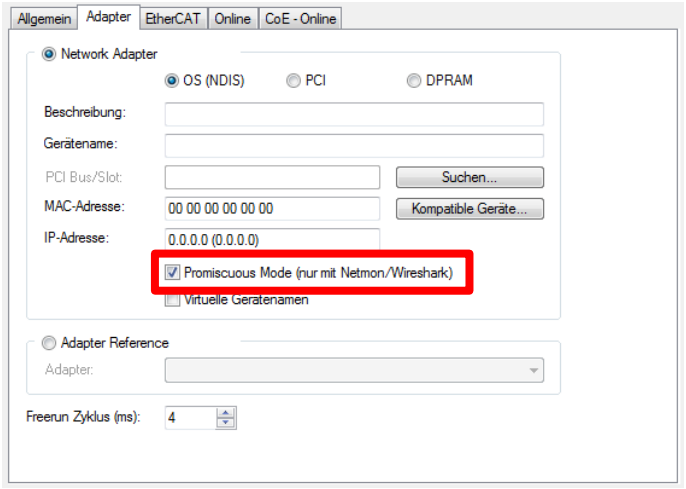
Technical data

Technical data	ET2000
Bus system	Ethernet (all Ethernet based protocols (IEEE 802.3)
Number of ports/channels	8/4
Ethernet interface	100BASE-TX Ethernet with RJ45
Uplink port	1 GBit/s
Baud rate	Probe ports: 100 MBit/s, Uplink port: 1 GBit/s
Delay	< 1 µs
Resolution time stamp	1 ns
Accuracy time stamp	40 ns
Diagnosis	2 LEDs per channel - Link/Activity 8 Status LED for future diagnostics
Supply voltage	24 V_{DC} (18 V_{DC} ... 30 V_{DC})
Software interface	"Wireshark" extension currently required "Wireshark" version

