

作者： 陈利君
 职务： 华南区 技术工程师
 日期： 2019-10-23
 邮箱： l.chen@beckhoff.com.cn
 电话： 020-38010300-811（可选）

BECKHOFF New Automation Technology

中国上海市静安区汶水路 299 弄 9-10 号
 市北智汇园 4 号楼（200072）

TEL: 021-66312666
 FAX: 021-66315696

解析 NC 轴 18005 错误及 AX5000 驱动 F415 故障

摘要：使用 TwinCAT NC 控制伺服驱动器时 NC 轴报警 0x4655（18005），同时倍福的伺服驱动器 AX5000 报警 0xF415，或者第三方驱动器报错“同步丢失”。18005 和 F415，可能是同一个硬件故障导致的分别在 EtherCAT 主站侧和从站侧触发的报警代码。本文阐述故障触发的机制，列举可能的原因，并提供故障排查的方法和顺序。本文也适用于普通 IO 从站 WcState 出现 Data Invalid 的异常处理。

关键字：同步丢失，Sync Lost，18005，F415，WcState，SyncUnit，Data Invalid，EtherCAT

附件：

序号	文件名	备注

历史版本：

2019.10.17	邱少彬	F415.docx

免责声明：

我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，也欢迎您提出改进建议。

参考信息：

目 录

解析 NC 轴 18005 错误及 AX5000 驱动 F415 故障	3
1 NC 轴故障 0x4655 (18005) 的触发机制	3
1.1 报警信息和相关现象	3
1.2 什么情况下 IO 从站会刷新失败	4
1.2.1 什么情况下 Frame 没来?	4
1.2.2 什么情况下 Frame 迟到?	4
1.2.3 Frame 按时到达, 为什么 Slave 刷新不了数据?	5
1.3 为什么 PLC 不报“Invalid IO data”?	5
2 伺服报警 Sync Lost (0xF415) 的触发机制	5
2.1 EtherCAT 同步的机制	5
2.1.1 EtherCAT Master	5
2.1.2 EtherCAT slave controller (ESC)	6
2.2 故障示例	7
2.2.1 Sample 1	7
2.2.2 Sample 2	9
3 现场排故的基本方法和顺序	10
3.1 使用诊断工具分析故障性质	10
3.2 硬件故障导致 EtherCAT Frame 出错或者丢失	12
3.2.1 硬件故障排查基本要求	12
3.2.2 软件基本配置检查	13
3.2.3 软件配置的基本优化	14
3.2.4 EtherCAT 链路优化	14
3.2.5 其它说明	15
3.3 软件原因导致的 EtherCAT Frame 迟到	15
3.3.1 EtherCAT 主站和从站的 Time Shift 优化	15
3.3.2 优先级不合理的解决办法	17
3.3.3 程序计算量波动的解决办法	17
3.3.4 DC 主时钟 (Reference Clock) 紊乱	20

解析 NC 轴 18005 错误及 AX5000 驱动 F415 故障

Version 1.1

倍福广州 陈利君 2019-10-23

适用范围:

使用 TwinCAT NC 控制伺服驱动器时 NC 轴报警 0x4655 (18005)，同时倍福的伺服驱动器 AX5000 报警 0xF415，或者第三方驱动器报错“同步丢失”。18005 和 F415，可能是同一个硬件故障导致的分别在 EtherCAT 主站侧和从站侧触发的报警代码。本文阐述故障触发的机制，列举可能的原因，并提供故障排查的方法和顺序。本文也适用于普通 IO 从站 WcState 出现 Data Invalid 的异常处理。

1 NC 轴故障 0x4655 (18005) 的触发机制

1.1 报警信息和相关现象

(1) 故障信息

Invalid IO data for more than 'n' continuous NC cycles

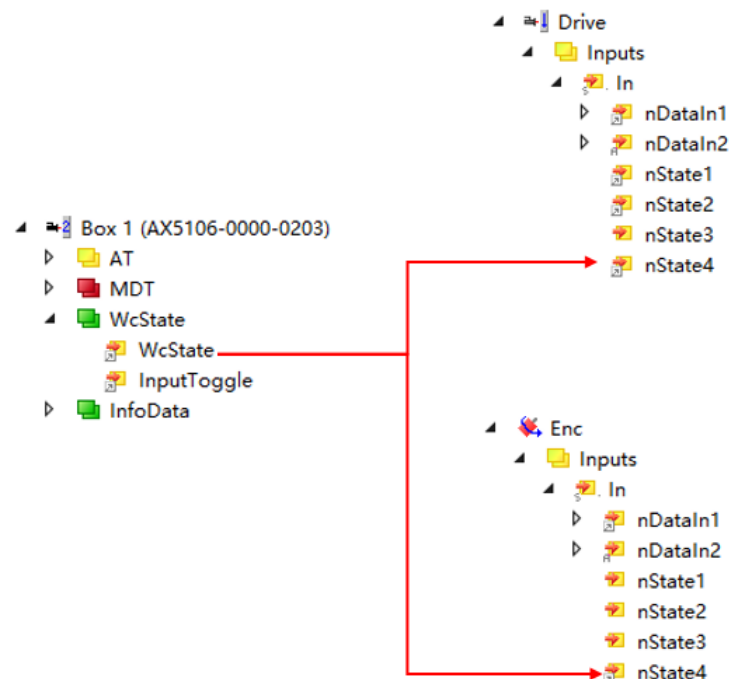
意即连续若干个 NC 周期伺服驱动器的数据无效。

(2) 故障现象

NC 报 18005 后 CoE Online 断掉显示 Offline Data: NC 报 18005 (Invalid Data) 时，NC 轴 Online 页面显示的当前位置为灰色，伺服的 CoE Online 就会断掉，只显示 Offline Data。需要 MC_Reset 后，CoE Online 才恢复正常。

(3) NC 报警的依据

对于 EtherCAT 伺服，NC 会自动通过伺服的 WcState 状态位来检测数据是否有效。



(4) WcState 与同步单元 SyncUnit 的关系

WcState 的产生机制与同步单元 (SyncUnit) 相关。根据主站的配置，每个 EtherCAT Frame 中

包括若干个 SyncUnit，其中又包含若干个 Slave。Frame 经过一个 Slave 时如果 **IO 刷新失败**，该 Slave 所属 SyncUnit 下的所有模块 WcState 都会置为 TRUE，即数据无效（Data Invalid）。

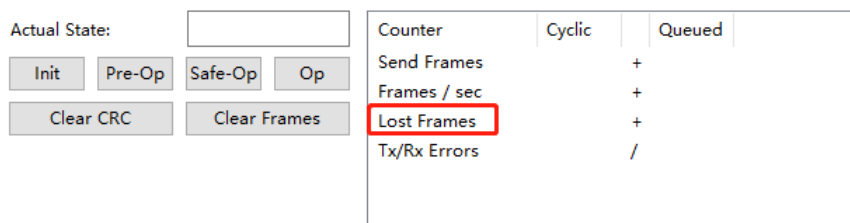
按 TwinCAT 的默认设置，一个 Frame 下的所有 Slave 都在同一个 Sync Unit，所以默认一条 EtherCAT 网络上的所有伺服都属于同一个 Sync Unit，一损俱损。

1.2 什么情况下 IO 从站会刷新失败

在 Frame 经过 Slave 时为什么 IO 刷新会失败呢？有两个可能性：要么是 Frame 没来、来晚了，或者数据破坏了，要么是 Frame 正常但 Slave 自己出了问题，不能从 Frame 提取或者插入数据。

1.2.1 什么情况下 Frame 没来？

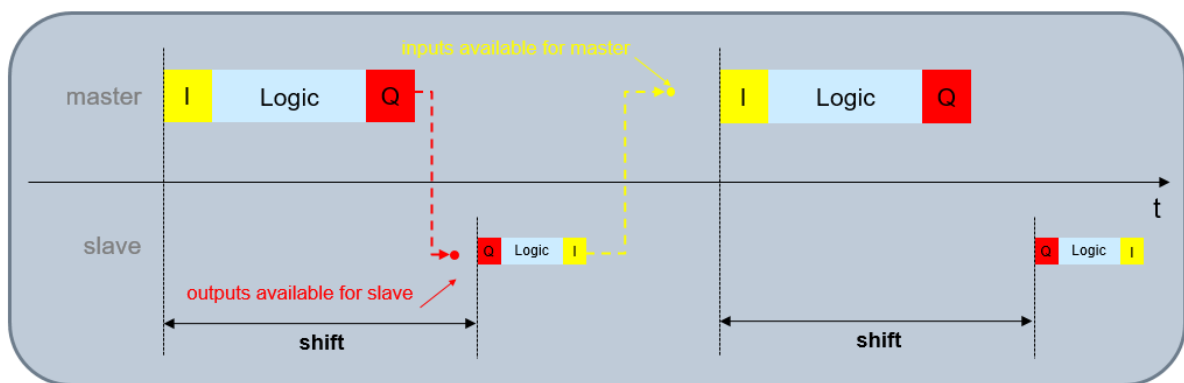
对于从站来说 Frame 没来，在主站侧来说就是 Frame Lost。这在 EtherCAT Online 界面可以监视到：