为了执行 Wireshark 抓包，有多种不同的硬件配置。

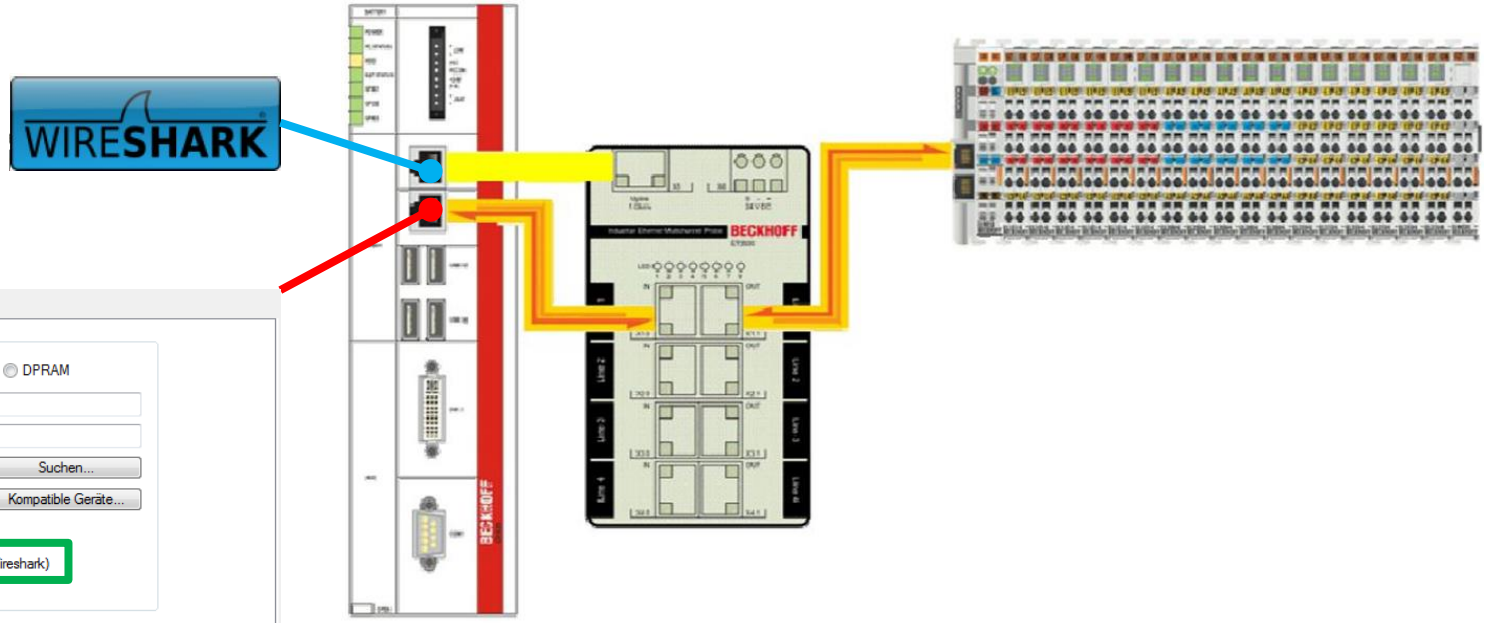
Wireshark 抓包的硬件配置 1

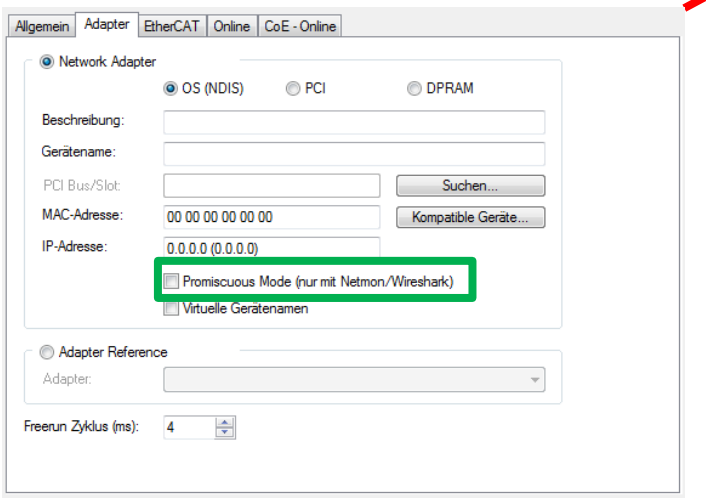




- 不支持 Windows CE
- 没有精确的时间戳

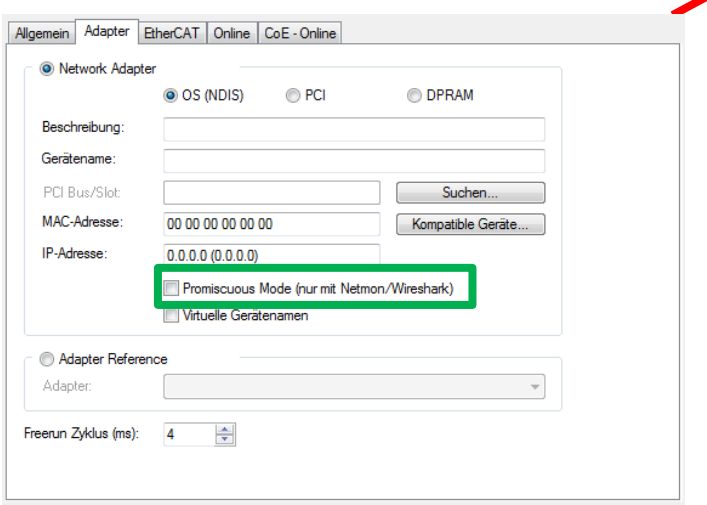
Wireshark 抓包的硬件配置 2

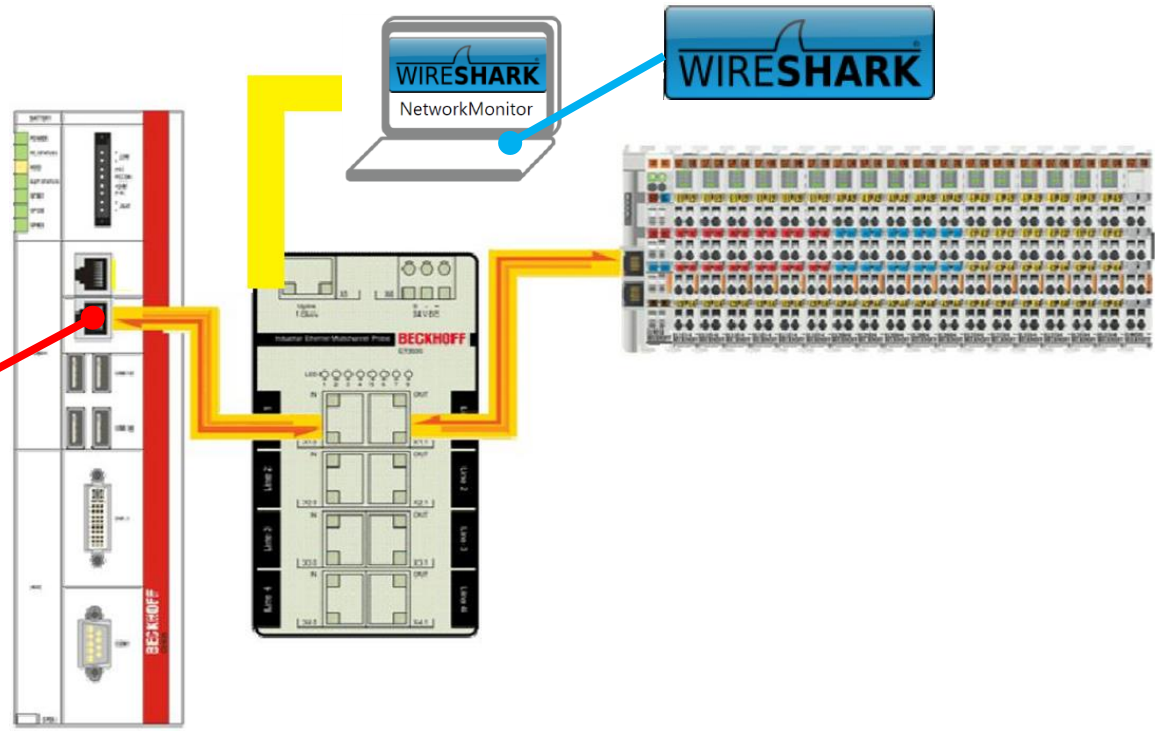




- 不支持 Windows CE
- 有精确的时间戳

Wireshark 抓包的硬件配置 3



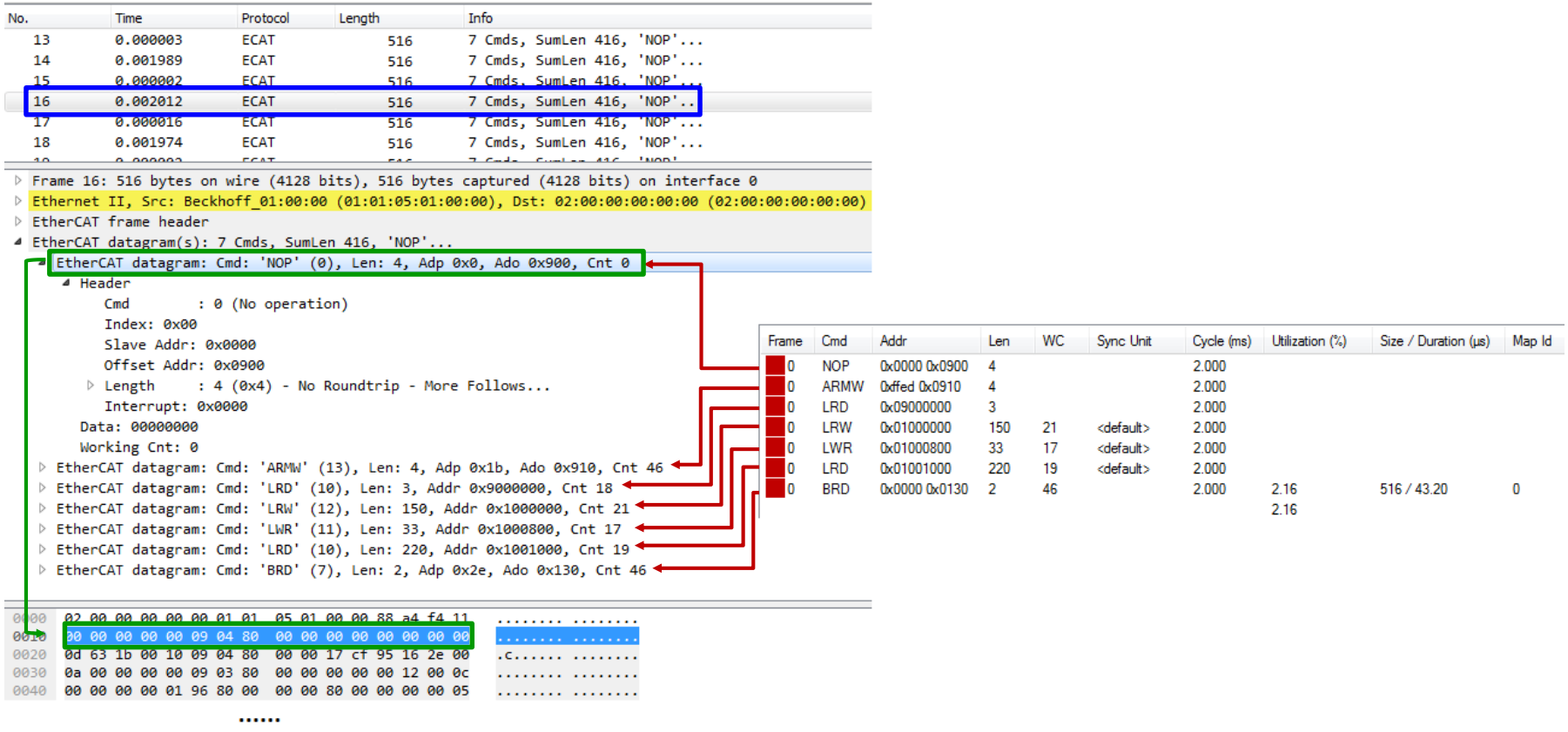


- 支持 Windows CE
- 有精确的时间戳

Wireshark 示例