Frame Lost 的原因，并不一定是主站没有发送 Frame，只要发送了 Frame 就不会凭空丢失，更多的可能是数据破坏了，因此自破坏点之后的从站收到的都是“不合格”的 Frame。根据破坏发生的位置、时间，分类统计在 Lost Frames 或者 Tx/Rx Error 的计数器中。对从站来说，都等于是没有收到 Frame，这种错误往往是硬件引起的。

1.2.2 什么情况下 Frame 迟到？

所谓迟到，是指相对于“预定”的时间来说。只有工作在 DC Sync Mode 下的从站才需要设置 Frame 到达的“预定”时间。下图表示主站程序、EtherCAT 传输、从站程序的时间关系：



图中的 Master 就是 TwinCAT，Slave 就是伺服驱动器。以 2ms 周期的任务为例，TwinCAT 每 2ms 运算一次，然后发送数据包。伺服也是每 2ms 从数据包提取一次数据，所以必须确保 Frame 经过伺服后，伺服才提取数据。反言之，数据包必须在伺服周期开始之前到达。所以主站和从站的运算周期的点必须有所偏移，即上图中的 Shift。

TwinCAT 会自动设置这个 Shift 值，如果 Frame 不能在这个 Shift 之前到达，就叫“迟到”。**该从站在本周期就叫刷新失败**，反映在 WcState 上就是 Invalid Data。

1.2.3 Frame 按时到达，为什么 Slave 刷新不了数据？

从站应用程序定期从它的 EtherCAT 缓存区中提取 Frame 送来的数据。如果 EtherCAT 缓存中的数据正常而从站应用程序却无法处理，那一定是从站自己的问题，比如电压不够、工作不稳定，内部芯片问题、干扰问题，诸如此类。

1.3 为什么 PLC 不报 “Invalid IO data” ？

为什么总是 NC 任务报 “Invalid IO data” 而 PLC 就不会报这个错呢？

因为 NC 轴出于安全考虑，默认连续 3 个周期即 6ms 时长 WcState 都无效的话，就禁止电机继续动作。这是由于 WcState 无效时，NC 检测到的伺服状态字已经不能代表伺服的实际状态，继续动作可能有危险。根据实际的工艺要求，也可以禁止 NC 轴检测 WcState，以扩大容错能力。

另一方面，PLC 程序需要人为添加对 WcState 的检测，并编写相应的安全处理程序。因为如果 IO 模块的输入数据不能如实反应现场传感器的状态，那么 PLC 代码根据输入数据进行运算并控制输出机构的动作就可能潜藏危险。考虑到 PLC 程序的实时性要求本来就比 NC 低，连续无效触发报警的时间限制可以加长。有的伺服驱动器 WcState 旁边还有一个 InputToggle 变量，是一个握手信号，可以检测网络断线等硬件故障，在 Input Toggle 正常的前提下 WcState 的状态才有意义。实际项目中，应该两个变量配合使用，编写通讯安全逻辑。

2 伺服报警 Sync Lost (0xF415) 的触发机制

说明 1：本段文字来源于 Beckhoff Infosystem 中的 AX5000 手册。

说明 2：同样是 Sync Lost 报警，在倍福 AX5000 伺服的故障代码是 0xF415，在 OMRON 伺服的故障代码是 83，在其它品牌伺服的故障代码又是其它数字。下面的论述均以 F415 指代此类故障。

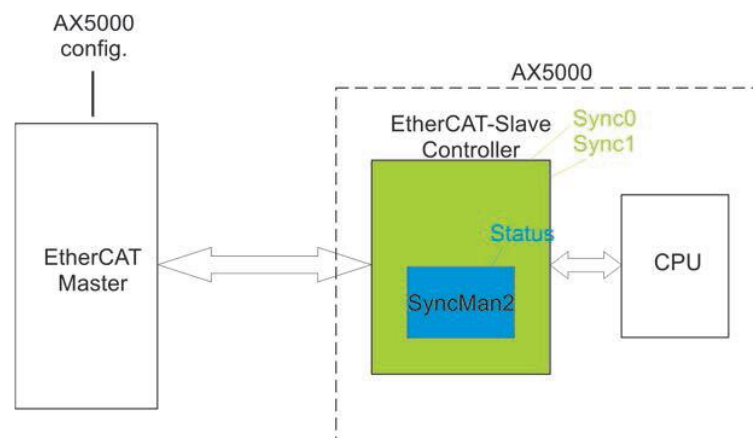
说明 3：从站报告 Sync Lost，其实就是主站的 Frame 没有按时到达或者到达时数据破坏。

2.1 EtherCAT 同步的机制

提示：此同步机制也适用于其它带分布时钟的 EtherCAT 从站，比如第三方伺服。

EtherCAT 主站发送 EtherCAT 报文至所有从站。每个从站都有一个专门的 EtherCAT 报文处理单元，称为 EtherCAT 从站控制器（ESC）。为了达到较高的定位精度和满足对同心度特性的严格要求，主机和所有连接的伺服驱动器的生成设定值应该保持同步。在 EtherCAT 系统中，这个同步任务通过分布式时钟（distributed clocks）实现。详情请参考：www.ethercat.org。

以下描述专门处理数据的同步



2.1.1 EtherCAT Master

TwinCAT 生成配置时，System Manager 根据 TwinCAT 项目和 ESI 文件，对所连接的 EtherCAT

从站自动完成其分布时钟 (DC) 参数的基本设置。

EtherCAT 网络启动时发送 Init 初始化命令将这些参数传给从站, 无须人工干预。应咨询 AX5000 技术支持才能修改默认设置。

2.1.2 EtherCAT slave controller (ESC)

AX5000 的 EtherCAT 从站控制器 (ESC) 根据主站 (在 TwinCAT) 的配置产生两个同步信号 (Sync0 和 Sync1)。AX5000 的 CPU 分析这两个同步信号, 并与内部控制算地进行同步。

(1) Sync0

Sync0 信号默认每 250us 触发一次, 如果 Sync0 触发失败 (见 F0), CPU 报 F414 错误。电机按急停减速曲线停止。

相关报警信息:

Sync0 周期只能设置为 62.5 μ s、125 μ s 或 250 μ s, 否则 CPU 报 F409 错误;

如果 ESC 中的 Sync0 信号没有激活, CPU 报 F410 错误。

如果中断脉冲宽度不达标, CPU 报 F411 错误

发生以上任何错误, 电机按急停减速曲线停止。

(2) Sync1

Sync1 信号的周期于 NC 周期决定, 它总是 Sync0 的整数倍。如果 Sync1 信号不能触发 (见 F1), CPU 也会报 F414 错误, 电机按急停减速曲线停止。

相关报警信息:

Sync1 周期必须是 Sync0 的整数倍, 并且与参数 S-0-0001 和 S-0-0002 相同, 否则 CPU 报 F412 错误。

如果 ESC 中的 Sync1 信号没有激活, CPU 报 F413 错误。

如果中断脉冲宽度不达标, CPU 报 F411 错误

发生以上任何错误, 电机按急停减速曲线停止。

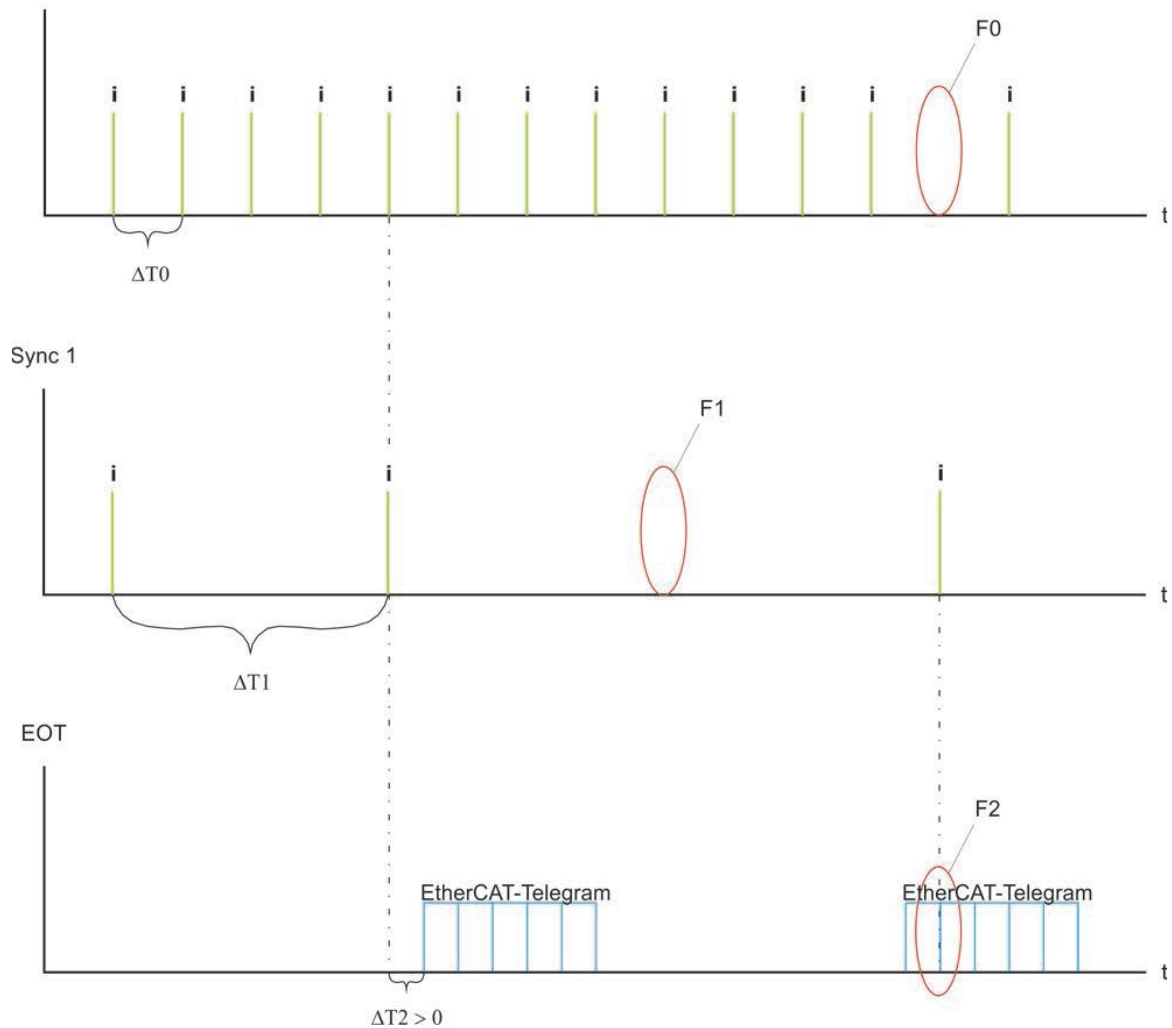
(3) End of telegram (EOT)

从站内部的 EtherCAT 状态控制器 (ESC) 动态处理 EtherCAT 报文。ESC 检测到报文末尾 (EOT) 标记后, 如果报文是发到本站的, 并且没有发生 CRC 错误, 它就将报文内容传输到指定的同步管理器 SyncManager2, 紧接着将 SyncManager2 的状态置为 “SyncManager written”。Sync 1 触发时, 如果 SyncManager2 状态为 “SyncManager written”, AX5000 的 CPU 才会从 SyncManager2 复制数据到自己的内存区。

CPU 要求 Sync1 事件触发时 SyncManager2 中的数据已经刷新, 所以必须在 Sync1 触发前检测到 EOT 标记, 否则 CPU 就不会复制数据。如果连续两次 Sync1 事件触发时都没有复制数据, AX5000 就会报 F415 错误, 电机按急停减速曲线停止。

Jitter!

由于“抖动”等原因, 必须确保在规定时间内接收到新数据。EtherCAT 主站必须确保数据及时到达 SyncMan2 (同步管理器 2)。



在上图中，EtherCAT 报文从左至后通行，最后一个 BIT 发送完毕就是 EOT 标记，左边是正常的情况：EOT 标记之后一段时间（ ΔT_2 ）才检测到 Sync1 脉冲。右边是错误的情况，Sync1 脉冲发生时，EOT 标记还没到。

2.2 故障示例

AX5000 驱动器在运行过程中持续监视设备的实时性，其中一个重要的监视目标就是数据传输链路中所有硬件和软件组件的同步性能。下面通过简单的插图举例说明这个数据传输过程，请重点关注“NC”驱动任务和“PLC”普通逻辑任务。

2.2.1 Sample 1

(1) 基本条件

1. TwinCAT 控制器的 CPU 定时器以固定时基发送中断信号 (default: base time = 1 ms)
2. 按多任务优先级管理原则依次执行各个任务
3. 任务管理：

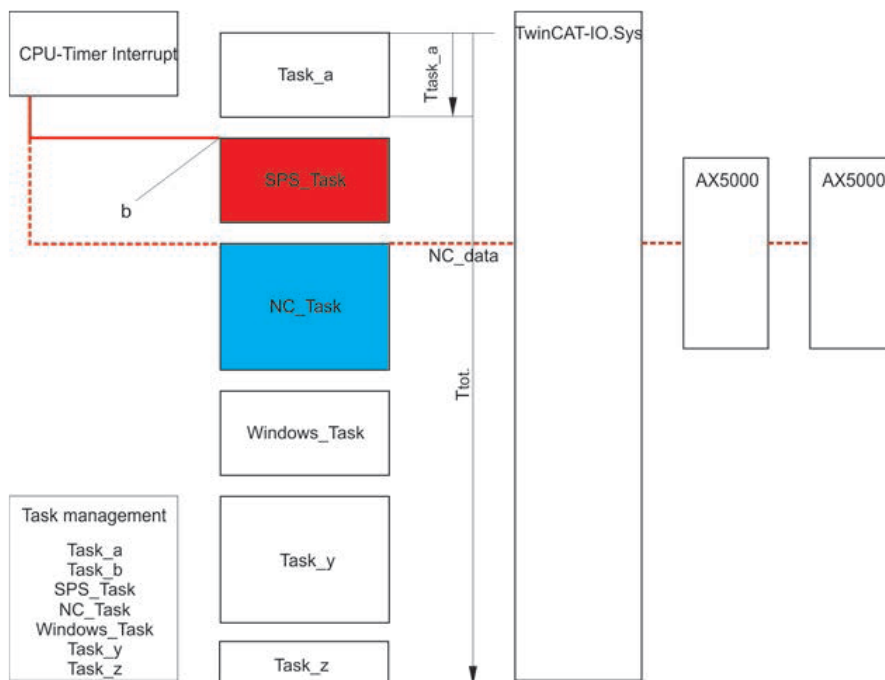
由于 PLC 计算量的增加或减少，任务占用时间相应增减，因此“I/O update”应设置为“At task beginning”，这样就可以消除一个同步错误的隐患。

不合理的任务优先级排序，则是另一个同步错误的隐患。

4. 在“I/O Update”之后，生成的数据被传输到 TwinCAT IO 系统，然后由 EtherCAT 电报发送

到连接的设备。EtherCAT 报文穿过每个物理连接的设备，并传递或接收该设备的数据。

5. 任务的优先级决定了它的执行顺序，首先执行高优先级的任务并向 TwinCAT IO 发送数据，然后 TwinCAT IO 就会发送 EtherCAT 报文。当系统里有多个不同周期的任务时，如果优先级排列不合理就容易出现问題，如图所示：



(2) 优先级设置

以下段落描述了优先级设置对数据同步的影响，

假定（PLC 优先级比 NC 高）：

Sync1 = 3 ms

NC cycle time = 3 ms

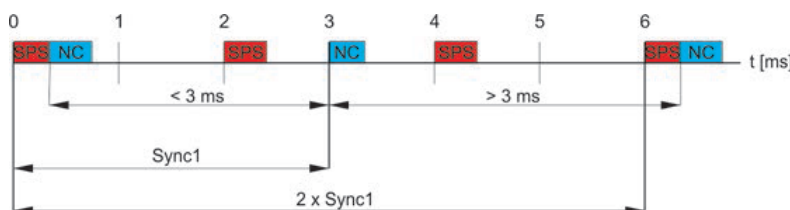
NC priority = 10

PLC cycle time = 2 ms

PLC priority = 5

NC 数据需要周期性地传输至驱动器，但在 PLC 占用计算时间时，就没有数据传输到驱动器。

由于优先级较高，PLC 任务总是在 NC 任务之前计算；这些任务在开始点时间“0 ms”相互影响，然后每“6 ms”重复一次，即 2x Sync1。驱动器的 ESC 要求 EtherCAT 报文在每个 Sync1 信号（3ms）时都送来 NC 数据。但是这一点无法保证，因为更高优先级的 PLC 任务总是先于 NC 任务计算，所以同步映射（Sync Mapping）时报文发送延迟。这样每 6 毫秒就会发生一次 NC 报文延迟，可能导致 AX5000 中的 F415 错误。