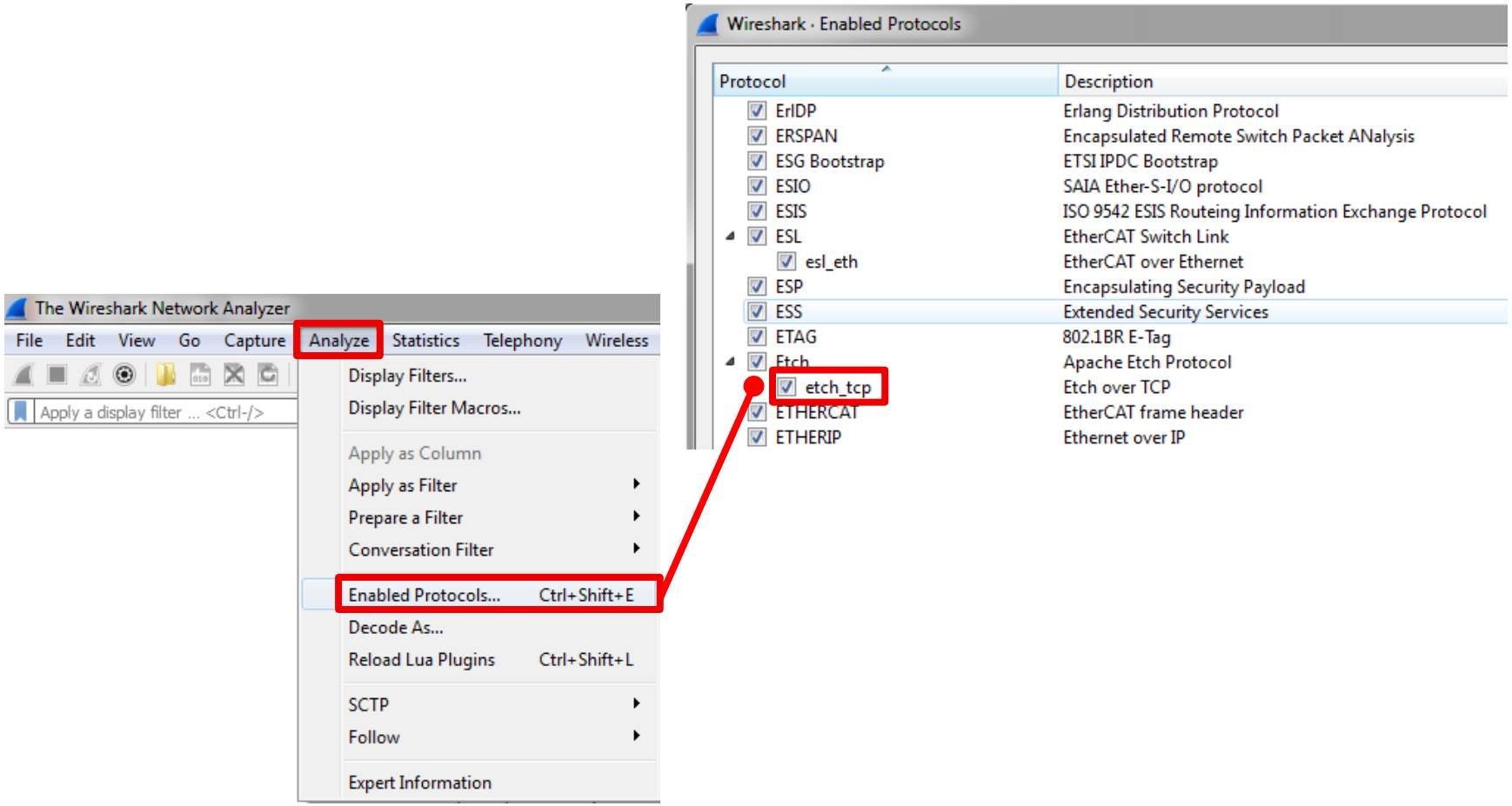
Wireshark 允许追踪每个 EtherCAT 数据帧的 **list**, **structure** and **content** of (在一轮捕捉过程中，每个数据帧会被捕获两次!).

下图显示了捕获的一个周期性数据帧：



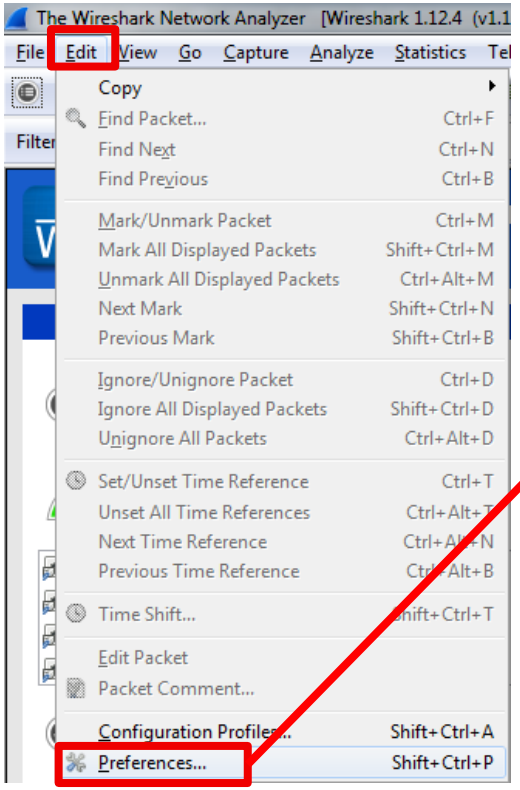
Wireshark – 时间戳协议(仅当使用ET2000时)