2.2.2 Sample 2

(1) 基本情况

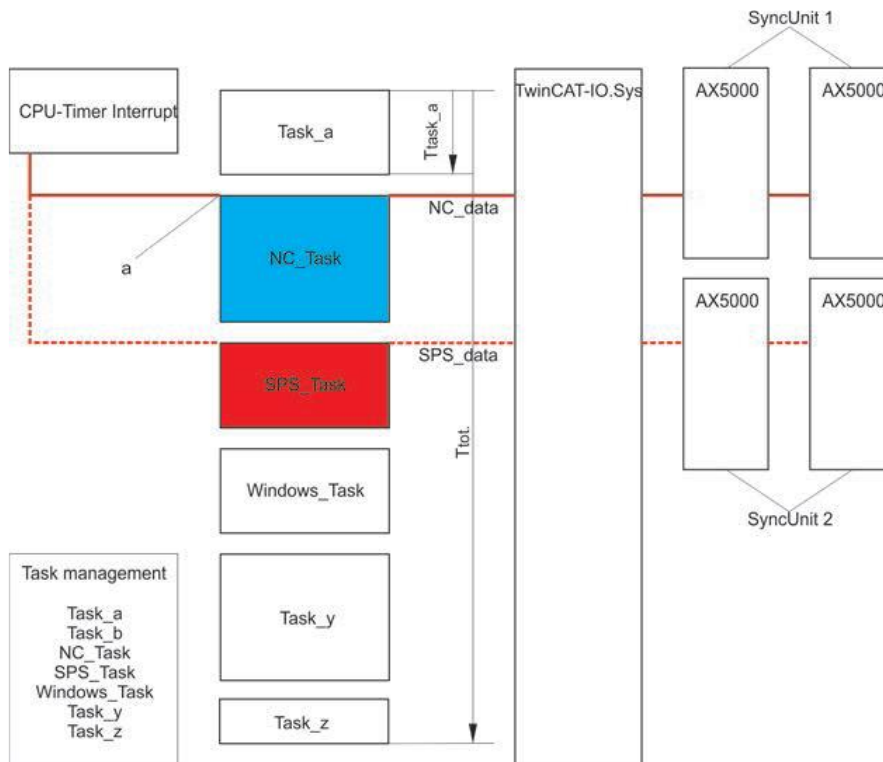
1. TwinCAT 控制器的 CPU 定时器以固定时基发送中断信号 (default: base time = 1 ms)
2. 按多任务优先级管理原则依次执行各个任务
3. 任务管理:

由于 PLC 计算量的增加或减少, 任务占用时间相应增减, 因此“**I/O update**”应设置为“**At task beginning**”, 这样就可以消除一个同步错误的隐患。

不合理的任务优先级排序, 则是另一个同步错误的隐患。

4. 在“**I/O Update**”之后, 生成的数据被传输到 TwinCAT IO 系统, 然后由 EtherCAT 电报发送到连接的设备。EtherCAT 报文穿过每个物理连接的设备, 并传递或接收该设备的数据。

5. 任务的优先级决定了它的执行顺序, 首先执行高优先级的任务并向 TwinCAT IO 发送数据, 然后 TwinCAT IO 就会发送 EtherCAT 报文。当系统里有多个不同周期的任务时, 如果优先级排列不合理就容易出现问題, 如图所示:



(2) 优先级设置

以下段落描述了优先级设置对数据同步的影响,

假定 (NC 优先级比 PLC 高, 皆为 Sync Mapping 并驱动 DC 从站):

Sync1 = 3 ms

NC cycle time = 2 ms

NC priority = 5

PLC cycle time = 3 ms

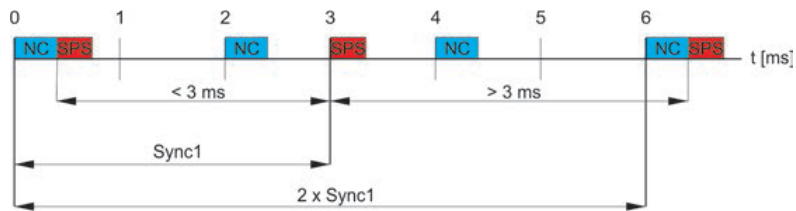
PLC priority = 25

NC 任务只处理 SyncUnit 1 中的设备，同步映射 (synchronous mapping)

PLC 任务只处理 SyncUnit 2 中的设备，同步映射 (synchronous mapping)

NC 和 PLC 数据都需要周期性地传输。

由于优先级较高，NC 任务总是先于 PLC 执行，并先发送报文；这些任务在开始点时间“0 ms”相互影响，然后每“6 ms”重复一次，即 2x Sync1。而驱动器的 ESC 要求在每次 sync1 (3ms) 时都有一个 EtherCAT 报文。这在 NC 提供服务的 SyncUnit 1 中不是问题，因为高优先级的 NC 总是以相同周期发送报文，但 PLC 报文每 6 毫秒就会出现一次延迟，可能导致 SyncUnit 2 中的 AX5000 出现 F415 错误。



3 现场排故的基本方法和顺序

由于 NC 轴的故障 0x4655 (18005) 和 AX5000 的故障 0xF415，只是同一个问题在主站侧和从站侧分别引发的报警，以下处理方法适用于这两个报警，也可以用于 PLC 检测到 WcState 为 Invalid Data 的情况。基本内容为：

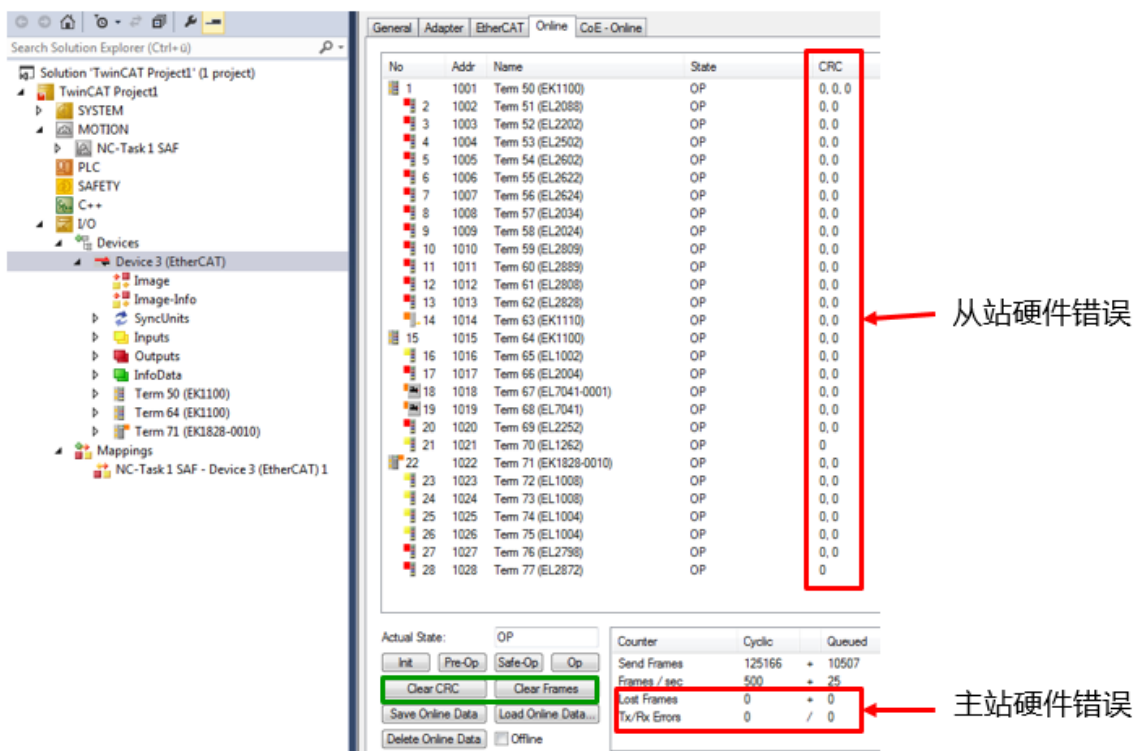
- a) 分析故障性质：软件故障还是硬件故障
- b) 硬件故障的排除方法
- c) 软件故障的排除方法