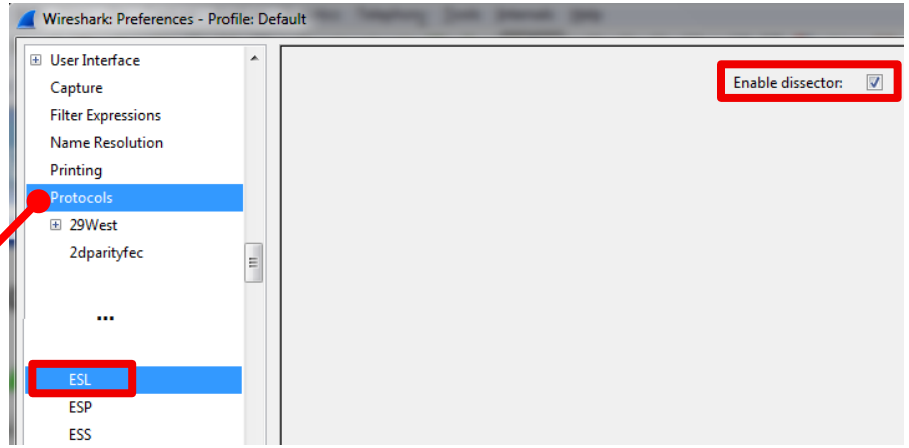
使用ET2000时，为了在Wireshark中获取到极为精确的时间戳（time-stamping）信息，应当启用对ESL协议的解析功能。Wireshark Version 2 支持此功能。



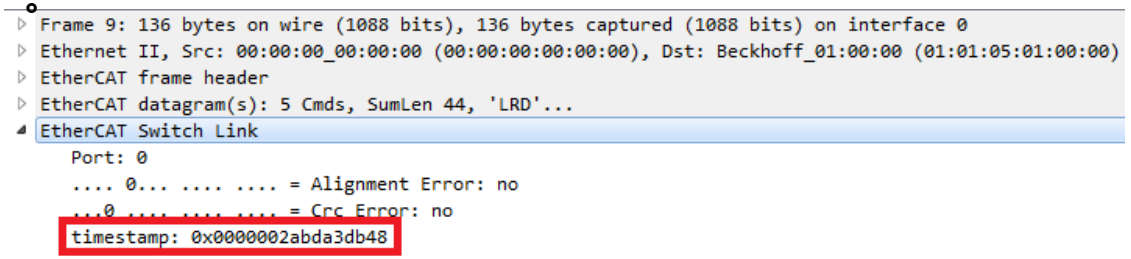
Wireshark – 时间戳协议(仅当使用ET2000时)

…在早期的 Wireshark **Version 1**中操作如下：.



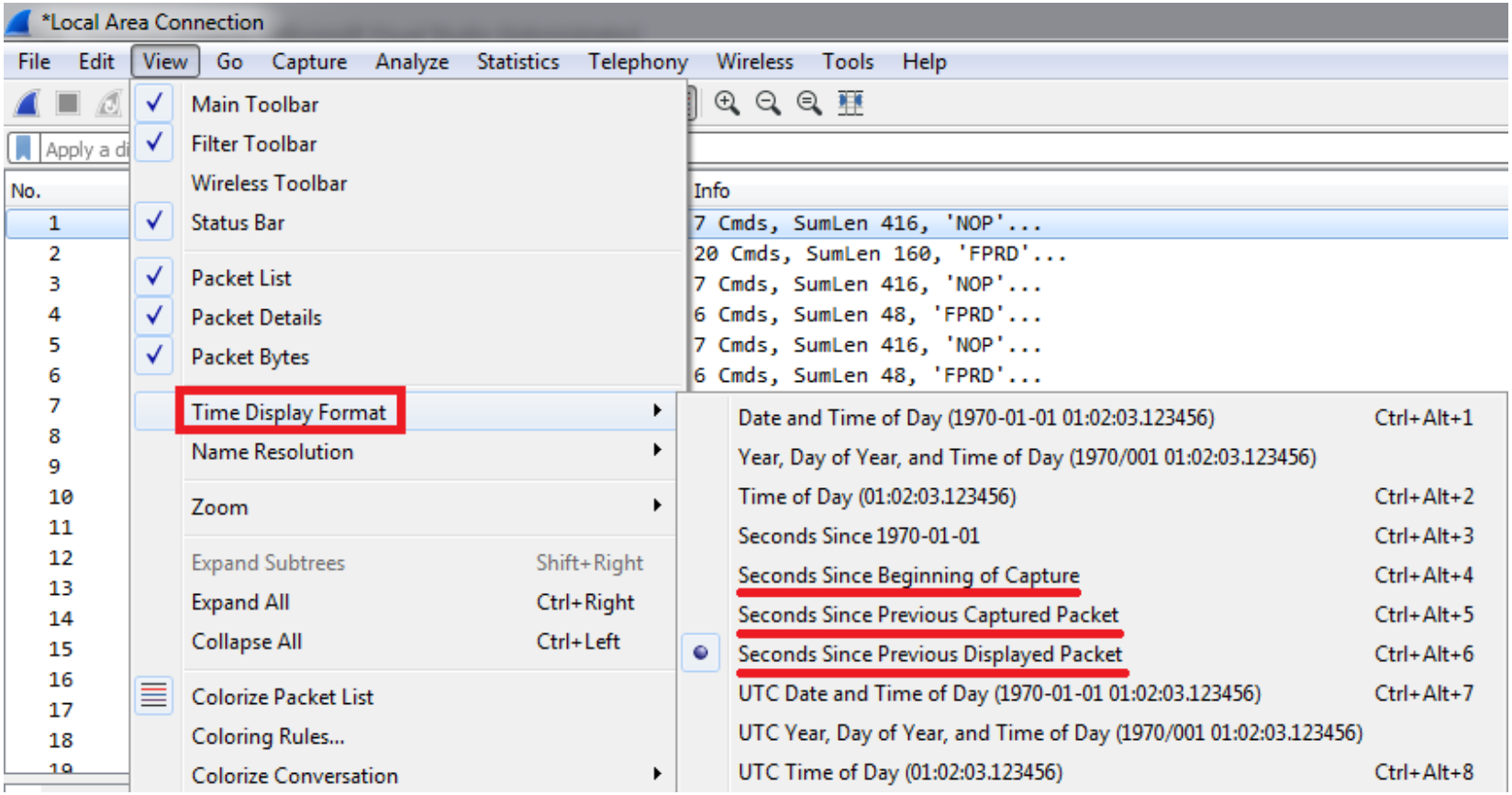


两个版本都在数据帧内容的末尾提供时间戳
(每个增量表示 1 ns).



Wireshark – 显示时间的格式

使用Wireshark追踪时，用户可以选择最合适的 **时间信息** 显示格式:



Wireshark – 部分过滤选项（Filter Options）

捕获的数据帧可以使用过滤功能，可用的过滤选项包括:

- **仅捕获周期性数据帧（Cyclic Frame）**

  Filter: (ecat.cmd == 0x0a) || (ecat.cmd == 0x0b) || (ecat.cmd == 0x0c)

- **仅捕获邮箱通讯（mailbox communication）**

  Filter: ecat_mailbox

  Filter: ecat_mailbox.coe

  Filter: ecat_mailbox.soe

  Filter: ecat_mailbox.foe

  Filter: ecat_mailbox.eoe

- **仅访问特定的注册字或者注册字区间 （register or register range）**

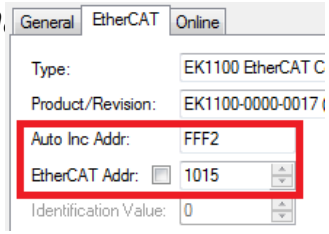
  Filter: ecat.ado == *<register_Add（Internal Use）ress>*

  Filter: (ecat.ado >= *<lower_Add（Internal Use）ress>*) && !(ecat.ado >= *<upper_Add（Internal Use）ress>*)