3.1 使用诊断工具分析故障性质

使用 TwinCAT 集成的 EtherCAT 诊断界面，可以大致定位故障是出于硬件原因还是软件原因。

(1) EtherCAT Online

在 EtherCAT 主站的 Online 界面可以见到主站和从站硬件故障的计数：

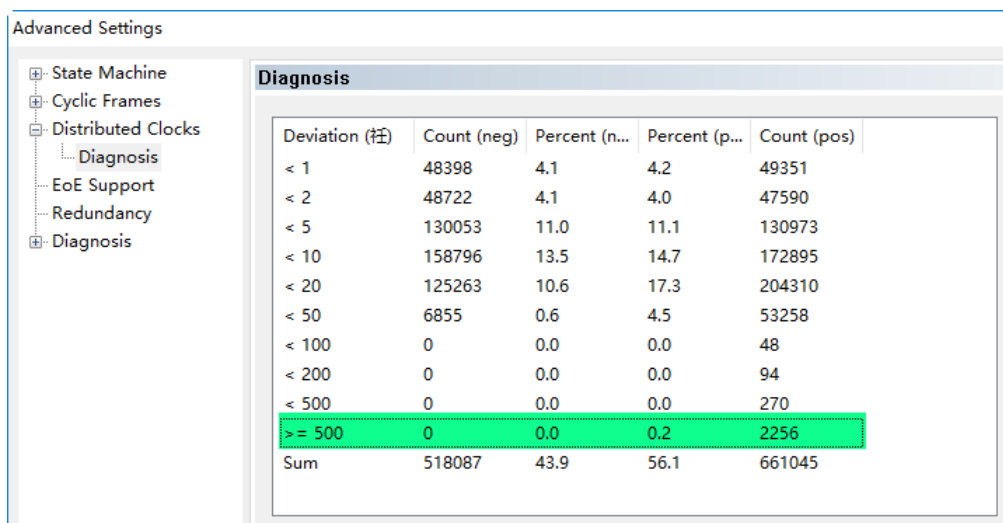


CRC 校验可以看出从站硬件错误，普通模块只有两列，左边的表示入口的 CRC 错误次数，右边表示出口的 CRC 校验错误次数。对 EL 模块来说，左边的 E-Bus 触点为入口，右边的 E-Bus 触点为出口。对 EtherCAT 驱动来说，左进右出上进下出都有可能，需要看从站的说明书。从 CRC 错误次数的发生位置，可以定位最有可能发生干扰或者故障的地方。

如果上图中红框以内的数值都是 0，说明硬件没有问题，都是软件故障。

(2) 主站 DC Diagnosis

另一个检测通讯质量的工具是主站 Advanced Setting 中的 Distributed Clocks|Diagnosis，又称为 Jitter 检测：



上图中，jitter 大于 500us 的 Frame 突然增加，可以判断是硬件故障。而小于 500 的 Frame 各个计数都稳步增加，就是 shift time 或是 PLC 等软件设置引起的。

3.2 硬件故障导致 EtherCAT Frame 出错或者丢失

严格说来，这里讲的 Frame 丢失，是指从站没有接收到“合法”的 Frame，包括 Frame Lost 和 Frame Error 两种情况。Frame 丢失通常是由硬件引起的，包括控制器、网线及接头、IO 模块及伺服驱动器等从站，以及电源、屏蔽、外部干扰源。比如 EtherCAT 接头的 4 个触点时通时断、有通有断。这种故障常常使用生产状态有关，设备开着容易报错、高速运动容易报错、高温时（夏季白天）容易报错，在最终用户现场生产了几个月甚至一两年后报错。

注意：完全拔掉网线，反而不会导致 Frame Lost，因为前一个从站可以检测到后面的网线是否拔掉，如果发现走不通就会自动原路返回，持续检测不到 Frame 的从站会切回 Init 状态，而不是报同步丢失。

3.2.1 硬件故障排查基本要求

(1) 电源

检查控制电源的电压、功率是否满足要求。电压或是功率不够时，有时会报通讯错误。

比如 AX5000 要求的控制电压允许误差+25%，考虑到开关电源的老化使用一段时间以后，实际输出电压会有所降低，以及从开关电源到驱动器如果线路较长的话，其压降也不可忽视，所以根据经验，调试之初会把开关电源的输出电压调到 26.5V。此后每次检查开关电源时，都尽可能调到此值。

4.8.5 X03 – 24 V_{DC} supply

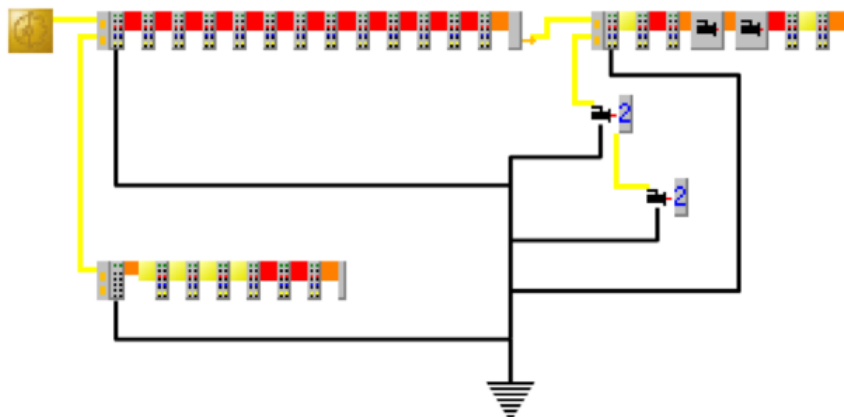


Terminal point	Connection	Current consumption
U _p +	24 V _{DC} -0/+25% - periphery (e.g. separate braking voltage)	Depending on the connected consumers (see X06 and X14, X24)
U _s +	24 V _{DC} ±25% - system supply	60 A-72 A = 3 A 90 A-170 A = 10 A
GND	GND	

(2) 屏蔽和接地

屏蔽、接地不可靠时，容易有 EMC 干扰。检查驱动器接地是否良好。

整个 EtherCAT 网络的应共用一个接地，否则不同接地之间的环流可以导致通讯数据破坏。



另外，EtherCAT 网线加磁环有助于提高抗干扰能力，第三方伺服的动力线加磁环有助于减少对外部设备的 EMC 干扰。

其它 EMC 规则……

(3) 网线和接头

网线和接头的质量

首先要使用专业的工业网线和接头，最好是倍福的预制 EtherCAT 电缆。如果客户购买倍福电缆和倍福接头自己压线时，要注意屏蔽层要和金属外壳接触牢固。

切忌使用廉价网线加水晶头自己制作 EtherCAT 通讯电缆，如果在排故现场发现这种情况，要特别检查网线是否带屏蔽，屏蔽层是否和接头接触良好。

如果条件允许，使用倍福预制网线挨个替换有疑点的网线。

安装牢固

检查网线接头：是否有氧化现象。。

检查驱动器网口：是否损坏，接触弹片不够力，变形或是氧化的情况。

连接网线后，网口最好不要受力，否则网口容易变形。

晃动网线，检查插接是否松动，是否出现报警，更换网线等。

3.2.2 软件基本配置检查

(1) Ebus 电流是否足够

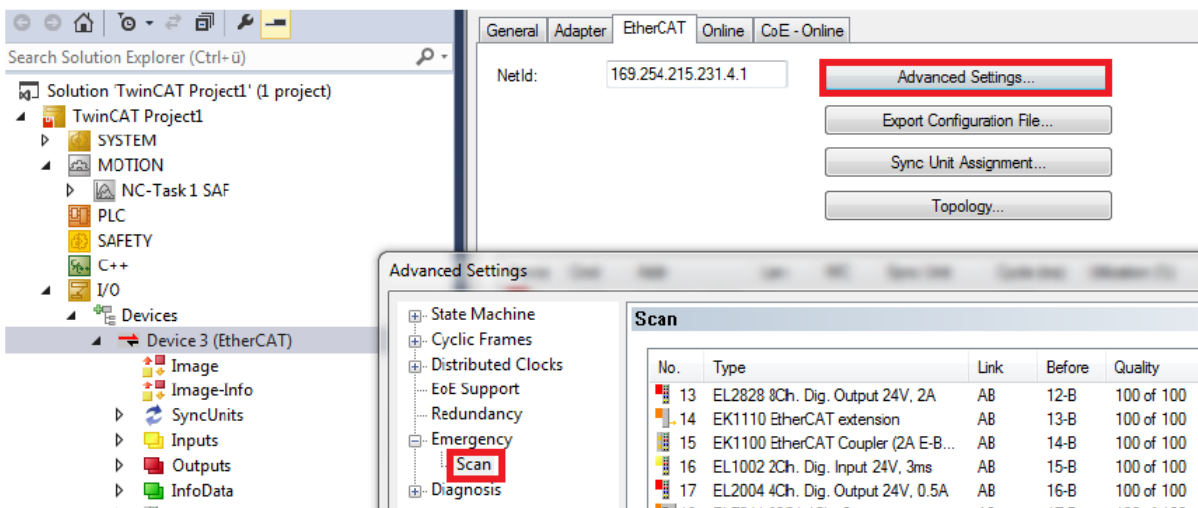
检查 IO 模块的 EBus 电流是否足够，如果模块和驱动都再同一个 EtherCAT 网络里，某些 IO 出问题导致 EtherCAT 数据不能返回主站，所有驱动都会报同步丢失。

(2) 配置文件与实际安装的硬件相符

最简单的方法是重新扫描 IO，与项目文件中配置对比是否一致。

其实 TwinCAT 还集了一个工具“主站的 Emergency Scan（紧急扫描）”。使用 Emergency Scan 可以验证离线配置和在线配置是否一致。方法如下：