Wireshark – 部分过滤选项（Filter Options）

- **仅捕获特定一个从站或者几个连续地址的从站的数据报文**

  Filter: ecat.adp == <*slave_Add（Intern*



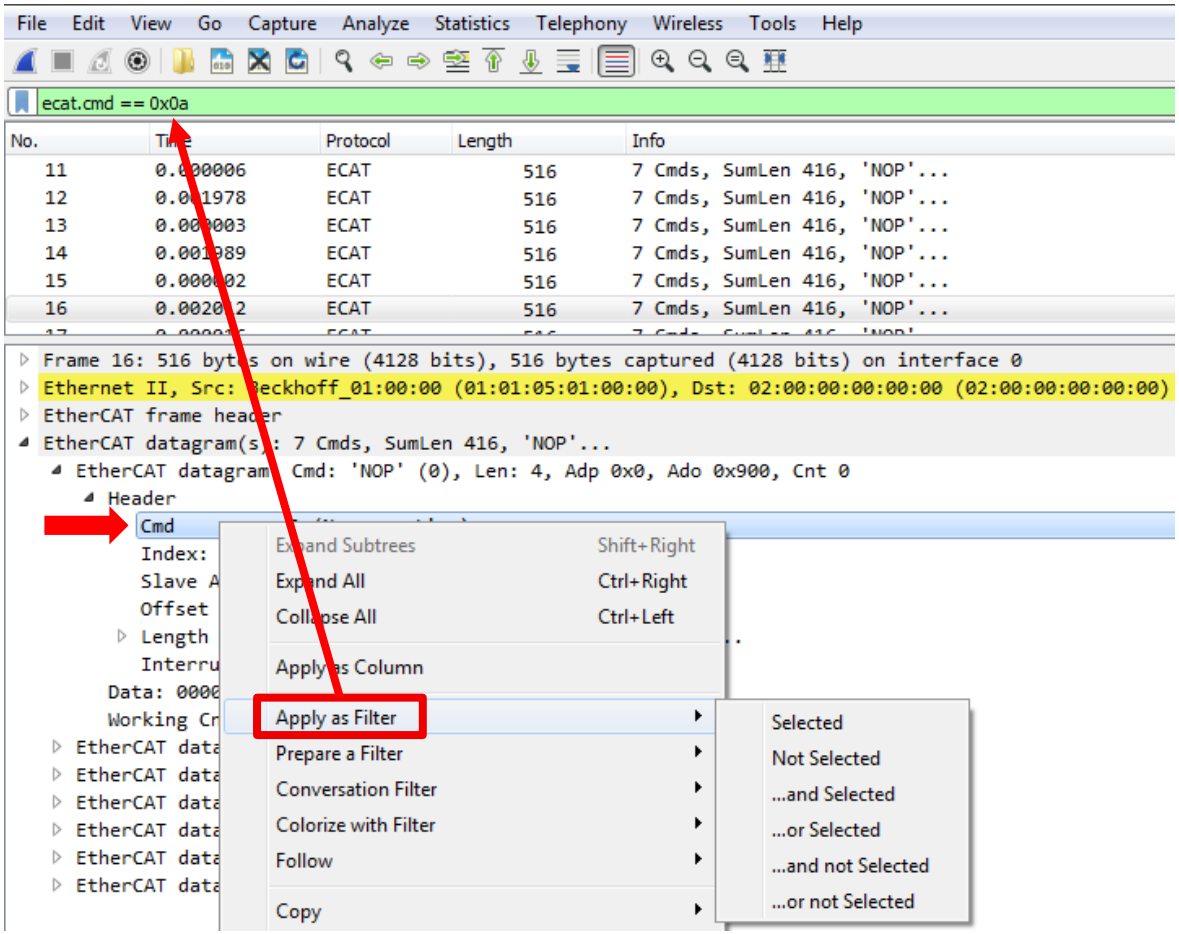
过滤选项还可以**自由组合** 成逻辑表达式：

Operator	Meaning
&&	AND
||	OR
!	NOT
==	equal
>=	greater or equal
<=	smaller or equal
>	greater
<	smaller