在 TwinCAT Config Mode 时，执行 Emergency Scan 可以发送预定数量的探测数据帧，用于快速测试物理连接的永久硬件问题（设备、电缆或者接头损坏），但是这个方法很难检测到随机干扰。



如果实际 IO 模块和从站与配置一致，则上图中最后一列“Quality”总是 100 of 100，表示发送 100 次，成功返回 100 次。如果为 0 of 100，当然表示配置的模块或者从站不存在。如果小于 100，表示部分数据包没有成功返回，这个从站多半有问题。

3.2.3 软件配置的基本优化

(1) 每个伺服驱动器设置独立同步单元

这是为了排除个别驱动器自己的原因出错时影响其它驱动器也报 Data Invalid，而 EtherCAT 对这些报 Data Invalid 的从站一视同仁，伤及无辜。

(2) 清除伺服的 WcState 与 NC 轴的变量映射

默认 NC 检测到连接 3 个 SAF 周期 WcState 为 TRUE，就触发 18005 报警。而有的第三方从站，硬件设置的 Sync Lost 容错次数可能比 NC 还多，结果伺服还没报警 NC 就报了。从而 NC 不再发送新数据（此时 NC 轴的 Online 页面看到的当前位置为灰色，伺服的 CoE Online 页面也只显示 Offline 数据），引起伺服报警，需要用 MC_Reset 才能恢复。

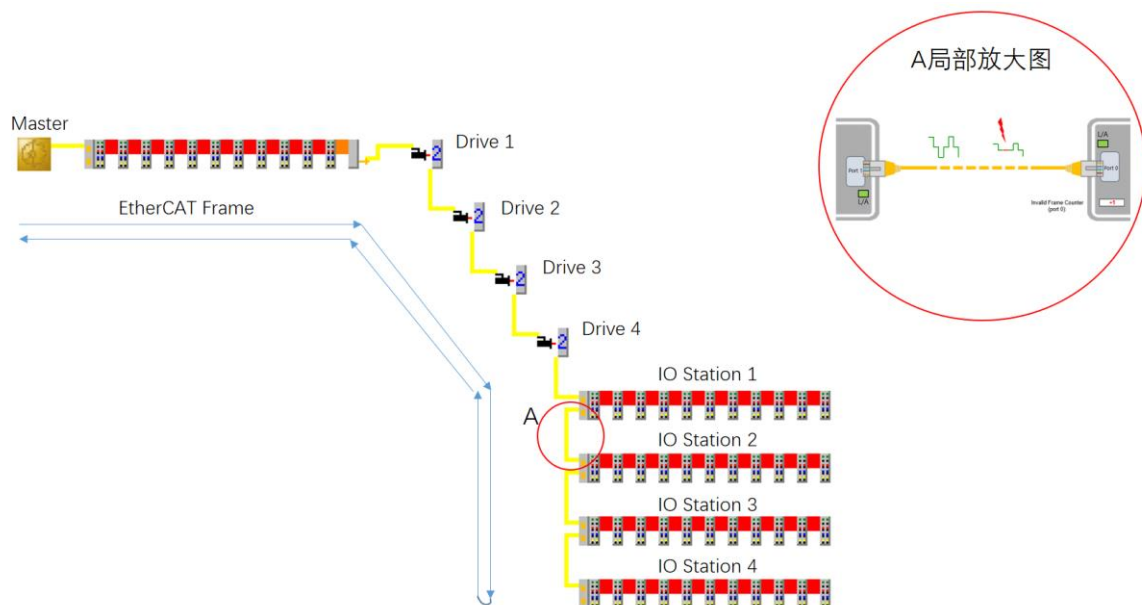
Sync Unit 的局限性

如果是 Frame 漏发、延迟或者数据破坏，无论 Sync Unit 怎样设置，所有从站或者后半部分从站都会 IO 刷新失败。如果 Frame 正常而是 Slave 自己出了问题，那么把每个伺服设置独立同步单元（最好再配合星形拓普和 HotConnect 设置）就有助于精确定位到故障从站。

3.2.4 EtherCAT 链路优化

(1) 链路分布对抗干扰的影响。

国内用户使用嵌入式控制器 CX 较多，通常接线时 IO 模块接在 CX 后面，末尾通过 EK1110 再出一条网线连接到伺服和远程 IO。根据 IO 站和伺服的数量多少和安装位置，EtherCAT 拓普结构大致如下：



例如上图中有 40 个 IO 模块，分属于控制器和 4 个远程 IO 站。4 个驱动器和控制器一起安装在主电箱。PLC 任务每 10ms 发送 Frame1 刷新 50 个 IO 模块的数据，而 NC 任务每 2ms 发送 Frame2 刷新 4 个驱动器的数据。在这个 EtherCAT 网络中，每个 Frame 都要经过上图蓝色箭头所示路径，“环游”所有从站。对于传输 Drive 数据的 Frame2，它第 1 次经过 Drive4 后所有 IO 数据都已交互完成，但它还必须经过后面的 4 个 IO 站上 40 个模块。图中黄色线条标记的就是 EtherCAT 网线，从局部大放图可以看到，每一段网线都可能受到干扰。**网线越长、网线段数越多，受到干扰的可能性越大。**已经完成数据交互的 Frame2 在经过这些 IO 站的时候如果受到干扰引起数据破坏，回到主站时就会被判断为无效。

普通的 IO 模块或者第三方 EtherCAT 从站偶尔数据无效是可以接受的，从站不会报错，PLC 程序也不必报错。但是对于伺服驱动器来说，连续几个周期数据无效，伺服会报警，NC 也会报警。

(2) 解决办法：IO 和驱动分开两条 EtherCAT 网络

从上面的分析可知，要解决伺服的通讯受干扰问题，除了做好各种 EMC 措施之外，让伺服（以及需要 DC 时钟的从站）单独走一条 EtherCAT 网络，可以最大限度减少不良影响。

建议在电气设计之初，如果驱动器比较多，而且成本预算和安装空间都允许，推荐让伺服驱动和普通 IO 分开，各走一条 EtherCAT 网络。

CX50、CX51、CX20 系列 EPC 除了右侧的内置 EtherCAT 之外左侧都有两个独立网卡，其中一个就可以用来走 EtherCAT 带伺服驱动。

对于 C60、C69 等 IPC，标配虽有两个独立网口，但是一条用于编程和外部 Ethernet 通讯，一条用于 EtherCAT，如果 IO 和驱动要分开两条 EtherCAT 网络，就需要扩展一个网卡或者增加 CU2508 网络倍增器。CU2508 需要占用一个 1000M 网卡，支持 CE 和 Win7、Win10 等。

3.2.5 其它说明

(1) 主站网卡的问题

控制器网卡或是驱动问题，如果是第三方工控机或是安装了较多其他驱动的控制器的，也容易出现问題，不稳定。

(2) 重点检查第 1 个报 Sync Lost 的从站

(3) 伺服靠前放置或者星形连接的作用

对于软件引起的 Frame 迟到问题，将驱动器靠前放置或者使用 EK1122 或 CU1128 做星形连接可以改善。但是对于 EMC 引起的 Frame 丢失或者损坏，改变伺服接线顺序或者星形连接就收效甚微。

3.3 软件原因导致的 EtherCAT Frame 迟到

Frame 迟到，通常是由软件引起的。

一是 EtherCAT 主站和从站的 Time Shift 不够；

二是任务优先级不合理；

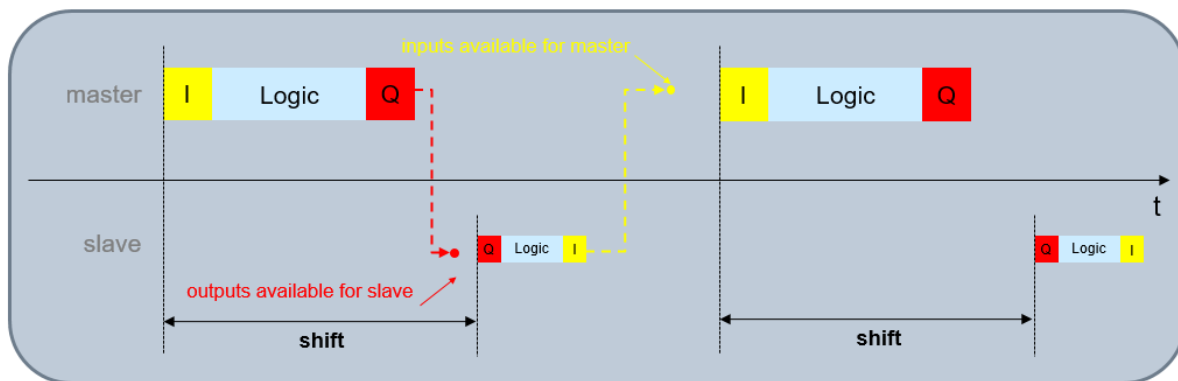
三是程序计算量波动。

一个不常见的软件错误是 DC 主时钟紊乱。

软件引起的 Frame 迟到通常不会伴随丢包、TxRxError 等累加，并且与设备的生产状态无关，并且越是链路上靠后的从站越容易报错。通常在厂内调试阶段就会发现，而不会在最终用户现场生产了一段时间才报错。

3.3.1 EtherCAT 主站和从站的 Time Shift 优化

所谓 Frame “迟到”，是指相对于“预定”的时间来说。只有工作在 DC Sync Mode 下的从站才需要设置 Frame 到达的“预定”时间。下图表示主站程序、EtherCAT 传输、从站程序的时间关系：

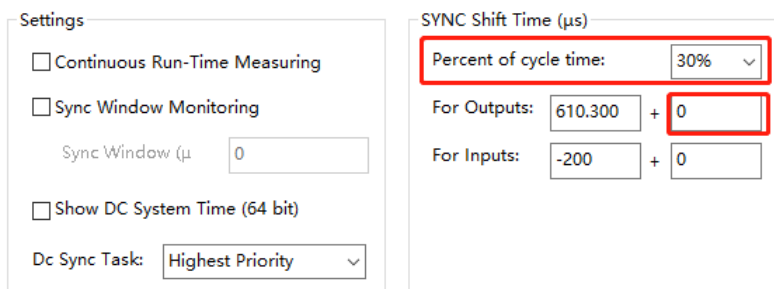


图中的 Master 就是 TwinCAT，Slave 就是伺服驱动器。以 2ms 周期的任务为例，TwinCAT 每 2ms 运算一次，然后发送数据包。伺服也是每 2ms 从数据包提取一次数据，所以必须确保 Frame 经过伺服后，伺服才提取数据。反言之，数据包必须在伺服周期开始之前到达。所以主站和从站的运算周期的点必须有所偏移，即上图中的 Shift。

对于特定的从站来说，这个 Shift 由两部分组成：一是主站设置的**适用于所有 DC 从站**的偏移，二是**个别从站**单独设置的“额外”偏移量。**二者迭加**就是该从站的应用程序起始点相对于主站的偏移时间。在这个时间内，Frame 必须出发并到达该 DC 从站，否则该从站就会报告 Sync Lost。

EtherCAT 主站的 SYNC Shift Time 设置

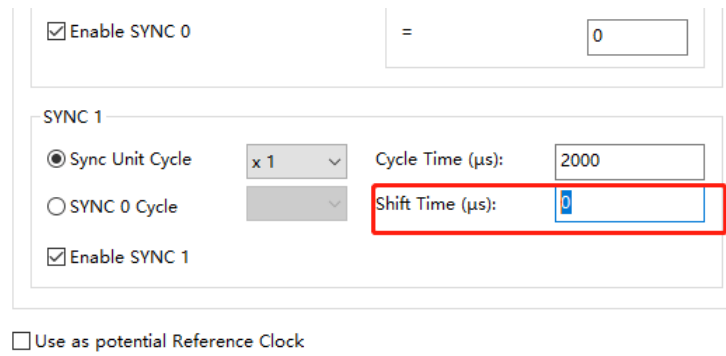
EtherCAT 主站 Advanced Setting 的 Distributed Clock 界面的右下方“SYNC Shift Time”参数：



上图中的 30%，就是表示所有 DC 从站的应用周期，必须在主站周期开始后 600us 才开始（以任务周期 2ms 为例）。由于 NC 任务默认是先刷新 IO 后进行运算，所以 NC 任务一开始就会发送 EtherCAT Frame，只要 600us 内 Frame 可以到达最后一个 DC 从站，Frame 就不算迟到。

从站 Shift Time 的设置

从站的 DC 设置中，可以设置自己的“额外”偏移，与主站的 Shift Time 迭加，就是它相对于主站任务起始点的实际偏移时间。



3.3.2 优先级不合理的解决办法

如前所述，系统中存在多个带 DC 从站的任务时，Frame 会以多个 DC 任务的最小公倍数周期性“延迟”，如果任务周期短，从站数量多，优先级低的任务发出的 Frame 就更容易迟到，因而与之同步映射的 DC 从站就容易报 Sync Lost。

优先级不合理导致的 Sync Lost，从第一次联机调试就会出现，容易发现，也容易解决。方法是：

(1) 确保周期短的任务优先级高

确保周期短的任务优先级高，勾选“Auto Priority”可以实现

(2) 多任务驱动从站的优化原则

对于不同周期的任务都要驱动 DC 从站的情况，延迟几乎无可避免，为此可以做 3 点优化：

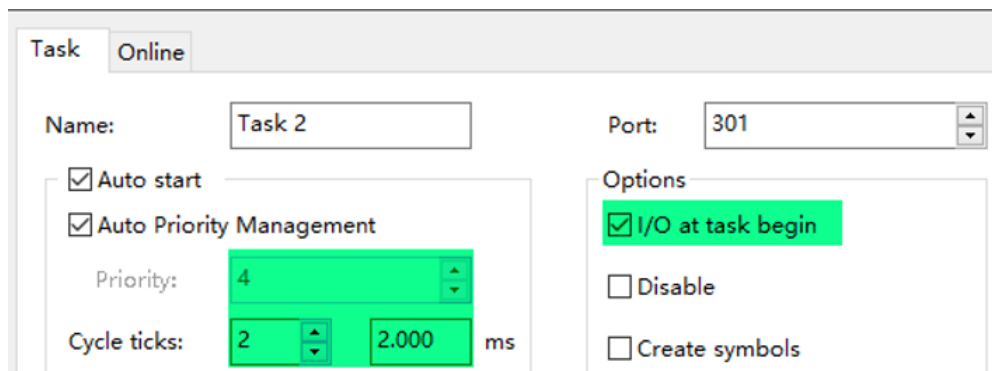
- a) 只保留一个同步映射，
- b) 或者把需要 DC 同步的从站都放到一个任务。
- c) 或者把低优先级任务的从站单个设置 Shift Time，保留足够的时间余量，以备它排在最后执行时其 Frame 仍能赶在从站要求的时间之前到达。

3.3.3 程序计算量波动的解决办法

对于 PLC 程序，每个周期的计算量波动是正常的。克服计算量波动的影响，要从两个方面进行：

(1) 启用 I/O at task beginning

TC2 中的界面如下：



在 TwinCAT 3 中，PLC 任务和 C++ Modules 都不提供“IO at task beginning”的标记。此时可以使用属性 `TcCallAfterOutputUpdate` 实现同样的功能：

Attribute TcCallAfterOutputUpdate

The pragma {attribute 'TcCallAfterOutputUpdate'} defines whether a program is to be executed after an output update. This attribute replaces the TwinCAT 2 functionality of the option "IO at Task begin".

Syntax:

```
{attribute 'TcCallAfterOutputUpdate' }
```

This attribute must be added to all program POU's, which are to be called after the output update.

Example:

```
{attribute 'TcCallAfterOutputUpdate'}
PROGRAM MAIN
VAR
END_VAR
```

(2) 确认 PLC 程序运算没有持续超时

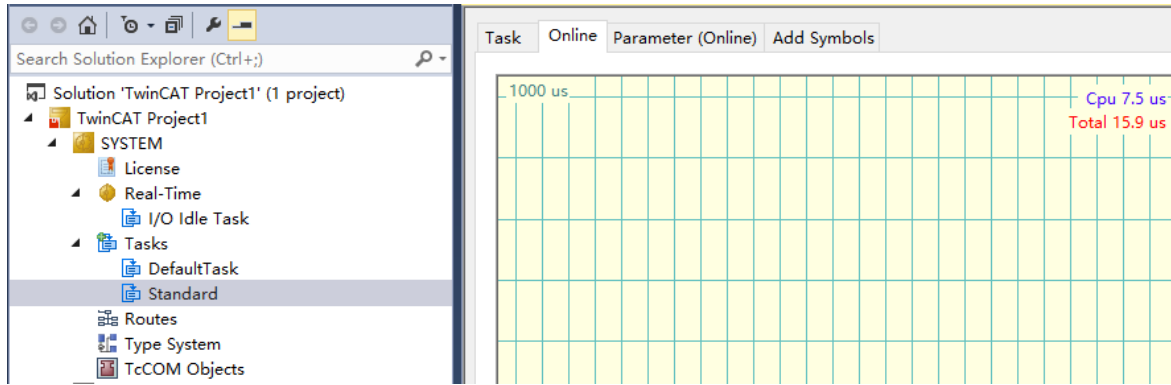
从 TwinCAT PLC 提供系统变量 lastExecTime 和 cycleTimeExceeded, 可以观察 PLC 任务执行的波动情况。在 TC2 和 TC3 中查看这两个变量的方法如下:

TwinCAT 2 中的任务执行信息	TwinCAT 3 中的任务执行信息
<pre>SystemInfo (%MB32768) ├──.runTimeNo = 1 ├──.projectName = '编程语言' ├──.numberOfTasks = 1 ├──.onlineChangeCount = 0 ├──.bootDataFlags = 16 ├──.systemStateFlags = 32 └──SystemTaskInfoArr (%MB32832) ├──SystemTaskInfoArr[1] │ ├──.active = TRUE │ ├──.taskName = 'Standard' │ ├──.firstCycle = FALSE │ ├──.cycleTimeExceeded = FALSE │ ├──.cycleTime = 100000 │ ├──.lastExecTime = 115 │ ├──.priority = 25 │ └──.cycleCount = 1729 ├──SystemTaskInfoArr[2] ├──SystemTaskInfoArr[3] └──SystemTaskInfoArr[4]</pre>	<pre>Y:=GETCURTASKINDEXEX(); 获取任务编号 TwinCAT_SystemInfoVarList._TaskInfo[Y].CycleTime; TwinCAT_SystemInfoVarList._AppInfo.;</pre> <p>隐含全局变量 TwinCAT_SystemInfoVarList 的子元素</p> <ul style="list-style-type: none"> AdsPort AppName AppTimestamp BootDataLoaded Flags KeepOutputsOnBP LicensesPending ObjId OldBootData OnlineChangeCnt ProjectName ShutdownInProgress TaskCnt TComSrvPtr <pre>Y:=GETCURTASKINDEXEX(); TwinCAT_SystemInfoVarList._TaskInfo[Y].; TwinCAT_SystemInfoVarList._AppInfo.Task</pre> <ul style="list-style-type: none"> AdsPort CycleCount CycleTime CycleTimeExceeded DcTaskTime FirstCycle LastExecTime ObjId Priority TaskName

lastExecTime 波动的影响

对于 PLC 任务, 默认是先运算后发送 Frame, 如果有的周期运算量特别大, 可能发送 Frame 的时间就比平时晚, 以致 Frame 不能在设置的 Shift Time 之前到达每个 DC 从站。所以, 对于有 DC 要求的 Frame, 在 PLC 任务中设置 “IO at task beginning” 是很重要的, 以此避免 PLC 计算量波动引起的 Frame 延迟。

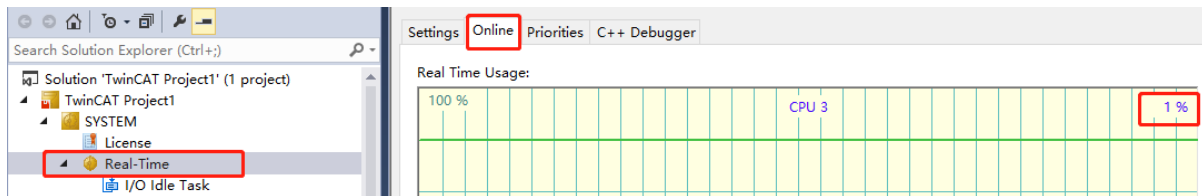
除了进 PLC 程序监视 lastExecTime, 从 Task 的 Online 里也能大致看出执行时间的波动情况:



cycleTimeExceeded 的影响

当 cycleTimeExceeded 置位 1 次, Frame 就会少发 1 次, DC 从站就有 1 个周期没有收到 Frame。大多数普通模块偶尔少一个 Frame 是可以接受的, 但对于某些敏感的 DC 从站, 或者刚好在这个周期需要输出的 XFC 通道, 就会有不良后果。

通常任务超时 (cycleTimeExceeded) 发生时, CPU 利用率会达甚至超过 Realtime 设置的限值, 默认 80%。所以直接观察 CPU 利用率, 也可以估算是否出现任务超时:



如果检测到持续任务超时, 就要查找原因, 优化程序。

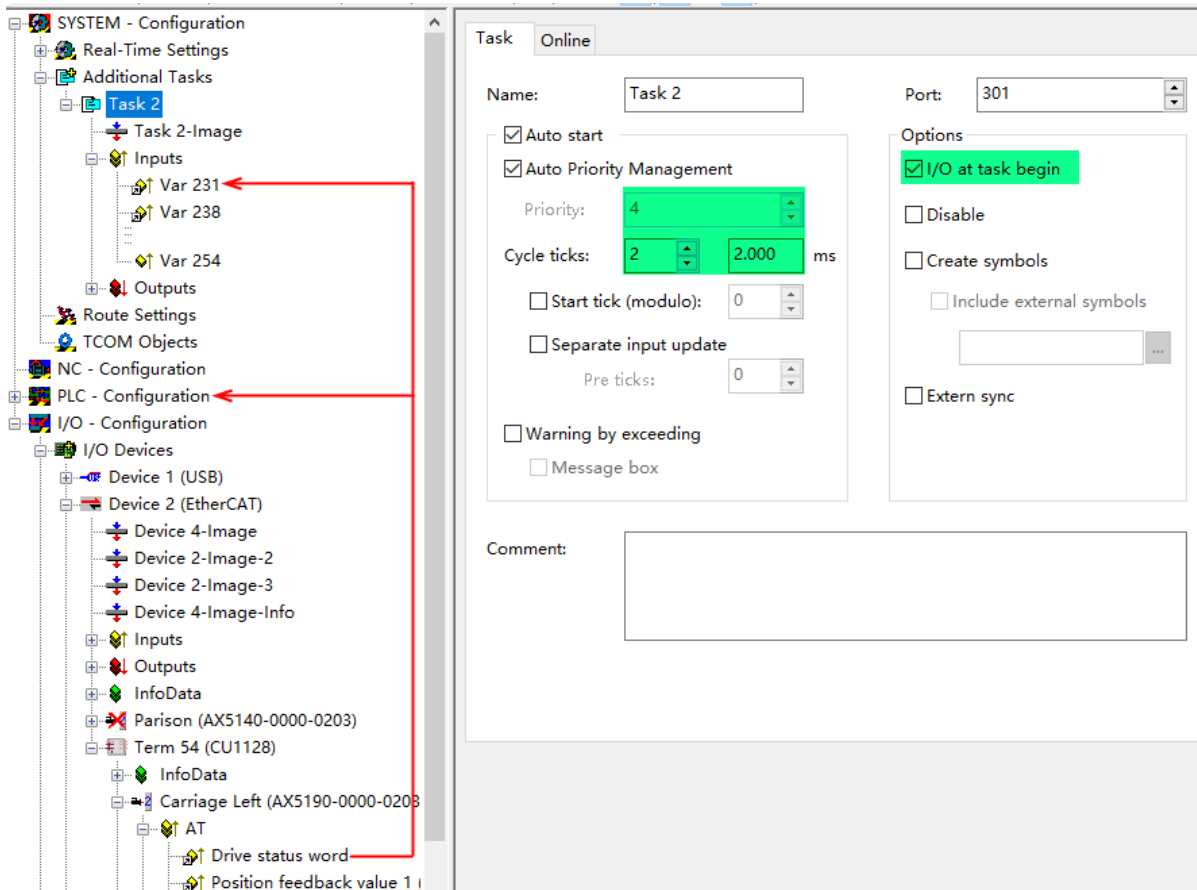
(3) 一个测试手段: 新建 Additional Task 刷新 IO

为了排查 PLC 运算量波动引起的 Frame 迟到或者漏发, 可以新建一个高优先级、Auto Start 的 Additional Task 来刷新 IO, 并把需要链接到 DC 从站的变量置于这个任务下。这样就可以完全排除 PLC 运算量波动的影响。

提示 1: 如果验证并非 PLC 运算量波动引起 Frame 迟到或者丢失, 最好恢复原样: 删除临时添加的 Task, 恢复变量链接到 PLC 任务。

提示 2: 如果不是对 Task、Mapping 和 EtherCAT 通讯机制深入了解的用户, 请应尽量使用“常规”“默认”的配置。如果要修改, 应充分了解这样做的理由, 和可能产生的关联结果。

Prio...	Cycle	Task	Comment
1			
2			
3			
4	2.0	Task 2	
5			
6		'Boost Priority'	PLC Run-Time 1 'AE_970-...
7	2.0	FastTask (Task 0)	PLC Run-Time 1 'AE_970-...
8			
9	1.0	I/O Idle Task	
10			
11	5.0	MiddleTask (Task 1)	PLC Run-Time 1 'AE_970-...
12			
13	25.0	SlowTask (Task 2)	PLC Run-Time 1 'AE_970-...
14			



3.3.4 DC 主时钟（Reference Clock）紊乱

如果客户临时修改了硬件配置，比如禁用了作为参考时钟的 EtherCAT 从站，或者控制器正常运行中单独重启作为参考时钟的 EtherCAT 从站，就会引发 DC 时钟紊乱，以至于判断 Frame 是否准点或者迟到完全失去了依据。

检查到是禁用了主时钟，则另外指定主时钟更新配置，重新激活。

检查到是主时钟断电重启，则重启 EtherCAT 主站（掉电重启，或者切换状态 OP-Init-OP）