其它过滤选择可参考：https://www.wireshark.org/docs/dfref/e/ecat.html

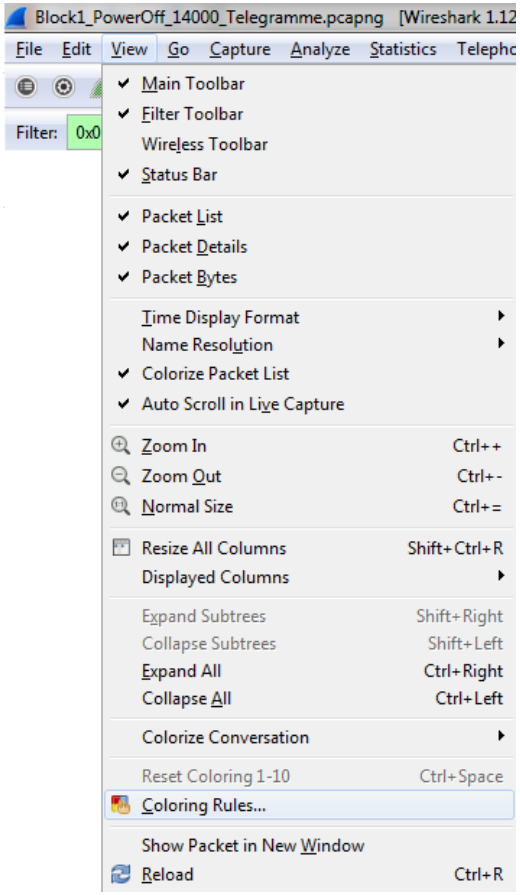
Wireshark – 部分过滤选项（Filter Options）

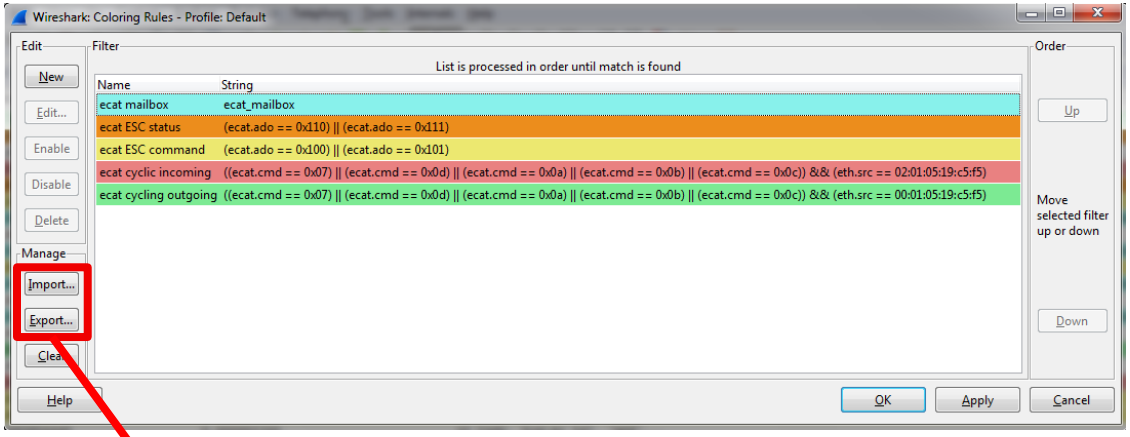
点击“**Apply as Filter**”，可以从追踪记录中直接提取过滤选项。



Wireshark – 颜色标记的规则（Coloring Rules）

为了更好显示不同的信息类型，可以为特定的过滤协议指定颜色标记规则（ **colouring rules** ）

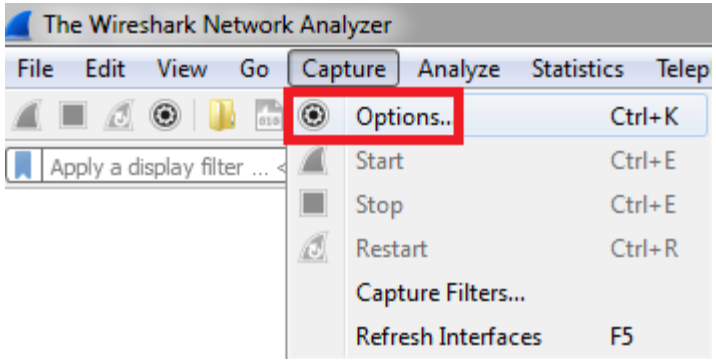


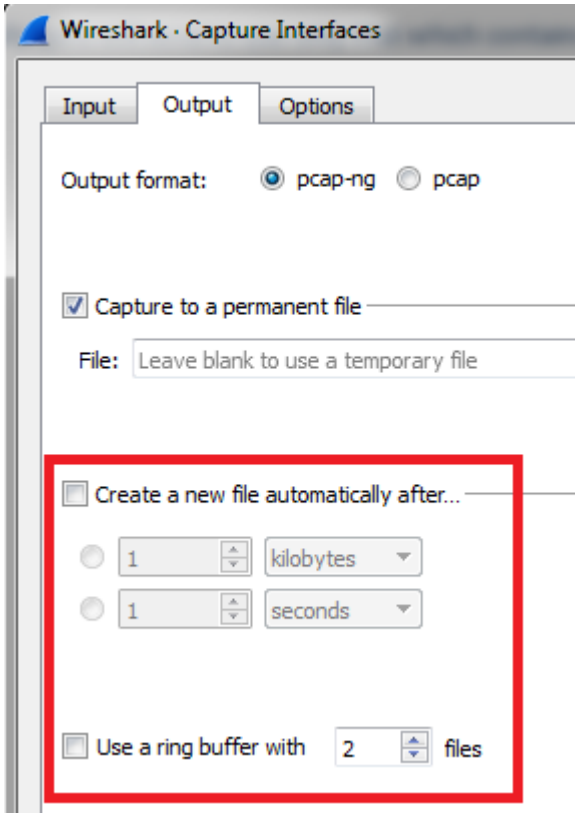


一旦定义了颜色标记规则，此后这台PC上所有的Wireshark追踪都会应用这套规则。通过导出和导入功能，一台机的颜色标记规则还可以移植到另一台PC上。

Wireshark – 缩小保存的捕捉记录文件

在开始捕捉之前，可以设置Wireshark把捕获的信息到指定大小的独立文件，而不是把所有数据帧保存在一个单独的文件里面：





Wireshark – 缩小保存的捕捉记录文件

捕捉完成后, 还可以只保存选择的部分数据帧: