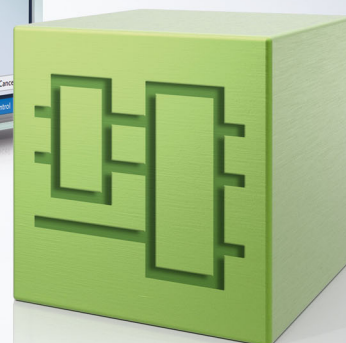
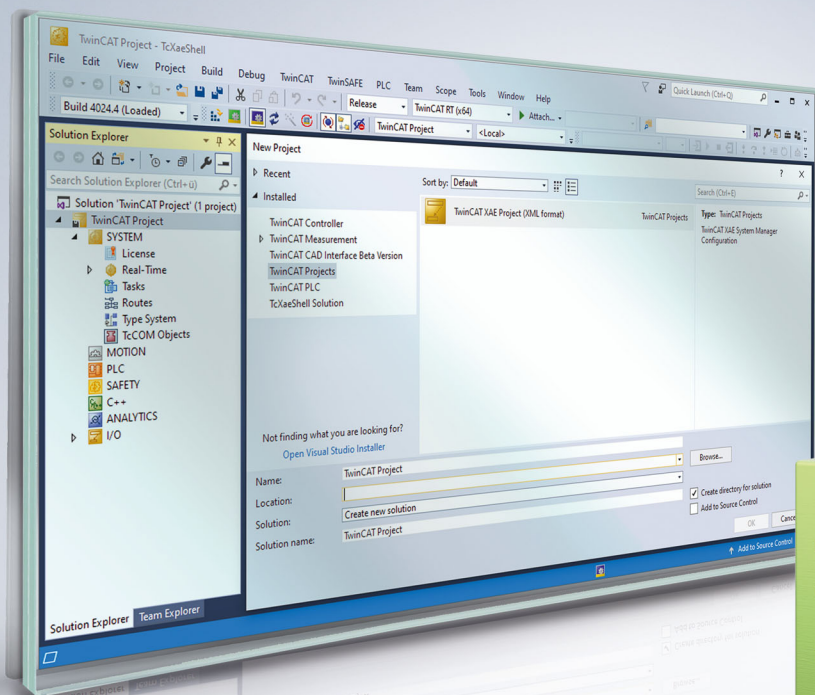


手册 | ZH

# TwinCAT 3

TwinCAT 3 | PLC Lib: Tc2\_EtherCAT





# 目录

|                                                  |           |
|--------------------------------------------------|-----------|
| <b>1 前言 .....</b>                                | <b>7</b>  |
| 1.1 文档说明 .....                                   | 7         |
| 1.2 安全信息 .....                                   | 8         |
| 1.3 信息安全说明 .....                                 | 8         |
| <b>2 概述 .....</b>                                | <b>10</b> |
| <b>3 EtherCAT 命令 .....</b>                       | <b>11</b> |
| 3.1 FB_EcPhysicalReadCmd .....                   | 11        |
| 3.2 FB_EcPhysicalWriteCmd .....                  | 12        |
| 3.3 FB_EcLogicalReadCmd .....                    | 14        |
| 3.4 FB_EcLogicalWriteCmd .....                   | 15        |
| <b>4 EtherCAT Diagnostic .....</b>               | <b>17</b> |
| 4.1 FB_EcGetAllMasters .....                     | 17        |
| 4.2 FB_EcGetAllSlaveAbnormalStateChanges .....   | 18        |
| 4.3 FB_EcGetAllSlaveAddr .....                   | 19        |
| 4.4 FB_EcGetAllSlaveCrcErrors .....              | 20        |
| 4.5 FB_EcGetAllSlavePresentStateChanges .....    | 21        |
| 4.6 FB_EcGetConfSlaves .....                     | 22        |
| 4.7 FB_EcGetLastProtErrInfo .....                | 23        |
| 4.8 FB_EcGetMasterDevState .....                 | 24        |
| 4.9 FB_EcGetScannedSlaves .....                  | 25        |
| 4.10 FB_EcGetSlaveCount .....                    | 26        |
| 4.11 FB_EcGetSlaveCrcError .....                 | 27        |
| 4.12 FB_EcGetSlaveCrcErrorEx .....               | 28        |
| 4.13 FB_EcGetSlaveIdentity .....                 | 29        |
| 4.14 FB_EcGetSlaveTopologyInfo .....             | 30        |
| 4.15 FB_EcMasterFrameCount .....                 | 31        |
| 4.16 FB_EcMasterFrameStatistic .....             | 32        |
| 4.17 FB_EcMasterFrameStatisticClearCRC .....     | 33        |
| 4.18 FB_EcMasterFrameStatisticClearFrames .....  | 34        |
| 4.19 FB_EcMasterFrameStatisticClearTxRxErr ..... | 34        |
| 4.20 F_CheckVendorId .....                       | 35        |
| 4.21 F_EcGetLinkedTaskOfSyncUnit .....           | 36        |
| 4.22 F_EcGetSyncUnitName .....                   | 36        |
| <b>5 EtherCAT 状态机 .....</b>                      | <b>38</b> |
| 5.1 FB_EcGetAllSlaveStates .....                 | 38        |
| 5.2 FB_EcGetMasterState .....                    | 39        |
| 5.3 FB_EcGetSlaveState .....                     | 40        |
| 5.4 FB_EcReqMasterState .....                    | 41        |
| 5.5 FB_EcReqSlaveState .....                     | 42        |
| 5.6 FB_EcSetMasterState .....                    | 44        |
| 5.7 FB_EcSetSlaveState .....                     | 45        |

|                                            |           |
|--------------------------------------------|-----------|
| <b>6 CoE 接口 .....</b>                      | <b>47</b> |
| 6.1 FB_EcCoeSdoRead .....                  | 47        |
| 6.2 FB_EcCoeSdoReadEx.....                 | 48        |
| 6.3 FB_EcCoeSdoWrite .....                 | 49        |
| 6.4 FB_EcCoeSdoWriteEx .....               | 51        |
| 6.5 FB_CoERead_ByDriveRef .....            | 52        |
| 6.6 FB_CoEWrite_ByDriveRef.....            | 53        |
| 6.7 FB_EcCoeReadBIC .....                  | 55        |
| 6.8 FB_EcCoeReadBTN .....                  | 56        |
| <b>7 FoE 接口.....</b>                       | <b>58</b> |
| 7.1 FB_EcFoeAccess.....                    | 58        |
| 7.2 FB_EcFoeClose.....                     | 59        |
| 7.3 FB_EcFoeLoad .....                     | 59        |
| 7.4 FB_EcFoeOpen.....                      | 61        |
| 7.5 FB_EcFoeReadFile .....                 | 62        |
| 7.6 FB_EcFoeWriteFile .....                | 64        |
| <b>8 SoE 接口 .....</b>                      | <b>66</b> |
| 8.1 FB_EcSoeRead .....                     | 66        |
| 8.2 FB_EcSoeWrite.....                     | 67        |
| 8.3 FB_SoERead_ByDriveRef.....             | 69        |
| 8.4 FB_SoEWrite_ByDriveRef .....           | 71        |
| <b>9 转换功能.....</b>                         | <b>73</b> |
| 9.1 F_ConvBK1120CouplerStateToString ..... | 73        |
| 9.2 F_ConvMasterDevStateToString.....      | 73        |
| 9.3 F_ConvProductCodeToString .....        | 74        |
| 9.4 F_ConvSlaveStateToString .....         | 74        |
| 9.5 F_ConvSlaveStateToBits .....           | 75        |
| 9.6 F_ConvSlaveStateToBitsEx .....         | 75        |
| 9.7 F_ConvStateToString .....              | 76        |
| <b>10 分布式时钟.....</b>                       | <b>77</b> |
| 10.1 DCTIME32 .....                        | 77        |
| 10.1.1 ConvertDcTimeToPos.....             | 77        |
| 10.1.2 ConvertPosToDcTime.....             | 78        |
| 10.1.3 ConvertDcTimeToPathPos.....         | 79        |
| 10.1.4 ConvertPathPosToDcTime.....         | 80        |
| 10.2 DCTIME64 .....                        | 81        |
| 10.2.1 DCTIME_TO_DCTIME64 .....            | 81        |
| 10.2.2 DCTIME64_TO_DCTIME .....            | 81        |
| 10.2.3 DCTIME64_TO_DCTIMESTRUCT .....      | 81        |
| 10.2.4 DCTIME64_TO_FILETIME64 .....        | 82        |
| 10.2.5 DCTIME64_TO_STRING.....             | 83        |
| 10.2.6 DCTIME64_TO_SYSTEMTIME.....         | 83        |
| 10.2.7 DCTIMESTRUCT_TO_DCTIME64.....       | 84        |

|         |                                      |     |
|---------|--------------------------------------|-----|
| 10.2.8  | FILETIME64_TO_DCTIME64 .....         | 84  |
| 10.2.9  | STRING_TO_DCTIME64.....              | 85  |
| 10.2.10 | SYSTEMTIME_TO_DCTIME64.....          | 86  |
| 10.2.11 | FB_EcDcTimeCtrl64 .....              | 86  |
| 10.3    | DCTIME64 和 ULINT .....               | 88  |
| 10.3.1  | F_ConvExtTimeToDcTime64.....         | 88  |
| 10.3.2  | F_ConvTcTimeToDcTime64 .....         | 88  |
| 10.3.3  | F_ConvTcTimeToExtTime64 .....        | 89  |
| 10.3.4  | F_GetActualDcTime64 .....            | 89  |
| 10.3.5  | F_GetCurDcTaskTime64.....            | 90  |
| 10.3.6  | F_GetCurDcTickTime64.....            | 90  |
| 10.3.7  | F_GetCurExtTime64.....               | 91  |
| 10.3.8  | FB_EcExtSyncCalcTimeDiff64 .....     | 91  |
| 10.3.9  | FB_EcExtSyncCheck64 .....            | 92  |
| 10.4    | [废弃].....                            | 93  |
| 10.4.1  | [过时的 DCTIME] .....                   | 93  |
| 10.4.2  | [过期的 DCTIME 和 T_LARGE_INTEGER] ..... | 100 |
| 10.4.3  | DCTIME64_TO_FILETIME .....           | 106 |
| 10.4.4  | FILETIME_TO_DCTIME64 .....           | 106 |
| 11      | [废弃].....                            | 108 |
| 11.1    | F_GetVersionTcEtherCAT .....         | 108 |
| 12      | 数据类型.....                            | 109 |
| 12.1    | E_EcAdressingType.....               | 109 |
| 12.2    | E_EcFoeMode.....                     | 109 |
| 12.3    | E_EcMbxProtType.....                 | 109 |
| 12.4    | ST_EcCrcError .....                  | 109 |
| 12.5    | ST_EcCrcErrorEx .....                | 110 |
| 12.6    | ST_EcLastProtErrInfo .....           | 110 |
| 12.7    | ST_EcMasterStatistic .....           | 111 |
| 12.8    | ST_EcSlaveConfigData .....           | 111 |
| 12.9    | ST_EcSlaveIdentity .....             | 112 |
| 12.10   | ST_EcSlaveScannedData .....          | 112 |
| 12.11   | ST_EcSlaveState .....                | 113 |
| 12.12   | ST_EcSlaveStateBits.....             | 115 |
| 12.13   | ST_EcSlaveStateBitsEx.....           | 115 |
| 12.14   | ST_PortAddr.....                     | 115 |
| 12.15   | ST_TopologyDataEx .....              | 116 |
| 12.16   | DCTIMESTRUCT .....                   | 116 |
| 12.17   | T_DCTIME32 .....                     | 117 |
| 12.18   | T_DCTIME64 .....                     | 117 |
| 12.19   | T_DCTIME .....                       | 118 |
| 12.20   | T_HFoe .....                         | 118 |
| 13      | 常量.....                              | 120 |

|                              |            |
|------------------------------|------------|
| 13.1 全局常量 .....              | 120        |
| 13.2 库版本.....                | 121        |
| 13.3 EtherCAT 邮箱协议错误代码 ..... | 121        |
| <b>14 示例 .....</b>           | <b>123</b> |

# 1 前言

## 1.1 文档说明

本说明仅适用于熟悉国家标准且经过培训的控制和自动化工程专家。  
在安装和调试组件时，必须遵循文档和以下说明及解释。  
操作人员应具备相关资质，并始终使用最新的生效文档。

相关负责人员必须确保所述产品的应用或使用符合所有安全要求，包括所有相关法律、法规、准则和标准。

### 免责声明

本文档经过精心准备。然而，所述产品正在不断开发中。  
我们保留随时修改和更改本文档的权利，恕不另行通知。  
不得依据本文档中的数据、图表和说明对已供货产品的修改提出赔偿。

### 商标

Beckhoff®、TwinCAT®、TwinCAT/BSD®、TC/BSD®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS® 和 XPlanar® 是德国倍福自动化有限公司的注册商标并由其授权使用。

本出版物中所使用的其它名称可能是商标名称，任何第三方出于其自身目的使用它们可能会侵犯商标所有者的权利。



EtherCAT® 是注册商标和专利技术，由德国倍福自动化有限公司授权使用

### 版权所有

© 德国倍福自动化有限公司。  
未经明确授权，不得复制、分发、使用和传播本文档内容。  
违者将被追究赔偿责任。德国倍福自动化有限公司保留所有发明、实用新型和外观设计专利权。

### 第三方商标

本文档可能使用了第三方商标。有关商标信息，可以访问：<https://www.beckhoff.com/trademarks>。

## 1.2 安全信息

### 安全规范

为了确保您的使用安全，请务必仔细阅读并遵守本文档中每个产品的安全使用说明。

### 责任免除

所有组件在供货时都配有适合应用的特定硬件和软件配置。严禁未按文档所述修改硬件或软件配置，否则，德国倍福自动化有限公司对由此产生的后果不承担责任。

### 人员资格

本说明仅供熟悉适用国家标准的控制、自动化和驱动工程专家使用。

### 警示性词语

文档中使用的警示信号词分类如下。为避免人身伤害和财产损失，请阅读并遵守安全和警告注意事项。

#### 人身伤害警告

##### ⚠ 危险

存在死亡或重伤的高度风险。

##### ⚠ 警告

存在死亡或重伤的中度风险。

##### ⚠ 谨慎

存在可能导致中度或轻度伤害的低度风险。

#### 财产或环境损害警告

##### 注意

可能会损坏环境、设备或数据。

### 操作产品的信息



这些信息包括：  
有关产品的操作、帮助或进一步信息的建议。

## 1.3 信息安全说明

Beckhoff Automation GmbH & Co.KG (简称 Beckhoff) 的产品，只要可以在线访问，都配备了安全功能，支持工厂、系统、机器和网络的安全运行。尽管配备了安全功能，但为了保护相应的工厂、系统、机器和网络免受网络威胁，必须建立、实施和不断更新整个操作安全概念。Beckhoff 所销售的产品只是整个安全概念的一部分。客户有责任防止第三方未经授权访问其设备、系统、机器和网络。它们只有在采取了适当的保护措施的情况下，方可与公司网络或互联网连接。

此外，还应遵守 Beckhoff 关于采取适当保护措施的建议。关于信息安全和工业安全的更多信息，请访问本公司网站 <https://www.beckhoff.com/secguide>。

Beckhoff 的产品和解决方案持续进行改进。这也适用于安全功能。鉴于持续进行改进，Beckhoff 明确建议始终保持产品的最新状态，并在产品更新可用后马上进行安装。使用过时的或不支持的产品版本可能会增加网络威胁的风险。



如需了解 Beckhoff 产品信息安全的信息，请订阅 <https://www.beckhoff.com/secinfo> 上的 RSS 源。

## 2 概述

PLC 库 Tc2\_EtherCAT 是包含在 EtherCAT 主站设备和/或其从站设备上执行服务或功能的功能块。

### 3 EtherCAT 命令

#### 3.1 FB\_EcPhysicalReadCmd



功能块 FB\_EcPhysicalReadCmd 可用于向特定 EtherCAT 从站或所有 EtherCAT 从站发送 EtherCAT 读取命令（FPRD、APRD、BRD）。PLC 可使用该命令读取 EtherCAT 从站控制器的寄存器或 DPRAM。

输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    adp         : UINT;
    ado         : UINT;
    len         : UDINT;
    eType       : E_EcAdressingType := eAdressingType_Fixed;
    pDstBuf     : PVOID;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型                | 描述                                                            |
|----------|-------------------|---------------------------------------------------------------|
| sNetId   | T_AmsNetId        | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId）              |
| adp      | UINT              | 该值决定了该命令要寻址的 EtherCAT 从站。该值的含义取决于使用 eType 选择的寻址模式。（请参见 adp 值） |
| ado      | UINT              | 要读取的物理内存 (DPRAM) 或寄存器。                                        |
| len      | UDINT             | 要读取的字节数。                                                      |
| eType    | E_EcAdressingType | 根据 eType 的值发送不同的 EtherCAT 命令。（请参见 eType）                      |
| pDstBuf  | PVOID             | 接收缓冲器的地址（指针）。                                                 |
| bExecute | BOOL              | 功能块由该输入的上升沿激活。                                                |
| tTimeout | TIME              | 允许功能块执行的最长时间。                                                 |

**adp 值**

该值决定了该命令要寻址的 EtherCAT 从站。该值的含义取决于使用 eType 选择的寻址模式。

| eType                    | 描述                                                                                                                                                                 |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eAdressingType_Fixed     | 从站通过配置的 EtherCAT 地址寻址。这些 EtherCAT 地址可通过功能块 FB_EcGetAllSlaveAddr 读取。                                                                                                |
| eAdressingType_AutoInc   | 从站根据其在网络中的位置进行寻址。第一个设备的地址为 0 (adp=0)；其后所有从站的 adp 均递减 1：<br>1. 从站 adp = 0<br>2. 从站 adp = 16#ffff (-1)<br>3. 从站 adp = 16#fffe (-2)<br>4. 从站 adp = 16#fffd (-3)<br>等。 |
| eAdressingType_BroadCAST | 该命令可以寻址所有从站。可将 adp 设置为 0。                                                                                                                                          |

### eType

根据 eType 的值，会发送不同的 EtherCAT 命令：

| eType                    | 命令               |
|--------------------------|------------------|
| eAdressingType_Fixed     | 配置的地址物理读取 (FPRD) |
| eAdressingType_AutoInc   | 自动递增物理读数 (APRD)  |
| eAdressingType_BroadCAST | 广播读取 (BRD)       |

各个命令的寻址模式不同（请参见 adp）。

### 输出

```
VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    wkc : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                |
|--------|-------|-------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                      |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                              |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。                         |
| wkc    | UINT  | 每个成功处理该命令的 EtherCAT 从站都会使工作计数器递增。如果该命令只寻址了一个 EtherCAT 从站，则该值应为 1。 |

### 在 ST 中实施的示例：

```
PROGRAM TEST_PhysicalReadCmd
VAR
    fbReadCmd : FB_EcPhysicalReadCmd;
    bExecute : BOOL;
    value : UINT;
    adp : UINT:=16#3E9;
    ado : UINT:=16#1100;
    eType : E_EcAdressingType := eAdressingType_Fixed;
    sNetId : T_AmsNetId:='192.168.1.5.3.1';
    wkc : UINT;
    bError : BOOL;
    nErrId : UDINT;
END_VAR

fbReadCmd (sNetId:=sNetID, ado:=ado, adp:=adp, eType:=eType, LEN := SIZEOF(value), pDstBuf:=ADR(value), bExecute:=bExecute);
wkc := fbReadCmd.wkc;
bError:= fbReadCmd.bError;
nErrId:= fbReadCmd.nErrId;
```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm*) | Tc2_EtherCAT |

## 3.2 FB\_EcPhysicalWriteCmd



功能块 FB\_EcPhysicalWriteCmd 可用于向特定 EtherCAT 从站或所有 EtherCAT 从站发送 EtherCAT 写入命令（FPWR、APWR、BWR）。PLC 可使用该命令向寄存器或 EtherCAT 从站控制器的 DPRAM 写入数据。

 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    adp         : UINT;
    ado         : UINT;
    len         : UDINT;
    eType       : E_EcAdressingType := eAdressingType_Fixed;
    pSrcBuf     : PVOID;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型                | 描述                                                            |
|----------|-------------------|---------------------------------------------------------------|
| sNetId   | T_AmsNetId        | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId）              |
| adp      | UINT              | 该值决定了该命令要寻址的 EtherCAT 从站。该值的含义取决于使用 eType 选择的寻址模式。（请参见 adp 值） |
| ado      | UINT              | 要读取的物理内存 (DPRAM) 或寄存器。                                        |
| len      | UDINT             | 要写入的字节数。                                                      |
| eType    | E_EcAdressingType | 根据 eType 的值发送不同的 EtherCAT 命令：（请参见 eType）                      |
| pSrcBuf  | PVOID             | 传输缓冲器的地址（指针）。                                                 |
| bExecute | BOOL              | 功能块由该输入的上升沿激活。                                                |
| tTimeout | TIME              | 允许功能块执行的最长时间。                                                 |

adp 值

该值决定了该命令要寻址的 EtherCAT 从站。该值的含义取决于使用 eType 选择的寻址模式。

| eType                    | 描述                                                                                                                                                                 |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eAdressingType_Fixed     | 从站通过配置的 EtherCAT 地址寻址。这些 EtherCAT 地址可通过功能块 FB_EcGetAllSlaveAddr 读取。                                                                                                |
| eAdressingType_AutoInc   | 从站根据其在网络中的位置进行寻址。第一个设备的地址为 0 (adp=0)；其后所有从站的 adp 均递减 1：<br>1. 从站 adp = 0<br>2. 从站 adp = 16#ffff (-1)<br>3. 从站 adp = 16#fffe (-2)<br>4. 从站 adp = 16#fffd (-3)<br>等。 |
| eAdressingType_BroadCAST | 该命令可以寻址所有从站。应将 adp 设置为 0。                                                                                                                                          |

eType

根据 eType 的值发送不同的 EtherCAT 命令：

| eType                    | 命令               |
|--------------------------|------------------|
| eAdressingType_Fixed     | 配置的地址物理写入 (FPWR) |
| eAdressingType_AutoInc   | 自动递增物理写入 (APWR)  |
| eAdressingType_BroadCAST | 广播写入 (BWR)       |

各个命令的寻址模式不同（请参见 adp）。

### 输出

```
VAR_OUTPUT
  bBusy   : BOOL;
  bError   : BOOL;
  nErrId   : UDINT;
  wkc      : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                |
|--------|-------|-------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                      |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                              |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。                         |
| wkc    | UINT  | 每个成功处理该命令的 EtherCAT 从站都会使工作计数器递增。如果该命令只寻址了一个 EtherCAT 从站，则该值应为 1。 |

### 在 ST 中实施的示例：

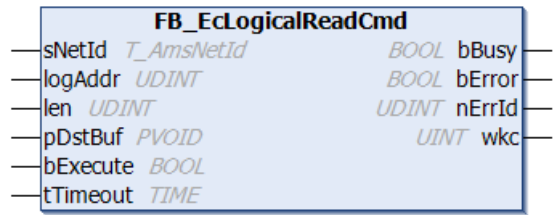
```
PROGRAM Test_PhysicalWriteCmd
VAR
  fbWriteCmd : FB_EcPhysicalWriteCmd;
  bExecute   : BOOL;
  value      : UINT :=16#5555;
  adp        : UDINT:=16#3E9;
  ado        : UDINT:=16#1100;
  eType      : E_EcAddressingType := eAddressingType_Fixed;
  sNetId     : T_AmsNetId:='192.168.1.5.3.1';
  wkc        : UDINT;
  bError     : BOOL;
  nErrId     : UDINT;
END_VAR

fbWriteCmd(sNetId:=sNetId, ado:=ado, adp:=adp, eType:=eType, LEN := SIZEOF(value), pSrcBuf:=ADR(value), bExecute:=bExecute);
wkc := fbWriteCmd.wkc;
bError:= fbWriteCmd.bError;
nErrId:= fbWriteCmd.nErrId;
```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 3.3 FB\_EcLogicalReadCmd



主站通过功能块 FB\_EcLogicalReadCmd 发送逻辑 EtherCAT 读取命令 (LRD)。在每个从站中，本地的地址范围 (DPRAM) 可映射到全局逻辑地址范围。因此，该命令可寻址所有已为所选逻辑地址范围配置映射的 EtherCAT 从站。

### 输入

```
VAR_INPUT
  sNetId : T_AmsNetId;
  logAddr : UDINT;
  len : UDINT;
  pDstBuf : PVOID;
  bExecute : BOOL;
  tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| logAddr  | UDINT      | 逻辑地址                                             |
| len      | UDINT      | 要读取的字节数                                          |
| pDstBuf  | PVOID      | 接收缓冲器的地址（指针）                                     |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    wkc        : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                |
|--------|-------|-------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                      |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                              |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。                         |
| wkc    | UINT  | 每个成功处理该命令的 EtherCAT 从站都会使工作计数器递增。如果该命令只寻址了一个 EtherCAT 从站，则该值应为 1。 |

在 ST 中实施的示例：

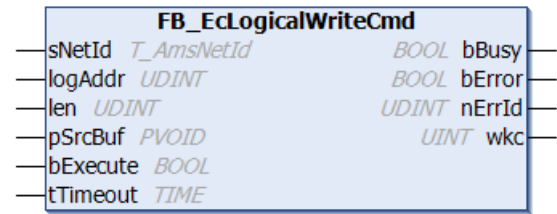
```
PROGRAM Test_LogicalReadCmd
VAR
    fbReadCmd : FB_EcLogicalReadCmd;
    bExecute  : BOOL;
    value     : USINT;
    logAddr   : UDINT := 16#10000;
    sNetId    : T_AmsNetId := '192.168.1.5.3.1';
    wkc       : UINT;
    bError    : BOOL;
    nErrId    : UDINT;
END_VAR

fbReadCmd (sNetId:=sNetID, logAddr:=logAddr, LEN := SIZEOF(value), pDstBuf:=ADR(value), bExecute:=bExecute);
wkc := fbReadCmd.wkc;
bError:= fbReadCmd.bError;
nErrId:= fbReadCmd.nErrId;
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

3.4 FB\_EcLogicalWriteCmd



主站通过功能块 FB\_EcLogicalWriteCmd 发送逻辑 EtherCAT 写入命令 (LWR)。在每个从站中，本地的地址范围 (DPRAM) 可映射到全局逻辑地址范围。因此，该命令可寻址所有已为所选逻辑地址范围配置映射的 EtherCAT 从站。

## 输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  logAddr     : UDINT;
  len         : UDINT;
  pSrcBuf     : PVOID;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| logAddr  | UDINT      | 逻辑地址                                             |
| len      | UDINT      | 要写入的字节数                                          |
| pSrcBuf  | PVOID      | 传输缓冲区的地址（指针）                                     |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

## 输出

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
  wkc         : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                |
|--------|-------|-------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                      |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                              |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。                         |
| wkc    | UINT  | 每个成功处理该命令的 EtherCAT 从站都会使工作计数器递增。如果该命令只寻址了一个 EtherCAT 从站，则该值应为 1。 |

## 在 ST 中实施的示例：

```
PROGRAM Test_LogicalWriteCmd
VAR
  fbWriteCmd : FB_EcLogicalWriteCmd;
  bExecute   : BOOL;
  value      : USINT := 16#55;
  logAddr    : UDINT := 16#10000;
  sNetId     : T_AmsNetId := '192.168.1.5.3.1';
  wkc        : UINT;
  bError     : BOOL;
  nErrId     : UDINT;
END_VAR

fbWriteCmd (sNetId:=sNetID, logAddr:=logAddr, LEN := SIZEOF(value), pSrcBuf:=ADR(value), bExecute:=bExecute);
wkc := fbWriteCmd.wkc;
bError := fbWriteCmd.bError;
nErrId := fbWriteCmd.nErrId;
```

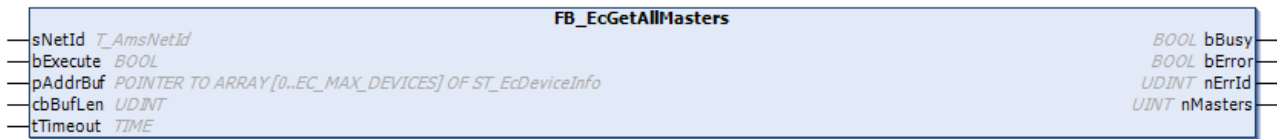
## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |



## 4 EtherCAT Diagnostic

### 4.1 FB\_EcGetAllMasters



功能块 FB\_EcGetAllMasters 可用于从所有 EtherCAT 主站读取信息。如果调用成功，那么 pAddrBuf 参数中传输的缓冲器会以 ST\_EcDeviceInfo 数组的形式包含所有主站的信息（设备 ID、AMS 网络 ID、名称）。

如果缓冲器过小，无法容纳所有现有主站的信息，那么缓冲器会被填满到指定大小，输出 nMasters 会输出所有现有主站的总数。

#### 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    pAddrBuf    : POINTER TO ARRAY[0..EC_MAX_DEVICES] OF ST_EcDeviceInfo;
    cbBufLen    : UDINT;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型                                                         | 描述                                                                                         |
|----------|------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| sNetId   | T_AmsNetId                                                 | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId）<br>默认 = 本地主机                              |
| bExecute | BOOL                                                       | 功能块由该输入的上升沿启用。                                                                             |
| pAddrBuf | POINTER TO ARRAY<br>[0..EC_MAX_DEVICES] OF ST_EcDeviceInfo | ST_EcDeviceInfo 数组的地址，各个主站的信息会写入该数组。<br>ST_EcDeviceInfo 包含设备 ID、AMS 网络 ID 和 EtherCAT 主站名称。 |
| cbBufLen | UDINT                                                      | 要读取的数据的最大可用缓冲器大小（以字节为单位）。                                                                  |
| tTimeout | TIME                                                       | 执行 ADS 调用期间不得超过的最长时间。                                                                      |

#### 输出

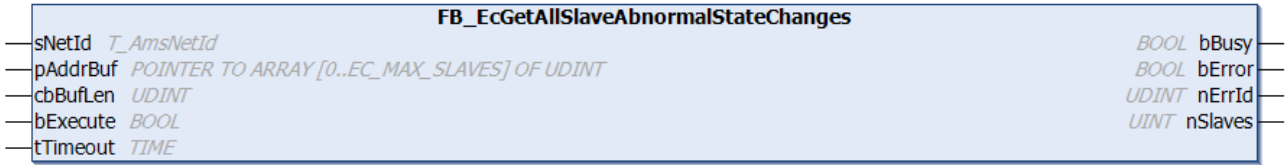
```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    nMasters   : UINT;
END_VAR
```

| 名称       | 类型    | 描述                                                                                |
|----------|-------|-----------------------------------------------------------------------------------|
| bBusy    | BOOL  | 激活功能块时设置该输出，并保持设置状态，直至收到反馈。                                                       |
| bError   | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                              |
| nErrId   | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。<br>举例说明：错误 1798 (0x706) 表示缓冲器地址处有一个空指针。 |
| nMasters | UINT  | 与主站相连的从站数量。                                                                       |

#### 前提条件

| 开发环境                    | 目标平台                   | 需要的 PLC 库                |
|-------------------------|------------------------|--------------------------|
| TwinCAT >= v3.1.4024.62 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT >= v3.6.1.0 |

4.2 FB\_EcGetAllSlaveAbnormalStateChanges



功能块 FB\_EcGetAllSlaveAbnormalStateChanges 可用于读取与主站相连的所有从站的意外 EtherCAT 状态更改。如果调用成功，那么参数 pBufAddr 中传输的缓冲器会以 UDINT 数组的形式包含所有从站的意外状态更改次数。如果 EtherCAT 主站未请求进行 EtherCAT 状态更改，例如 EtherCAT 从站自发地从 OP 状态切换为 SAFEOP 状态，则 EtherCAT 状态更改属于意外更改。

输入

```
VAR_INPUT
    sNetId : T_AmsNetId; (*AmsNetId of the EtherCAT master device*)
    pAddrBuf : POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT;
              (*Contains the address of the buffer the counters for the state changes f.i. Op to SafeOp-Err are copied to.*)
    cbBufLen : UDINT; (*Size of the buffer pAddrBuf. The size of the buffer must be at least nSlave *4 Bytes *)
    bExecute : BOOL; Function Block execution is triggered by a rising edge at this input*)
    tTimeout : TIME; (*States the time before the function is cancelled.*)
END_VAR
```

| 名称       | 类型                                           | 描述                                               |
|----------|----------------------------------------------|--------------------------------------------------|
| sNetId   | T_AmsNetId                                   | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| pAddrBuf | POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT | UDINT 数组的地址，用于写入各个从站的意外状态更改次数。                   |
| cbBufLen | UDINT                                        | 要读取的数据的最大可用缓冲器大小（以字节为单位）                         |
| bExecute | BOOL                                         | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME                                         | 允许功能块执行的最长时间。                                    |

输出

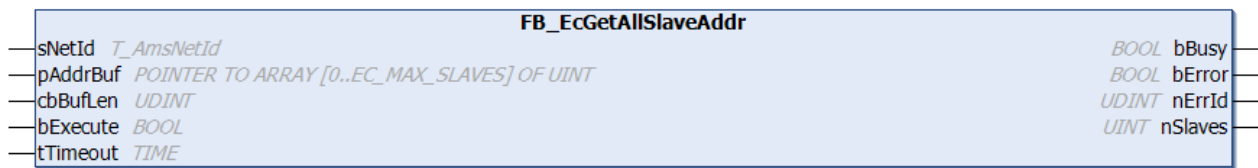
```
VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    nSlaves : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 与主站相连的从站数量。                                                                                        |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.3 FB\_EcGetAllSlaveAddr



通过 FB\_EcGetAllSlaveAddr 功能块，可以读取与主站相连的所有从站的地址。调用成功后，参数 pAddrBuf 中传递的缓冲器会以 UINT 数组的形式包含所有从站的地址。

### 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    pAddrBuf    : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF UINT;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型                                                 | 描述                                               |
|----------|----------------------------------------------------|--------------------------------------------------|
| sNetId   | T_AmsNetId                                         | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| pAddrBuf | POINTER TO ARRAY<br>[0..EC_MAX_SLAVES] OF UIN<br>T | UINT 数组的地址，用于写入各个从站的地址。                          |
| cbBufLen | UDINT                                              | 要读取的数据的最大可用缓冲器大小（以字节为单位）。                        |
| bExecute | BOOL                                               | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME                                               | 允许功能块执行的最长时间。                                    |

### 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    nSlaves    : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 与主站相连的从站数量。                                                                                        |

### 在 ST 中实施的示例：

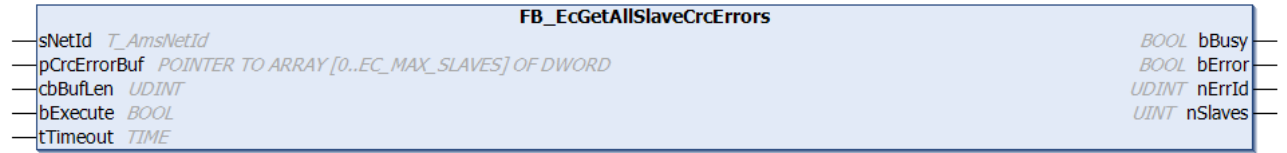
```
PROGRAM TEST_GetAllSlaveAddresses
VAR
    fbGetAllSlaveAddr : FB_EcGetAllSlaveAddr;
    sNetId            : T_AmsNetId := '172.16.2.131.2.1';
    bExecute          : BOOL;
    slaveAddresses    : ARRAY[0..255] OF UINT;
    nSlaves           : UINT := 0;
    bError            : BOOL;
    nErrId            : UDINT;
END_VAR

fbGetAllSlaveAddr(sNetId:= sNetId, pAddrBuf := ADR(slaveAddresses), cbBufLen:= SIZEOF(slaveAddresses)
, bExecute:=bExecute);
nSlaves := fbGetAllSlaveAddr.nSlaves;
bError := fbGetAllSlaveAddr.bError;
nErrId := fbGetAllSlaveAddr.nErrId;
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm*) | Tc2_EtherCAT |

4.4 FB\_EcGetAllSlaveCrcErrors



通过 FB\_EcGetAllSlaveCrcErrors 功能块，可以读取与主站相连的所有从站的 CRC 错误计数器。会添加从站各个端口的 CRC 错误。

要读取从站各个端口（A、B 和 C）的 CRC 错误，必须调用 FB\_EcGetSlaveCrcError [► 27] 功能块。

要读取从站各个端口（A、B、C 和 D）的 CRC 错误，必须调用 FB\_EcGetSlaveCrcErrorEx [► 28] 功能块。

输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  pCrcErrorBuf : POINTER TO ARRAY [0..EC_MAX_SLAVES] OF DWORD;
  cbBufLen    : UDINT;
  bExecute     : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称           | 类型                                           | 描述                                               |
|--------------|----------------------------------------------|--------------------------------------------------|
| sNetId       | T_AmsNetId                                   | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| pCrcErrorBuf | POINTER TO ARRAY [0..EC_MAX_SLAVES] OF DWORD | DWORD 类型的数组的地址，用于写入 CRC 错误计数器。                   |
| cbBufLen     | UDINT                                        | 要读取的数据的最大可用缓冲器大小（以字节为单位）                         |
| bExecute     | BOOL                                         | 功能块由该输入的上升沿激活。                                   |
| tTimeout     | TIME                                         | 允许功能块执行的最长时间。                                    |

输出

```
VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  nErrId : UDINT;
  nSlaves : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 与主站相连的从站数量。                                                                                        |

在 ST 中实施的示例：

```
PROGRAM TEST_GetAllSlaveCrcErrors
VAR
  fbGetAllSlaveCrcErrors : FB_EcGetAllSlaveCrcErrors;
  sNetId                 : T_AmsNetId := '172.16.2.131.2.1';
END_VAR
```

```

    bExecute          : BOOL;
    crcErrors         : ARRAY[0..255] OF DWORD;
    nSlaves           : UINT := 0;
    bError            : BOOL;
    nErrId            : UDINT;
END_VAR

fbGetAllSlaveCrcErrors(sNetId:= sNetId, pCrcErrorBuf := ADR(crcErrors), cbBufLen:= SIZEOF(crcErrors)
, bExecute:=bExecute);
nSlaves := fbGetAllSlaveCrcErrors.nSlaves;
bError := fbGetAllSlaveCrcErrors.bError;
nErrId := fbGetAllSlaveCrcErrors.nErrId;

```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.5 FB\_EcGetAllSlavePresentStateChanges

| FB_EcGetAllSlavePresentStateChanges |                                              |       |         |
|-------------------------------------|----------------------------------------------|-------|---------|
| sNetId                              | T_AmsNetId                                   | BOOL  | bBusy   |
| pAddrBuf                            | POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT | BOOL  | bError  |
| cbBufLen                            | UDINT                                        | UDINT | nErrId  |
| bExecute                            | BOOL                                         | UINT  | nSlaves |
| tTimeout                            | TIME                                         |       |         |

功能块 FB\_EcGetAllSlavePresentStateChanges 可用于读取所有与主站相连的从站从 “slave is present” 状态到 “INIT\_NO\_COMM” 状态的 EtherCAT 状态更改。如果调用成功，那么参数 pBufAddr 中传输的缓冲器会以 UDINT 数组的形式包含所有从站的状态更改次数。EtherCAT 状态从 “slave is present” 变为 “INIT\_NO\_COMM” 意味着与从站的连接已中断。例如，断开 EtherCAT 电缆的连接。

### 输入

```

VAR_INPUT
    sNetId      : T_AmsNetId; (*AmsNetId of the EtherCAT master device*)
    pAddrBuf    : POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT; (*Contains the address of the buffer
the counters for the state changes from Present to INIT_NO_COMM are copied to. *)
    cbBufLen    : UDINT; (*Size of the buffer pAddrBuf. The size of the buffer must be at least nSlav
e * 4 Bytes *)
    bExecute    : BOOL; (*Function Block execution is triggered by a rising edge at this input*)
    tTimeout    : TIME; (*States the time before the function is cancelled.*)
END_VAR

```

| 名称       | 类型                                           | 描述                                                                  |
|----------|----------------------------------------------|---------------------------------------------------------------------|
| sNetId   | T_AmsNetId                                   | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId）                    |
| pAddrBuf | POINTER TO ARRAY [0..EC_MAX_SLAVES] OF UDINT | UDINT 数组的地址，用于写入各个从站从 “slave is present” 变为 “INIT_NO_COMM” 的状态更改次数。 |
| cbBufLen | UDINT                                        | 要读取的数据的最大可用缓冲器大小（以字节为单位）                                            |
| bExecute | BOOL                                         | 功能块由该输入的上升沿激活。                                                      |
| tTimeout | TIME                                         | 允许功能块执行的最长时间。                                                       |

### 输出

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    nSlaves    : UINT;
END_VAR

```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 与主站相连的从站数量。                                                                                        |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.6 FB\_EcGetConfSlaves



功能块 FB\_EcGetConfSlaves 可用于从 EtherCAT 主站对象目录中读取已配置的从站列表。

#### 输入

```

VAR_INPUT
    sNetId          : T_AmsNetId;
    pArrEcConfSlaveInfo : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveConfigData;
    cbBufLen        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

| 名称                  | 类型                                                          | 描述                                                     |
|---------------------|-------------------------------------------------------------|--------------------------------------------------------|
| sNetId              | T_AmsNetId                                                  | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId）       |
| pArrEcConfSlaveInfo | POINTER TO ARRAY [0..EC_MAX_SLAVES] OF ST_EcSlaveConfigData | ST_EcSlaveConfigData [► 111] 类型结构数组的地址，用于写入每个已配置从站的数据。 |
| cbBufLen            | UDINT                                                       | 要读取的数据的最大可用缓冲器大小（以字节为单位）                               |
| bExecute            | BOOL                                                        | 功能块由该输入的上升沿激活。                                         |
| tTimeout            | TIME                                                        | 允许功能块执行的最长时间。                                          |

#### 输出

```

VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    nSlaves : UINT;
END_VAR

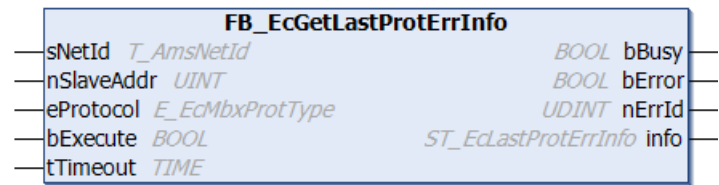
```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 返回已配置从站的数量。                                                                                        |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

4.7 FB\_EcGetLastProtErrInfo



功能块 FB\_EcGetLastProtErrInfo 可用于读取与近期邮箱协议错误有关的附加错误信息。错误清除 邮箱命令每次都会重置上次的错误。

🔧 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    eProtocol   : E_EcMbxProtType := eEcMbxProt_FoE;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型              | 描述                                               |
|------------|-----------------|--------------------------------------------------|
| sNetId     | T_AmsNetId      | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT            | 要读取其错误信息的 EtherCAT 从站的固定地址。                      |
| eProtocol  | E_EcMbxProtType | <a href="#">EtherCAT 邮箱协议类型 [► 109]</a>          |
| bExecute   | BOOL            | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME            | 允许功能块执行的最长时间。                                    |

🔧 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    info       : ST_EcLastProtErrInfo;
END_VAR
```

| 名称     | 类型                       | 描述                                        |
|--------|--------------------------|-------------------------------------------|
| bBusy  | BOOL                     | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL                     | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT                    | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| info   | ST_EcLastProtErr<br>Info | 附加错误信息 [► 110] 的结构                        |

### ST 中的示例：

bGet 的上升沿会触发读取与最近邮箱协议错误有关的附加错误信息。

```

PROGRAM MAIN
VAR
    fbGetInfo : FB_EcGetLastProtErrInfo := ( sNetID := '172.16.6.195.2.1',
                                              nSlaveAddr := 1004,
                                              eProtocol := eEcMbxProt_FoE,
                                              tTimeout := DEFAULT_ADS_TIMEOUT );

    bGet : BOOL;
    bBusy : BOOL;
    bError : BOOL;
    nErrID : UDINT;
    sInfo : T_MaxString;
END_VAR

fbGetInfo( bExecute:= bGet,
          bBusy=>bBusy,
          bError=>bError,
          nErrId=>nErrId );

sInfo := BYTEARR_TO_MAXSTRING( fbGetInfo.info.binDesc );
    
```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.8 FB\_EcGetMasterDevState



功能块 FB\_EcGetMasterDevState 可用于读取 EtherCAT 主站的当前状态。

### 输入

```

VAR_INPUT
    sNetId : T_AmsNetId;
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
    
```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

### 输出

```

VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    
```



```

    nErrId   : UDINT;
    nDevState : WORD;
END_VAR

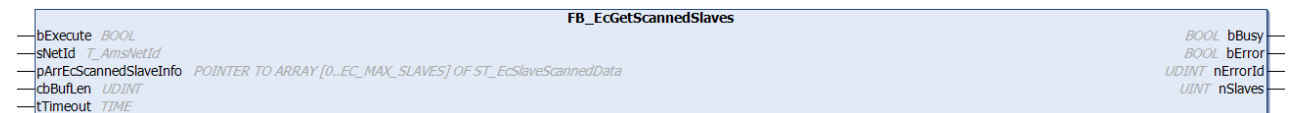
```

| 名称        | 类型    | 描述                                        |
|-----------|-------|-------------------------------------------|
| bBusy     | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError    | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId    | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| nDevState | WORD  | 主设备的当前状态                                  |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.9 FB\_EcGetScannedSlaves



功能块 FB\_EcGetScannedSlaves 可用于从 EtherCAT 主站对象目录中读取当前可用（已扫描）的从站列表。为此，会执行一次在线扫描，在扫描过程中读取 EtherCAT 从站的 EEPROM。扫描过程可能需要一些时间，具体取决于相连从站的数量。

#### 输入

```

VAR_INPUT
    bExecute      : BOOL;
    sNetId        : T_AmsNetId;
    pArrEcScannedSlaveInfo : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveScannedData;
    cbBufLen      : UDINT;
    tTimeout      : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

| 名称                     | 类型                                                          | 描述                                                      |
|------------------------|-------------------------------------------------------------|---------------------------------------------------------|
| bExecute               | BOOL                                                        | 功能块由该输入的上升沿激活。                                          |
| sNetId                 | T_AmsNetId                                                  | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId）        |
| pArrEcScannedSlaveInfo | POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveScannedData | ST_EcSlaveScannedData [► 112] 类型结构数组的地址，用于写入每个已扫描从站的数据。 |
| cbBufLen               | UDINT                                                       | 要读取的数据的最大可用缓冲器大小（以字节为单位）                                |
| tTimeout               | TIME                                                        | 允许功能块执行的最长时间。                                           |

#### 输出

```

VAR_OUTPUT
    bBusy   : BOOL;
    bError  : BOOL;
    nErrId  : UDINT;
    nSlaves : UINT;
END_VAR

```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 返回已扫描从站的数量。                                                                                        |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.10 FB\_EcGetSlaveCount



功能块 FB\_EcGetSlaveCount 可用于确定与主站相连的从站数量。

#### 输入

```

VAR_INPUT
    sNetId : T_AmsNetId;
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

#### 输出

```

VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    nSlaves : UINT;
END_VAR

```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| nSlave | UINT  | 与主站相连的从站数量                                |

在 ST 中实施的示例：

```
PROGRAM TEST_GetSlaveCount
VAR
    fbGetSlaveCount : FB_EcGetSlaveCount;
    sNetId           : T_AmsNetId := '172.16.2.131.2.1';
    bExecute         : BOOL;
    nSlaves           : UINT;
    bError           : BOOL;
    nErrId           : UDINT;
END_VAR

fbGetSlaveCount(sNetId:= sNetId, bExecute:=bExecute);
nSlaves := fbGetSlaveCount.nSlaves;
bError := fbGetSlaveCount.bError;
nErrId := fbGetSlaveCount.nErrId;
```

前提条件

| 开发环境           | 目标平台                               | 需要的 PLC 库    |
|----------------|------------------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm <sup>®</sup> ） | Tc2_EtherCAT |

4.11 FB\_EcGetSlaveCrcError



通过功能块 FB\_EcGetSlaveCrcError，可以读取从站各个端口（A、B 和 C）的 CRC 错误计数器。如果调用成功，那么输出变量 crcError（其类型为 ST\_EcCrcError）会包含请求的 CRC 错误计数器。

功能块 FB\_EcGetSlaveCrcError 只能用于最多有 3 个端口的从站（如 EK1100）。功能块 FB\_EcGetSlaveCrcErrorEx 也可用于最多有 4 个端口的从站（如 EK1122）。

🔴 输入

```
VAR_INPUT
    sNetId       : T_AmsNetId;
    nSlaveAddr   : UINT;
    bExecute     : BOOL;
    tTimeout     : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 要读取其 CRC 错误计数器的 EtherCAT 从站的固定地址。                |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

🔴 输出

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    crcError    : ST_EcCrcError;
END_VAR
```

| 名称       | 类型            | 描述                                        |
|----------|---------------|-------------------------------------------|
| bBusy    | BOOL          | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError   | BOOL          | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId   | UDINT         | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| crcError | ST_EcCrcError | 各个端口的 CRC 错误 [► 109]计数器                   |

#### 在 ST 中实施的示例：

```

PROGRAM TEST_GetSlaveCrcError
VAR
    fbGetSlaveCrcError : FB_EcGetSlaveCrcError;
    sNetId : T_AmsNetId := '172.16.2.131.2.1';
    bExecute : BOOL;
    crcError : ST_EcCrcError;
    nSlaveAddr : UDINT := 1001;
    bError : BOOL;
    nErrId : UDINT;
END_VAR

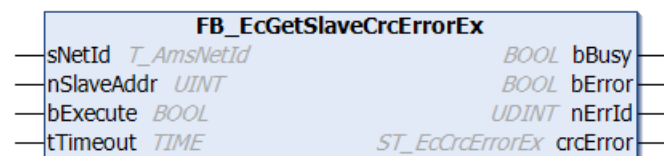
fbGetSlaveCrcError(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute);
crcError := fbGetSlaveCrcError.crcError;
bError := fbGetSlaveCrcError.bError;
nErrId := fbGetSlaveCrcError.nErrId;

```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.12 FB\_EcGetSlaveCrcErrorEx



通过功能块 FB\_EcGetSlaveCrcErrorEx，可以读取从站各个端口（A、D、B 和 C）的 CRC 错误计数器。如果调用成功，那么输出变量 crcError（其类型为 ST\_EcCrcErrorEx）会包含请求的 CRC 错误计数器。

功能块 FB\_EcGetSlaveCrcErrorEx 也可用于最多有 4 个端口的从站（如 EK1122）。功能块 FB\_EcGetSlaveCrcError 只能用于最多有 3 个端口的从站（如 EK1100）。

#### 输入

```

VAR_INPUT
    sNetId      : T_AmsNetId; (*AmsNetId of the EtherCAT master device*)
    nSlaveAddr  : UDINT; (*Address of the slave device*)
    bExecute    : BOOL; (*Function block execution is triggered by a rising edge at this input.*)
    tTimeout    : TIME; (*States the time before the function is cancelled.*)
END_VAR

```

| 名称         | 类型         | 描述                                              |
|------------|------------|-------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备 AMS 网络 ID 的字符串（类型：T_AmsNetId）。 |
| nSlaveAddr | UDINT      | 要读取其 CRC 错误计数器的 EtherCAT 从站的固定地址。               |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                  |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                   |

#### 输出

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;

```

```

nErrId      : UDINT;
CrcError    : ST_EcCrcErrorEx; (*Crc error of the EtherCAT slave device*)
END_VAR
```

| 名称       | 类型              | 描述                                               |
|----------|-----------------|--------------------------------------------------|
| bBusy    | BOOL            | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                     |
| bError   | BOOL            | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。             |
| nErrId   | UDINT           | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。        |
| CrcError | ST_EcCrcErrorEx | 各个端口的 CRC 错误计数器<br>(类型: ST_EcCrcErrorEx [► 110]) |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

4.13 FB\_EcGetSlaveIdentity



功能块 FB\_EcGetSlaveIdentity 可用于读取各个 EtherCAT 从站设备的 CANopen 标识。如果调用成功，则输出变量标识（其类型为 ST\_EcSlaveIdentity）会包含所请求的标识信息。

输入

```

VAR_INPUT
sNetId      : T_AmsNetId;
nSlaveAddr  : UINT;
bExecute    : BOOL;
tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型 T_AmsNetId） |
| nSlaveAddr | UINT       | EtherCAT 从站的固定地址                                 |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

输出

```

VAR_OUTPUT
bBusy       : BOOL;
bError      : BOOL;
nErrId      : UDINT;
identity    : ST_EcSlaveIdentity;
END_VAR
```

| 名称       | 类型                 | 描述                                        |
|----------|--------------------|-------------------------------------------|
| bBusy    | BOOL               | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError   | BOOL               | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId   | UDINT              | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| identity | ST_EcSlaveIdentity | EtherCAT 设备的 CANopen 标识 [► 112]           |

#### 在 ST 中实施的示例：

```

PROGRAM TEST_GetSlaveIdentity
VAR
    fbGetSlaveIdentity : FB_EcGetSlaveIdentity;
    sNetId              : T_AmsNetId := '172.16.2.131.2.1';
    bExecute            : BOOL;
    identity            : ST_EcSlaveIdentity;
    nSlaveAddr         : UDINT := 1001;
    bError              : BOOL;
    nErrId             : UDINT;
END_VAR

fbGetSlaveIdentity(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute);
identity := fbGetSlaveIdentity.identity;
bError := fbGetSlaveIdentity.bError;
nErrId := fbGetSlaveIdentity.nErrId;

```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.14 FB\_EcGetSlaveTopologyInfo

| FB_EcGetSlaveTopologyInfo                                         |               |
|-------------------------------------------------------------------|---------------|
| sNetId T_AmsNetId                                                 | BOOL bBusy    |
| pAddrBuf POINTER TO ARRAY [0..EC_MAX_SLAVES] OF ST_TopologyDataEx | BOOL bError   |
| cbBufLen UDINT                                                    | UDINT nErrId  |
| bExecute BOOL                                                     | UDINT nSlaves |
| tTimeout TIME                                                     |               |

功能块 FB\_EcGetSlaveTopologyInfo 可用于确定拓扑信息。

#### 输入

```

VAR_INPUT
    sNetId      : T_AmsNetId; (*AmsNetId of the EtherCAT master device*)
    pAddrBuf    : POINTER TO ARRAY [0..EC_MAX_SLAVES] OF ST_TopologyDataEx; (*Contains the address of
the buffer the topology data are copied to.*)
    cbBufLen    : UDINT; (*Size of the buffer pAddrBuf. The size of the buffer must be at least nSlav
e * 64 Bytes*)
    bExecute    : BOOL; (*Function block execution is triggered by a rising edge at this input*)
    tTimeout    : TIME; (*States the time before the function is cancelled*)
END_VAR

```

| 名称       | 类型                                                       | 描述                                               |
|----------|----------------------------------------------------------|--------------------------------------------------|
| sNetId   | T_AmsNetId                                               | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| pAddrBuf | POINTER TO ARRAY [0..EC_MAX_SLAVES] OF ST_TopologyDataEx | 包含拓扑数据的 ST_TopologyDataEx [► 116] 类型结构数组的地址。     |
| cbBufLen | UDINT                                                    | 要读取的数据的最大可用缓冲器大小（以字节为单位）                         |
| bExecute | BOOL                                                     | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME                                                     | 允许功能块执行的最长时间。                                    |

 输出

```

VAR_OUTPUT
  bBusy    : BOOL;
  bError   : BOOL;
  nErrId   : UDINT;
  nSlaves  : UINT;
END_VAR

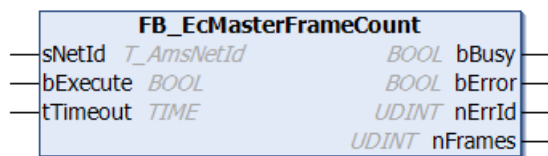
```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 与主站相连的从站数量。                                                                                        |

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.15 FB\_EcMasterFrameCount



功能块 FB\_EcMasterFrameCount 可用于确定主站中配置的 EtherCAT 帧数。

 输入

```

VAR_INPUT
  sNetId    : T_AmsNetId;
  bExecute  : BOOL;
  tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

 输出

```

VAR_OUTPUT
  bBusy    : BOOL;
  bError   : BOOL;
  nErrId   : UDINT;
  nFrames  : UDINT;
END_VAR

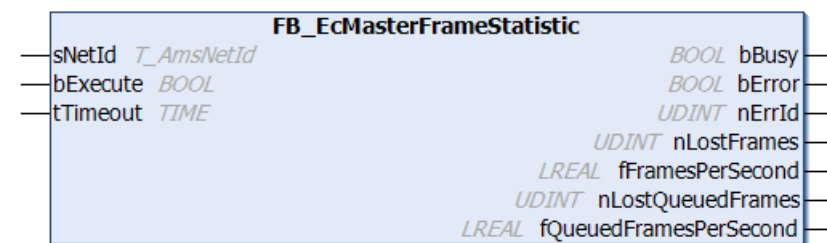
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| nFrame | UDINT | EtherCAT 帧数                               |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.16 FB\_EcMasterFrameStatistic



功能块 FB\_EcMasterFrameStatistic 可用于读取 EtherCAT 主站的帧统计信息。循环帧和非循环（排队）之间有区别。非循环帧用于初始化或 EtherCAT 从站的参数访问。如果帧未能返回主站或无效，则视为丢失。

功能块输出可提供丢失的帧数（即丢失帧或无效循环帧）、每秒循环帧数、丢失排队帧数（即丢失帧或无效非循环帧）和每秒排队帧数。

#### 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

#### 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    nLostFrames : UDINT;
    fFramesPerSecond : LREAL;
    nLostQueuedFrames : UDINT;
    fQueuedFramesPerSecond : LREAL;
END_VAR
```

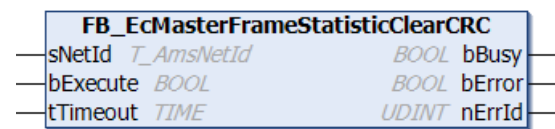


| 名称                     | 类型    | 描述                                        |
|------------------------|-------|-------------------------------------------|
| bBusy                  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError                 | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId                 | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| nLostFrame             | UDINT | 返回当前丢失帧或无效循环帧的数量。                         |
| fFramesPerSecond       | LREAL | 返回当前每秒循环帧数。                               |
| nLostQueuedFrame       | UDINT | 返回当前丢失帧或无效排队（非循环）帧的数量。                    |
| fQueuedFramesPerSecond | LREAL | 返回当前每秒排队（非循环）帧的数量。                        |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 4.17 FB\_EcMasterFrameStatisticClearCRC



功能块 FB\_EcMasterFrameStatisticClearCRC 可用于删除所有 EtherCAT 从站的 CRC 错误计数器。

#### 输入

```

VAR_INPUT
    sNetId    : T_AmsNetId;
    bExecute  : BOOL;
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

#### 输出

```

VAR_OUTPUT
    bBusy    : BOOL;
    bError   : BOOL;
    nErrId   : UDINT;
END_VAR

```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

# 4.18 FB\_EcMasterFrameStatisticClearFrames



功能块 FB\_EcMasterFrameStatisticClearFrames 可用于删除丢失帧计数器。

输入

```

VAR_INPUT
    sNetId    : T_AmsNetId;
    bExecute  : BOOL;
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

输出

```

VAR_OUTPUT
    bBusy    : BOOL;
    bError   : BOOL;
    nErrId   : UDINT;
END_VAR

```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

# 4.19 FB\_EcMasterFrameStatisticClearTxRxErr



功能块 FB\_EcMasterFrameStatisticClearTxRxErr 可用于删除网卡迷你端口驱动器的错误计数器。

🚩 输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  nEcMasterDevID : INT;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称             | 类型         | 描述                                         |
|----------------|------------|--------------------------------------------|
| sNetId         | T_AmsNetId | 包含 CPU (PC) AMS 网络 ID 的字符串。（类型：T_AMSNetId） |
| nEcMasterDevID | INT        | EtherCAT 主站的设备 ID。                         |
| bExecute       | BOOL       | 功能块由该输入的上升沿激活。                             |
| tTimeout       | TIME       | 允许功能块执行的最长时间。                              |

🚩 输出

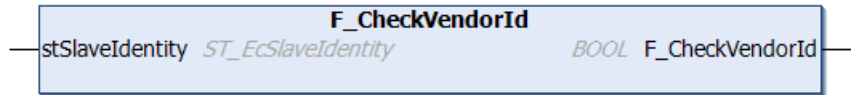
```
VAR_OUTPUT
  bBusy  : BOOL;
  bError : BOOL;
  nErrId : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

4.20 F\_CheckVendorId



如果 VendorID 是倍福，那么 F\_CheckVendorId 功能会返回 TRUE（真），否则返回 FALSE（假）。

🚩 返回值

```
METHOD F_CheckVendorId : BOOL
```

| 名称              | 类型   | 描述                                       |
|-----------------|------|------------------------------------------|
| F_CheckVendorId | BOOL | 如果 VendorID 是倍福，则为 TRUE（真），否则为 FALSE（假）。 |

🚩 输入

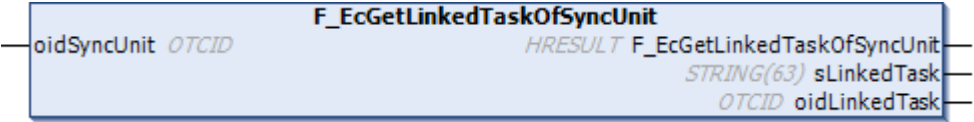
```
VAR_INPUT
  stSlaveIdentity : ST_EcSlaveIdentity;
END_VAR
```

| 名称              | 类型                 | 描述                                        |
|-----------------|--------------------|-------------------------------------------|
| stSlaveIdentity | ST_EcSlaveIdentity | 从站标识，可通过 FB_EcGetSlaveIdentity [► 29] 读取。 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

# 4.21 F\_EcGetLinkedTaskOfSyncUnit



使用该功能可读取 EtherCAT 同步单元的链接任务的名称和对象 ID。该功能的返回值代表调用是否成功的信号，如果出现错误，则会输出相应的错误代码。

返回值

METHOD F\_EcGetLinkedTaskOfSyncUnit : HRESULT

| 名称                          | 类型      | 描述                          |
|-----------------------------|---------|-----------------------------|
| F_EcGetLinkedTaskOfSyncUnit | HRESULT | 调用是否成功的信号，并在出现错误时输出相应的错误代码。 |

输入

```
VAR_INPUT
    oidSyncUnit : OTCID; // object ID of sync unit
END_VAR
```

| 名称          | 类型    | 描述                                          |
|-------------|-------|---------------------------------------------|
| oidSyncUnit | OTCID | 在此输入位置指定同步单元的对象 ID。可以在 EtherCAT 主站的过程图像中找到。 |

输出

```
VAR_OUTPUT
    sLinkedTask : STRING;
    oidLinkedTask : OTCID; // object ID of linked task
END_VAR
```

| 名称            | 类型     | 描述            |
|---------------|--------|---------------|
| sLinkedTask   | STRING | 返回链接任务的名称。    |
| oidLinkedTask | OTCID  | 返回链接任务的对象 ID。 |

前提条件

| 开发环境                 | 目标平台                   | 需要的 PLC 库                |
|----------------------|------------------------|--------------------------|
| TwinCAT v3.1.4024.22 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT >= 3.3.17.0 |

# 4.22 F\_EcGetSyncUnitName



使用该功能可通过对象 ID 读取 EtherCAT 同步单元的名称。该功能的返回值代表调用是否成功的信号，如果出现错误，则输出相关的错误代码。

🔴 返回值

```
METHOD F_EcGetSyncUnitName : HRESULT
```

| 名称                  | 类型      | 描述                          |
|---------------------|---------|-----------------------------|
| F_EcGetSyncUnitName | HRESULT | 调用是否成功的信号，并在出现错误时输出相关的错误代码。 |

🔴 输入

```
VAR_INPUT
    oidSyncUnit : OTCID; // object ID of sync unit
END_VAR
```

| 名称          | 类型    | 描述                                          |
|-------------|-------|---------------------------------------------|
| oidSyncUnit | OTCID | 在此输入位置指定同步单元的对象 ID。可以在 EtherCAT 主站的过程图像中找到。 |

🔴 输出

```
VAR_OUTPUT
    sSyncUnitName : STRING(63);
END_VAR
```

| 名称            | 类型     | 描述         |
|---------------|--------|------------|
| sSyncUnitName | STRING | 返回同步单元的名称。 |

前提条件

| 开发环境                 | 目标平台                   | 需要的 PLC 库               |
|----------------------|------------------------|-------------------------|
| TwinCAT v3.1.4024.48 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT >= 3.4.2.0 |

# 5 EtherCAT 状态机

## 5.1 FB\_EcGetAllSlaveStates



通过 FB\_EcGetAllSlaveStates 功能块可读取 EtherCAT 状态以及与主站相连的所有从站的链接状态。调用成功时，参数 pStateBuf 中传递的缓冲器会以 ST\_EcSlaveState 数组的形式包含所请求的状态信息。

### 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    pStateBuf   : POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveState;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称        | 类型                                                    | 描述                                               |
|-----------|-------------------------------------------------------|--------------------------------------------------|
| sNetId    | T_AmsNetId                                            | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| pStateBuf | POINTER TO ARRAY[0..EC_MAX_SLAVES] OF ST_EcSlaveState | 要写入从站状态的 ST_EcSlaveStates [113] 数组的地址。           |
| cbBufLen  | UDINT                                                 | 要读取的数据的最大可用缓冲器大小（以字节为单位）                         |
| bExecute  | BOOL                                                  | 功能块由该输入的上升沿激活。                                   |
| tTimeout  | TIME                                                  | 允许功能块执行的最长时间。                                    |

### 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    nSlaves    : UINT;
END_VAR
```

| 名称     | 类型    | 描述                                                                                                 |
|--------|-------|----------------------------------------------------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                       |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                               |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。错误 1798 (0x706) 表明缓冲器地址处有一个空指针。错误 1797 (0x705) 表明缓冲器大小不足。 |
| nSlave | UINT  | 与主站相连的从站数量                                                                                         |

### 在 ST 中实施的示例：

```
PROGRAM TEST_GetAllSlaveStates
VAR
    fbGetAllSlaveStates : FB_EcGetAllSlaveStates;
    sNetId              : T_AmsNetId := '172.16.2.131.2.1';
    bExecute            : BOOL;
    devStates           : ARRAY[0..255] OF ST_EcSlaveState;
    nSlaves             : UINT := 0;
    bError              : BOOL;
    nErrId             : UDINT;
END_VAR

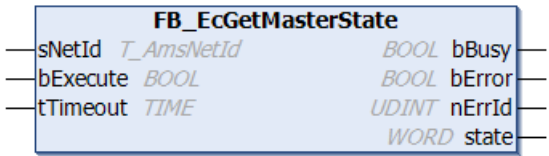
fbGetAllSlaveStates(sNetId:= sNetId, pStateBuf := ADR(devStates), cbBufLen:=SIZEOF(devStates), bExec
```

```
ute:=bExecute);
nSlaves := fbGetAllSlaveStates.nSlaves;
bError := fbGetAllSlaveStates.bError;
nErrId := fbGetAllSlaveStates.nErrId;
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

5.2 FB\_EcGetMasterState



功能块 FB\_EcGetMasterState 可用于读取主站的 EtherCAT 状态。如果调用成功，则 WORD 类型的状态输出变量会包含所请求的状态信息。

🔧 输入

```
VAR_INPUT
    sNetId : T_AmsNetId;
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |

🔧 输出

```
VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    nErrId : UDINT;
    state : WORD;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| state  | WORD  | 主站当前的 EtherCAT 状态。（请参见状态）                 |

state

主站当前的 EtherCAT 状态。可能的值为：

| 常量                     | 值    | 描述               |
|------------------------|------|------------------|
| EC_DEVICE_STATE_INIT   | 0x01 | 主站处于 Init（初始化）状态 |
| EC_DEVICE_STATE_PREOP  | 0x02 | 主站处于准备运行状态       |
| EC_DEVICE_STATE_SAFEOP | 0x04 | 主站处于安全运行状态       |
| EC_DEVICE_STATE_OP     | 0x08 | 主站处于运行状态         |

### 在 ST 中实施的示例：

```
PROGRAM TEST_GetMasterState
VAR
    fbGetMasterState : FB_EcGetMasterState;
    sNetId            : T_AmsNetId := '172.16.2.131.2.1';
    bExecute          : BOOL;
    state             : WORD;
    bError            : BOOL;
    nErrId            : UDINT;
END_VAR

fbGetMasterState(sNetId:= sNetId, bExecute:=bExecute);
state := fbGetMasterState.state;
bError := fbGetMasterState.bError;
nErrId := fbGetMasterState.nErrId;
```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 5.3 FB\_EcGetSlaveState



通过功能块 FB\_EcGetSlaveState 可以读取 EtherCAT 状态和各个 EtherCAT 从站的链接状态。如果调用成功，则输出变量状态（其类型为 ST\_EcSlaveState）会包含所请求的状态信息。

### 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 要读取其状态的 EtherCAT 从站的固定地址。                        |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

### 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    state      : ST_EcSlaveState;
END_VAR
```

| 名称     | 类型              | 描述                                                      |
|--------|-----------------|---------------------------------------------------------|
| bBusy  | BOOL            | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                            |
| bError | BOOL            | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                    |
| nErrId | UDINT           | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。               |
| state  | ST_EcSlaveState | 包含 EtherCAT 状态和从站的链接状态的结构体。（类型：ST_EcSlaveState [► 113]） |



在 ST 中实施的示例：

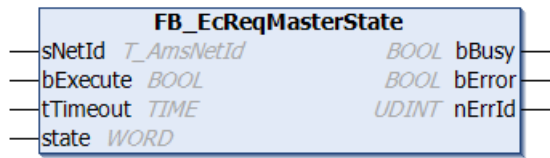
```
PROGRAM TEST_GetSlaveState
VAR
  fbGetSlaveState : FB_EcGetSlaveState;
  sNetId          : T_AmsNetId := '172.16.2.131.2.1';
  bExecute        : BOOL;
  state           : ST_EcSlaveState;
  nSlaveAddr      : UINT := 1001;
  bError          : BOOL;
  nErrId          : UDINT;
END_VAR

fbGetSlaveState(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute);
state := fbGetSlaveState.state;
bError := fbGetSlaveState.bError;
nErrId := fbGetSlaveState.nErrId;
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

5.4 FB\_EcReqMasterState



通过该功能块可以请求和设置主站设备的 EtherCAT 状态。所请求的 EtherCAT 状态会通过状态变量传输。功能块请求 EtherCAT 状态之后，便会立即变为无效状态。与功能块 FB\_EcSetMasterState 不同，它不会等待设置新状态。

另请参见：FB\_EcSetMasterState [► 44]

🔧 输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
  state       : WORD;
END_VAR
```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |
| state    | WORD       | 主站请求的 EtherCAT 状态。（请参见状态）                        |

状态

主站请求的 EtherCAT 状态。可能的状态值为：

| 常量                     | 值    | 描述               |
|------------------------|------|------------------|
| EC_DEVICE_STATE_INIT   | 0x01 | 主站请求 Init（初始化）状态 |
| EC_DEVICE_STATE_PREOP  | 0x02 | 主站请求准备运行状态       |
| EC_DEVICE_STATE_SAFEOP | 0x04 | 主站请求安全运行状态       |
| EC_DEVICE_STATE_OP     | 0x08 | 主站请求运行状态         |

## 输出

```
VAR_OUTPUT
  bBusy   : BOOL;
  bError  : BOOL;
  nErrId  : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

## 在 ST 中实施的示例：

```
PROGRAM TEST_ReqMasterState
VAR
  fbReqMasterState : FB_EcReqMasterState;
  sNetId           : T_AmsNetId:= '172.16.2.131.2.1';
  bExecute         : BOOL;
  state            : WORD :=EC_DEVICE_STATE_INIT;
  bError           : BOOL;
  nErrId           : UDINT;
END_VAR

fbReqMasterState(sNetId:= sNetId, bExecute:=bExecute, state:=state);
bError := fbReqMasterState.bError;
nErrId := fbReqMasterState.nErrId;
```

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

# 5.5 FB\_EcReqSlaveState



通过该功能块可将从站设置为指定的 EtherCAT 状态。所请求的 EtherCAT 状态会通过状态变量传输。发送更改状态的命令之后，功能块便会立即变为无效状态。与功能块 FB\_EcSetSlaveState 不同，它不会等到 EtherCAT 从站进入新状态。

另请参见：FB\_EcSetSlaveState [► 45]

## 输入

```
VAR_INPUT
  sNetId       : T_AmsNetId;
  nSlaveAddr   : UDINT;
  bExecute     : BOOL;
  tTimeout     : TIME := DEFAULT_ADS_TIMEOUT;
  state        : WORD;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 要设置其 EtherCAT 状态的 EtherCAT 从站的固定地址。              |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |
| state      | WORD       | 主站请求的 EtherCAT 状态。<br>(请参见状态)                    |

State

从站要设置为的 EtherCAT 状态。可能的状态值为：

| 常量                        | 值    | 描述                                                                                                                     |
|---------------------------|------|------------------------------------------------------------------------------------------------------------------------|
| EC_DEVICE_STATE_INIT      | 0x01 | 将从站设置为 Init（初始化）状态                                                                                                     |
| EC_DEVICE_STATE_PREOP     | 0x02 | 将从站设置为准备运行状态                                                                                                           |
| EC_DEVICE_STATE_BOOTSTRAP | 0x03 | 将从站设置为 Bootstrap 状态。该状态用于固件下载。                                                                                         |
| EC_DEVICE_STATE_SAFEOP    | 0x04 | 将从站设置为安全运行状态                                                                                                           |
| EC_DEVICE_STATE_OP        | 0x08 | 将从站设置为运行状态                                                                                                             |
| EC_DEVICE_STATE_ERROR     | 0x10 | 如果已在 EtherCAT 从站中设置状态字节中的错误位<br>(state.deviceState & EC_DEVICE_STATE_ERROR = TRUE)，则可通过设置 EC_DEVICE_STATE_ERROR 重置错误位。 |

输出

```
VAR_OUTPUT
    bBusy    : BOOL;
    bError   : BOOL;
    nErrId   : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

在 ST 中实施的示例：

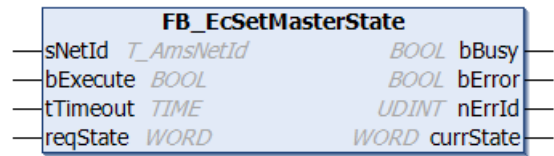
```
PROGRAM TEST_ReqSlaveState
VAR
    fbGetSlaveState : FB_EcReqSlaveState;
    sNetId          : T_AmsNetId := '172.16.2.131.2.1';
    bExecute        : BOOL;
    state           : WORD := EC_DEVICE_STATE_INIT;
    nSlaveAddr      : UINT := 1001;
    bError          : BOOL;
    nErrId          : UDINT;
END_VAR

fbGetSlaveState(sNetId:= sNetId, nSlaveAddr:= nSlaveAddr, bExecute:=bExecute, state:=state);
bError := fbGetSlaveState.bError;
nErrId := fbGetSlaveState.nErrId;
```

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

# 5.6 FB\_EcSetMasterState



通过该功能块可以请求和设置主站设备的 EtherCAT 状态。请求的 EtherCAT 状态通过 reqState 变量进行设置。该功能块请求 EtherCAT 状态，与功能块 FB\_EcReqMasterState 不同，该功能块一直处于激活状态，直到设置新状态或超出最长时间 tTimeout。当前状态在 currState 变量中输出。

另请参见：FB\_EcReqMasterState [► 41]

## 输入

```
VAR_INPUT
    sNetId    : T_AmsNetId;
    bExecute  : BOOL;
    tTimeout  : TIME := T#10s;
    reqState  : WORD;
END_VAR
```

| 名称       | 类型         | 描述                                               |
|----------|------------|--------------------------------------------------|
| sNetId   | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| bExecute | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout | TIME       | 允许功能块执行的最长时间。                                    |
| reqState | WORD       | （请参见 reqState）                                   |

## reqState

主站请求的 EtherCAT 状态。reqState 的可能值为：

| 常量                     | 值    | 描述               |
|------------------------|------|------------------|
| EC_DEVICE_STATE_INIT   | 0x01 | 主站请求 Init（初始化）状态 |
| EC_DEVICE_STATE_PREOP  | 0x02 | 主站请求准备运行状态       |
| EC_DEVICE_STATE_SAFEOP | 0x04 | 主站请求安全运行状态       |
| EC_DEVICE_STATE_OP     | 0x08 | 主站请求运行状态         |

## 输出

```
VAR_OUTPUT
    bBusy     : BOOL;
    bError    : BOOL;
    nErrId    : UDINT;
    currState : WORD;
END_VAR
```

| 名称        | 类型    | 描述                                        |
|-----------|-------|-------------------------------------------|
| bBusy     | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError    | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId    | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| currState | WORD  | 主站当前的 EtherCAT 状态                         |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

5.7 FB\_EcSetSlaveState



通过该功能块可将从站设置为指定的 EtherCAT 状态。请求的 EtherCAT 状态通过 reqState 变量进行设置。该功能块会发送更改状态的命令，与功能块 FB\_EcRegSlaveState 不同，该功能块在 EtherCAT 从站达到新状态或超出最长时间 tTimeout 之前一直处于激活状态。当前状态在 currState 变量中输出。

另请参见：FB\_EcReqSlaveState [► 42]

🔧 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := T#10s;
    reqState    : WORD;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 要设置其 EtherCAT 状态的 EtherCAT 从站的固定地址。              |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |
| reqState   | WORD       | 从站要设置为的 EtherCAT 状态。（请参见 reqState）               |

reqState

从站要设置为的 EtherCAT 状态。reqState 的可能值为：

| 常量                        | 值    | 描述                                                                                                                       |
|---------------------------|------|--------------------------------------------------------------------------------------------------------------------------|
| EC_DEVICE_STATE_INIT      | 0x01 | 将从站设置为 Init（初始化）状态                                                                                                       |
| EC_DEVICE_STATE_PREOP     | 0x02 | 将从站设置为准备运行状态                                                                                                             |
| EC_DEVICE_STATE_BOOTSTRAP | 0x03 | 将从站设置为 Bootstrap 状态。该状态用于固件下载。                                                                                           |
| EC_DEVICE_STATE_SAFEOP    | 0x04 | 将从站设置为安全运行状态                                                                                                             |
| EC_DEVICE_STATE_OP        | 0x08 | 将从站设置为运行状态                                                                                                               |
| EC_DEVICE_STATE_ERROR     | 0x10 | 如果已在 EtherCAT 从站中设置状态字节中的错误位（currState.deviceState AND EC_DEVICE_STATE_ERROR = TRUE），则可通过设置 EC_DEVICE_STATE_ERROR 重置错误位。 |

 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    currState  : ST_EcSlaveState;
END_VAR
```

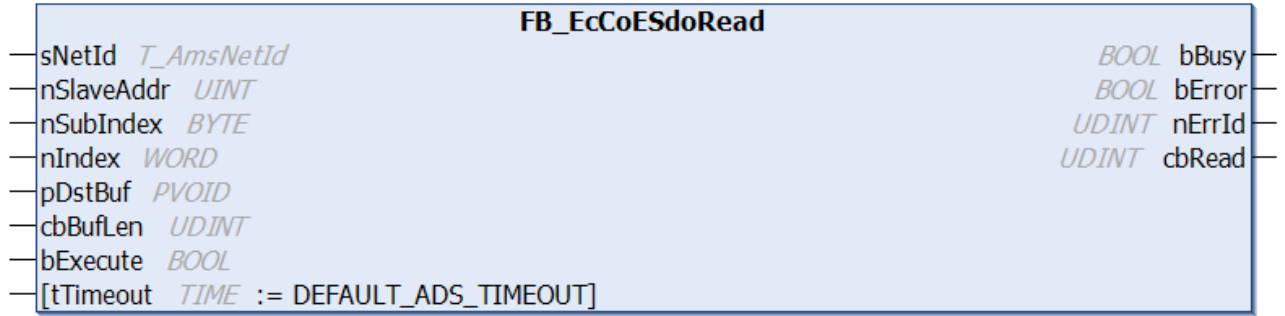
| 名称        | 类型               | 描述                                        |
|-----------|------------------|-------------------------------------------|
| bBusy     | BOOL             | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError    | BOOL             | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId    | UDINT            | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| currState | ST_EcSlave State | 从站当前的 <a href="#">EtherCAT 状态</a> [► 113] |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

6 CoE 接口

6.1 FB\_EcCoeSdoRead



功能块 FB\_EcCoeSdoRead 可以通过 SDO 通道（服务数据对象）从 EtherCAT 从站的对象字典读取数据。这就要求从站设有一个邮箱并支持“CANopen over EtherCAT”（CoE）协议。nSubIndex 和 nIndex 参数允许选择要读取的对象。必须使用功能块 FB\_EcCoeSdoReadEx [► 48] 访问完整参数，包括子元素。

输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    nSubIndex   : BYTE;
    nIndex      : WORD;
    pDstBuf     : PVOID;
    cbBufLen    : UDINT;
    bExecute     : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 应向其发送 SDO 上传命令的 EtherCAT 从站的固定地址。                |
| nSubIndex  | BYTE       | 要读取的对象的子索引。                                      |
| nIndex     | WORD       | 要读取的对象的索引。                                       |
| pDstBuf    | PVOID      | 接收缓冲器的地址（指针）。                                    |
| cbBufLen   | UDINT      | 要读取的数据的最大可用缓冲器大小（以字节为单位）。                        |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    cbRead     : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时设置该输出，并保持设置状态，直至收到反馈。               |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| cbRead | UDINT | 成功读取的数据字节数                                |

在 ST 中实施的示例：

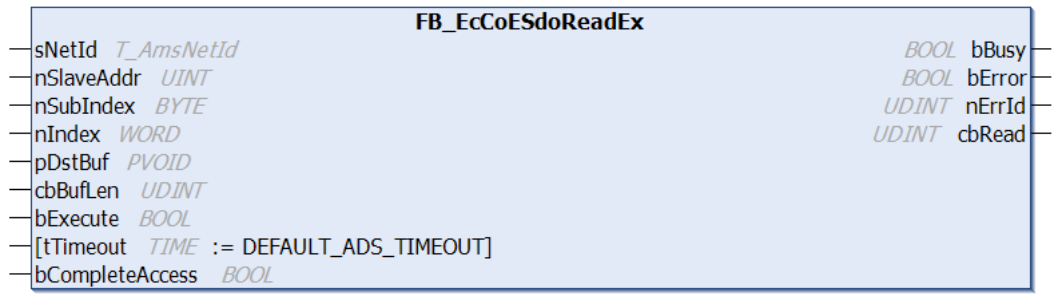
```
PROGRAM TEST_SdoRead
VAR
  fbSdoRead : FB_EcCoESdoRead;
  sNetId    : T_AmsNetId := '172.16.2.131.2.1';
  bExecute  : BOOL;
  nSlaveAddr : UINT := 1006;
  nIndex    : WORD := 16#1018;
  nSubIndex : BYTE := 1;
  vendorId  : UDINT;
  bError    : BOOL;
  nErrId    : UDINT;
END_VAR

fbSdoRead(sNetId:= sNetId,nSlaveAddr :=nSlaveAddr, nIndex:=nIndex, nSubIndex :=nSubIndex, pDstBuf:=
ADR(vendorId), cbBufLen:=SIZEOF(vendorId),bExecute:=bExecute);
bError:=fbSdoRead.bError;
nErrId:=fbSdoRead.nErrId;
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

6.2 FB\_EcCoeSdoReadEx



功能块 FB\_EcCoeSdoReadEx 可以通过 SDO 通道（服务数据对象）从 EtherCAT 从站的对象字典读取数据。这就要求从站设有一个邮箱并支持“CANopen over EtherCAT”（CoE）协议。nSubIndex 和 nIndex 参数允许选择要读取的对象。通过 bCompleteAccess := TRUE，可以读取带子元素的参数。

输入

```
VAR_INPUT
  sNetId      : T_AmsNetId; (* AmsNetId of the EtherCAT master device.*)
  nSlaveAddr  : UINT; (* Address of the slave device.*)
  nSubIndex   : BYTE; (* CANopen Sdo subindex.*)
  nIndex      : WORD; (* CANopen Sdo index.*)
  pDstBuf     : PVOID; (* Contains the address of the buffer for the received data. *)
  cbBufLen    : UDINT; (* Contains the max. number of bytes to be received. *)
  bExecute    : BOOL; (* Function block execution is triggered by a rising edge at this input. *)
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
  bCompleteAccess : BOOL; (* access complete object*)
END_VAR
```



| 名称              | 类型         | 描述                                               |
|-----------------|------------|--------------------------------------------------|
| sNetId          | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr      | UINT       | 应向其发送 SDO 上传命令的 EtherCAT 从站的固定地址。                |
| nSubIndex       | BYTE       | 要读取的对象的子索引。                                      |
| nIndex          | WORD       | 要读取的对象的索引。                                       |
| pDstBuf         | PVOID      | 接收缓冲器的地址（指针）                                     |
| cbBufLen        | UDINT      | 要读取的数据的最大可用缓冲器大小（以字节为单位）                         |
| bExecute        | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout        | TIME       | 允许功能块执行的最长时间。                                    |
| bCompleteAccess | BOOL       | 如果已设置 bCompleteAccess，则可以通过一次访问读取整个参数。           |

输出

```
VAR_OUTPUT
  bBusy   : BOOL;
  bError  : BOOL;
  nErrId  : UDINT;
  cbRead  : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时设置该输出，并保持设置状态，直至收到反馈。               |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| cbRead | UDINT | 成功读取的数据字节数                                |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

6.3 FB\_EcCoeSdoWrite



FB\_EcCoeSdoWrite 功能块允许利用 SDO 下载写入 EtherCAT 从站对象字典中的对象。这就要求从站设有一个邮箱并支持“CANopen over EtherCAT” (CoE) 协议。nSubIndex 和 nIndex 参数允许选择要写入的对象。必须使用功能块 [FB\\_EcCoeSdoWriteEx](#) [\[► 51\]](#) 访问完整参数，包括子元素。

输入

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  nSlaveAddr  : UDINT;
  nSubIndex   : BYTE;
  nIndex      : WORD;
  pSrcBuf     : PVOID;
  cbBufLen    : UDINT;
END_VAR
```

```

    bExecute      : BOOL;
    tTimeout      : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 应向其发送 SDO 下载命令的 EtherCAT 从站的固定地址。                |
| nSubIndex  | BYTE       | 要写入的对象的子索引。                                      |
| nIndex     | WORD       | 要写入的对象的索引。                                       |
| pSrcBuf    | PVOID      | 传输缓冲区的地址（指针）                                     |
| cbBufLen   | UDINT      | 以字节为单位的要发送的数据                                    |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

### 输出

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR

```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

### 在 ST 中实施的示例：

```
PROGRAM TEST_SdoWrite
```

```

VAR
    fbSdoWrite : FB_EcCoESdoWrite;
    sNetId      : T_AmsNetId := '172.16.2.131.2.1'; (* NetId of EtherCAT Master *)
    nSlaveAddr  : UINT := 1005; (* Port Number of EtherCAT Slave *)
    nIndex      : WORD := 16#4062; (* CoE Object Index *)
    nSubIndex   : BYTE := 1; (* Subindex of CoE Object *)
    nValue      : UINT := 2; (* variable to be written to the CoE Object *)
    bExecute    : BOOL; (* rising edge starts writing to the CoE Object *)
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR

```

```

fbSdoWrite(
    sNetId      := sNetId,
    nSlaveAddr  := nSlaveAddr,
    nIndex      := nIndex,
    nSubIndex   := nSubIndex,
    pSrcBuf     := ADR(nValue),
    cbBufLen    := SIZEOF(nValue),
    bExecute    := bExecute
);

```

```

IF NOT fbSdoWrite.bBusy THEN
    bExecute := FALSE;
    IF NOT fbSdoWrite.bError THEN
        (* write successful *)
        bError := FALSE;
        nErrId := 0;
    ELSE
        (* write failed *)
        bError := fbSdoWrite.bError;
        nErrId := fbSdoWrite.nErrId;
    END_IF
    fbSdoWrite(bExecute := FALSE);
END_IF

```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

6.4 FB\_EcCoeSdoWriteEx



FB\_EcCoeSdoWriteEx 功能块允许利用 SDO 下载写入EtherCAT 从站对象字典中的对象。这就要求从站设有一个邮箱并支持“CANopen over EtherCAT” (CoE) 协议。nSubIndex 和 nIndex 参数允许选择要写入的对象。通过 bCompleteAccess := TRUE，可以用子元素写入参数。

输入

```
VAR_INPUT
    sNetId      : T_AmsNetId; (* AmsNetId of the EtherCAT master device.*)
    nSlaveAddr  : UINT; (* Address of the slave device.*)
    nSubIndex   : BYTE; (* CANopen Sdo subindex.*)
    nIndex      : WORD; (* CANopen Sdo index.*)
    pSrcBuf     : PVOID; (* Contains the address of the buffer containing the data to be send. *)
    cbBufLen    : UDINT; (* Contains the max. number of bytes to be received. *)
    bExecute    : BOOL; (* Function block execution is triggered by a rising edge at this input. *)
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
    bCompleteAccess : BOOL; (* access complete object*)
END_VAR
```

| 名称              | 类型         | 描述                                               |
|-----------------|------------|--------------------------------------------------|
| sNetId          | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr      | UINT       | 应向其发送 SDO 下载命令的 EtherCAT 从站的固定地址。                |
| nSubIndex       | BYTE       | 要写入的对象的子索引。                                      |
| nIndex          | WORD       | 要写入的对象的索引。                                       |
| pSrcBuf         | PVOID      | 传输缓冲区的地址（指针）                                     |
| cbBufLen        | UDINT      | 以字节为单位的要发送的数据                                    |
| bExecute        | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout        | TIME       | 允许功能块执行的最长时间。                                    |
| bCompleteAccess | BOOL       | 如果已设置 bCompleteAccess，则可在一次访问中写入整个参数。            |

输出

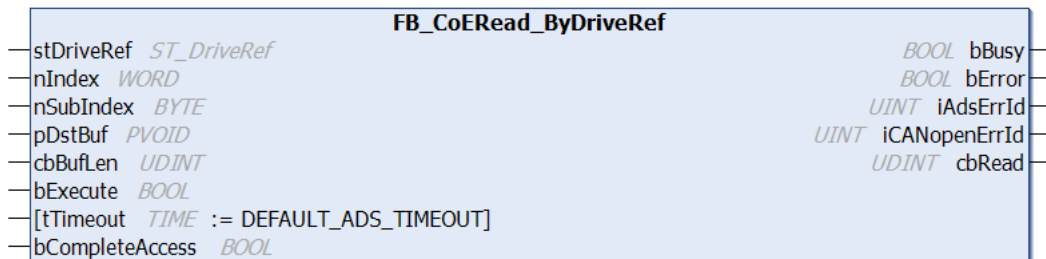
```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 6.5 FB\_CoERead\_ByDriveRef



功能块 FB\_CoERead\_ByDriveRef 可用于通过“CANopen over EtherCAT (CoE)”协议读取驱动器参数。这就要求从站设有一个邮箱并支持“CANopen over EtherCAT” (CoE) 协议。nSubIndex 和 nIndex 参数允许选择要读取的对象。通过 bCompleteAccess := TRUE，可以读取带子元素的参数。

#### 输入

```

VAR_INPUT
    stDriveRef      : ST_DriveRef; (*Contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo
of EcDrive, either preset or read from NC*)
    nIndex          : WORD; (*SoE IDN: e.g. "S_0_IDN+1" for S-0-0001 or "P_0_IDN+23" for
P-0-0023*)
    nSubIndex       : BYTE;
    pDstBuf         : PVOID; (*Contains the address of the buffer for the received data*)
    cbBufLen        : UDINT; (*Contains the max. number of bytes to be received*)
    bExecute        : BOOL; (*Function block execution is triggered by a rising edge at this
input*)
    tTimeout        : TIME; (*States the time before the function is cancelled*)
    bCompleteAccess : BOOL;
END_VAR

```

| 名称               | 类型          | 描述                                                                                                             |
|------------------|-------------|----------------------------------------------------------------------------------------------------------------|
| stDriveRef       | ST_DriveRef | 包含 EtherCAT 主站设备 AMS 网络 ID 和从站设备地址的结构。对驱动器的引用可直接链接到系统管理器中的 PLC。为此，必须使用 ST_PlCDriveRef 实例，并将字节数组的 NetID 转换为字符串。 |
| nIndex           | WORD        | 要读取的对象的索引。                                                                                                     |
| nSubIndex        | BYTE        | 要读取的对象的子索引。                                                                                                    |
| pDstBuf          | PVOID       | 接收缓冲器的地址（指针）。                                                                                                  |
| cbBufLen         | UDINT       | 要读取的数据的最大可用缓冲器大小（以字节为单位）                                                                                       |
| bExecute         | BOOL        | 功能块由该输入的上升沿激活。                                                                                                 |
| tTimeout         | TIME        | 允许功能块执行的最长时间。                                                                                                  |
| bComplete Access | BOOL        | 如果已设置 bCompleteAccess，则可以通过一次访问读取整个参数。                                                                         |

#### 输出

```

VAR_OUTPUT
    bBusy          : BOOL;
    bError         : BOOL;
    iAdsErrId      : UINT;
    iCANopenErrId  : UINT;
    cbRead         : UDINT;
END_VAR

```

| 名称            | 类型    | 描述                                   |
|---------------|-------|--------------------------------------|
| bBusy         | BOOL  | 激活功能块时设置该输出，并保持设置状态，直至收到反馈。          |
| bError        | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。 |
| iAdsErrId     | UINT  | 当已设置 bError 输出时，返回最后执行命令的 ADS 错误代码。  |
| iCANopenErrId | UINT  | 如果已设置 bError 输出，则返回 CANopen 的错误代码。   |
| cbRead        | UDINT | 成功读取的数据字节数                           |

### 在 ST 中实施的示例：

```

PROGRAM MAIN
VAR
    fbCoERead      : FB_CoERead_ByDriveRef;
    stDriveRef     : ST_DriveRef;
    nIndex         : WORD := 16#1018;
    nSubIndex      : BYTE := 1;
    bExecute       : BOOL := TRUE;
    tTimeout       : TIME := T#5S;
    bCompleteAccess : BOOL := TRUE;
    vendorId       : UDINT;
    bError         : BOOL;
    nAdsErrId      : UDINT;
    nCANopenErrId  : UDINT;
END_VAR

fbCoERead(
    stDriveRef:= stDriveRef,
    nIndex:= nIndex,
    nSubIndex:= nSubIndex,
    pDstBuf:= ADR(vendorId),
    cbBufLen:= SIZEOF(vendorId),
    bExecute:= bExecute,
    tTimeout:= tTimeout,
    bCompleteAccess:= bCompleteAccess,
);

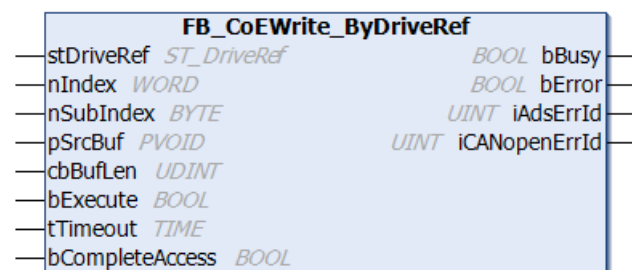
IF NOT fbCoERead.bBusy THEN
    bError:=fbCoERead.bError;
    nAdsErrId:=fbCoERead.iAdsErrId;
    nCANopenErrId:=fbCoERead.iCANopenErrId;
    bExecute := FALSE;
    fbCoERead(bExecute := bExecute);
END_IF

```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 6.6 FB\_CoEWrite\_ByDriveRef



功能块 FB\_CoEWrite\_ByDriveRef 可用于写入基于“CANopen over EtherCAT (CoE)”协议的驱动器参数。这就要求从站设有一个邮箱并支持“CANopen over EtherCAT” (CoE) 协议。nSubIndex 和 nIndex 参数允许选择要写入的对象。通过 bCompleteAccess := TRUE，可以用子元素写入参数。

### 输入

```

VAR_INPUT
    stDriveRef      : ST_DriveRef; (*Contains sNetID of EcMaster, nSlaveAddr EcDrive, nDriveNo of EcDrive, either preset or read from NC*)
    nIndex          : WORD; (*SoE IDN: e.g. "S_0_IDN+1" for S-0-0001 or "P_0_IDN+23" for P-0-0023*)
    nSubIndex       : BYTE; (*SoE element*)

```

```

    pSrcBuf      : PVOID; (*Contains the address of the buffer containing the data to be sent*)
    cbBufLen     : UDINT; (*Contains the max. number of bytes to be received*)
    bExecute     : BOOL; (*Function block execution is triggered by a rising edge at this
input*)
    tTimeout     : TIME; (*States the time before the function is cancelled*)
    bCompleteAccess : BOOL;
END_VAR

```

| 名称               | 类型          | 描述                                                                                                             |
|------------------|-------------|----------------------------------------------------------------------------------------------------------------|
| stDriveRef       | ST_DriveRef | 包含 EtherCAT 主站设备 AMS 网络 ID 和从站设备地址的结构。对驱动器的引用可直接链接到系统管理器中的 PLC。为此，必须使用 ST_PlcDriveRef 实例，并将字节数组的 NetID 转换为字符串。 |
| nIndex           | WORD        | 要写入的对象的索引。                                                                                                     |
| nSubIndex        | BYTE        | 要写入的对象的子索引。                                                                                                    |
| pSrcBuf          | PVOID       | 传输缓冲区的地址（指针）                                                                                                   |
| cbBufLen         | UDINT       | 要发送的数据的最大可用缓冲器大小（以字节为单位）。                                                                                      |
| bExecute         | BOOL        | 功能块由该输入的上升沿激活。                                                                                                 |
| tTimeout         | TIME        | 允许功能块执行的最长时间。                                                                                                  |
| bComplete Access | BOOL        | 如果已设置 bCompleteAccess，则可以通过一次访问读取整个参数。                                                                         |

### 输出

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    iAdsErrId  : UINT;
    iCANopenErrId : UINT;
END_VAR

```

| 名称            | 类型   | 描述                                   |
|---------------|------|--------------------------------------|
| bBusy         | BOOL | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。         |
| bError        | BOOL | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。 |
| iAdsErrId     | UINT | 当已设置 bError 输出时，返回最后执行命令的 ADS 错误代码。  |
| iCANopenErrId | UINT | 如果已设置 bError 输出，则返回 CANopen 的错误代码。   |

### 在 ST 中实施的示例：

```

PROGRAM MAIN
VAR
    fbCoEWrite      : FB CoEWrite_ByDriveRef;
    stDriveRef      : ST_DriveRef;
    nIndex          : WORD := 16#1018;
    nSubIndex       : BYTE := 1;
    bExecute        : BOOL := TRUE;
    tTimeout        : TIME := T#5S;
    bCompleteAccess : BOOL := TRUE;
    vendorId        : UDINT := 2;
    bError          : BOOL;
    nAdsErrId       : UDINT;
    nCANopenErrId   : UDINT;
END_VAR

fbCoEWrite(
    stDriveRef:= stDriveRef,
    nIndex:= nIndex,
    nSubIndex:= nSubIndex,
    pSrcBuf:= ADR(vendorId),
    cbBufLen:= SIZEOF(vendorId),
    bExecute:= bExecute,
    tTimeout:= tTimeout,
    bCompleteAccess:= bCompleteAccess,
);

IF NOT fbCoEWrite.bBusy THEN
    bError:= fbCoEWrite.bError;
    nAdsErrId:= fbCoEWrite.iAdsErrId;
    nCANopenErrId:= fbCoEWrite.iCANopenErrId;
    bExecute := FALSE;
    fbCoEWrite(bExecute := bExecute);
END_IF

```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

6.7 FB\_EcCoeReadBIC



功能块 FB\_EcCoeReadBIC 可用于通过 SDO（服务数据对象）访问从 EtherCAT 从站的对象字典中读取 BIC。为此，从站必须设有一个邮箱并支持“CANopen over EtherCAT (CoE)”协议，并且对象目录必须包含一个带有 BIC 的对象 0x10E2:01。

输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 应向其发送 SDO 上传命令的 EtherCAT 从站的固定地址。                |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    sBICValue  : STRING
    stMSID     : ST_SplittedBIC
END_VAR
```

| 名称        | 类型             | 描述                                                                                                                                                                                           |
|-----------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bBusy     | BOOL           | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                                                                                                                                                                 |
| bError    | BOOL           | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                                                                                                                                                         |
| nErrId    | UDINT          | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。                                                                                                                                                    |
| sBICValue | STRING(1023)   | 该输出包含（零错误运行后）重置 bBusy 输出后 EtherCAT 从站的 BIC，例如 “„1P193995SBTN0002agdw1KEL7411 Q1 2P112104020018””。                                                                                            |
| stMSID    | ST_SplittedBIC | 该输出包含（零错误运行后）重置 bBusy 输出后 EtherCAT 从站的 BIC 子字符串。<br>为上述 BIC 分配了以下子字符串：<br>sItemNo = "193995"<br>sBTN = "0002agdw"<br>sDescription = "EL7411"<br>sQuantity = "1"<br>sBatchNo = "112104020018" |

# ST 中的实施示例

```

PROGRAM TEST_EcCoEReadBIC
VAR
    fbCoEBIC      : FB_EcCoEReadBIC;
    sNetId        : T_AmsNetId := '172.16.2.131.2.1';
    bExecute      : BOOL := TRUE;
    nSlaveAddr    : UINT := 1006;
    sCoEBIC       : STRING(1023);
    stCoEBIC      : ST_SplittedBIC;
    bError        : BOOL;
    nErrId        : UDINT;
END_VAR

fbCoEBIC(sNetId:= sNetId, nSlaveAddr:= nPort, bExecute:= bExecute, tTimeout:= T#5s);
IF NOT fbCoEBIC.bBusy THEN
    bExecute := FALSE;
    IF NOT fbCoEBIC.bError THEN
        stCoEBIC := fbCoEBIC.stMSID;
        sCoEBIC := fbCoEBIC.sBICValue;
    END_IF
    fbCoEBIC(bExecute:= bExecute);
END_IF
    
```

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 6.8 FB\_EcCoeReadBTN



功能块 FB\_EcCoeReadBTN 可用于通过 SDO（服务数据对象）访问从 EtherCAT 从站的对象字典中读取 BTN。为此，从站必须设有一个邮箱并支持“CANopen over EtherCAT (CoE)”协议，并且对象字典必须包含一个带有 BTN 的对象 0xF083。

### 输入

```

VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
    
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 应向其发送 SDO 上传命令的 EtherCAT 从站的固定地址。                |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

### 输出

```

VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    sBTN       : STRING(9)
END_VAR
    
```



| 名称     | 类型     | 描述                                                       |
|--------|--------|----------------------------------------------------------|
| bBusy  | BOOL   | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                             |
| bError | BOOL   | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                     |
| nErrId | UDINT  | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。                |
| sBTN   | STRING | 该输出包含（零错误运行后）重置 bBusy 输出后 EtherCAT 从站的 BTN，例如“0002agdw”。 |

ST 中的实施示例

```
PROGRAM TEST_ EcCoEReadBtn
VAR
    fbCoEBTN      : FB_EcCoEReadBtn;
    sNetId         : T_AmsNetId := '172.16.2.131.2.1';
    bExecute       : BOOL := TRUE;
    nSlaveAddr     : UINT := 1006;
    sCoEBTN       : STRING;
    bError         : BOOL;
    nErrId         : UDINT;
END_VAR

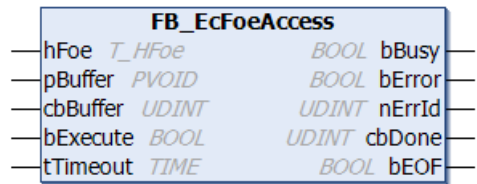
fbCoEBTN(sNetId:= sNetID, nSlaveAddr:= nPort, bExecute:= bExecute, tTimeout:= T#5S);
IF NOT fbCoEBTN.bBusy THEN
    bExecute := FALSE;
    IF NOT fbCoEBTN.bError THEN
        sCoEBTN := fbCoEBTN.sBTN;
    END_IF
    fbCoEBTN(bExecute:= bExecute);
END_IF
```

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

# 7 FoE 接口

## 7.1 FB\_EcFoeAccess



该功能块通过“File access over EtherCAT”邮箱协议的通信端口写入或读取数据。

### 输入

```
VAR_INPUT
    hFoe      : T_HFoe;
    pBuffer   : DWORD;
    cbBuffer  : UDINT;
    bExecute  : BOOL;
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型                | 描述                                                                            |
|----------|-------------------|-------------------------------------------------------------------------------|
| hFoe     | T_HFoe<br>[► 118] | “File access over EtherCAT” 路径                                                |
| pBuffer  | DWORD             | 包含要读入数据的缓冲器的地址（读取访问）或包含要写入数据的缓冲器的地址（写入访问）。缓冲器可以是单个变量、数组或结构体，其地址可以用 ADR 运算符找到。 |
| cbBuffer | UDINT             | 包含要写入或读取的数据字节数。                                                               |
| bExecute | BOOL              | 功能块由该输入的上升沿激活。                                                                |
| tTimeout | TIME              | 允许功能块执行的最长时间。                                                                 |

### 输出

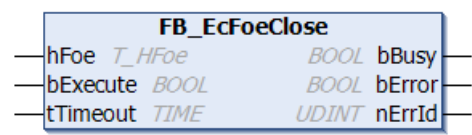
```
VAR_OUTPUT
    bBusy     : BOOL;
    bError    : BOOL;
    nErrId    : UDINT;
    cbDone    : UDINT;
    bEOF      : BOOL;
END_VAR
```

| 名称     | 类型    | 描述                                                      |
|--------|-------|---------------------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。                            |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。                    |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。               |
| cbDone | UDINT | 最近成功写入或读取的数据字节数                                         |
| bEOF   | BOOL  | 文件末尾，如果在读取访问过程中到达文件末尾，则该变量会变为 TRUE（真）。对于写入访问，该变量没有任何作用。 |

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

7.2 FB\_EcFoeClose



该功能块会关闭通过 “File access over EtherCAT” 邮箱协议的通信端口。

输入

```
VAR_INPUT
    hFoe      : T_HFoe;
    bExecute   : BOOL;
    tTimeout   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称       | 类型                | 描述                             |
|----------|-------------------|--------------------------------|
| hFoe     | T_HFoe<br>[► 118] | “File access over EtherCAT” 路径 |
| bExecute | BOOL              | 功能块由该输入的上升沿激活。                 |
| tTimeout | TIME              | 允许功能块执行的最长时间。                  |

输出

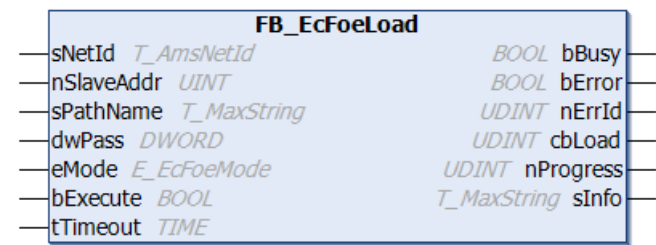
```
VAR_OUTPUT
    bBusy      : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

7.3 FB\_EcFoeLoad



功能块 FB\_EcFoeLoad 可用于通过 “File access over EtherCAT” 邮箱协议向 EtherCAT 设备下载或从中上传文件。

**i** 文件路径只能指向计算机上的本地文件系统。这意味着在这里不能使用网络路径。要通过 FoE 协议上传或下载文件，功能块会自动将 EtherCAT 设备重置为 BOOTSTRAP 模式。最后，功能块会尝试将设备复位到重置为原始状态。

 输入

```

VAR_INPUT
    sNetId      : T_AmsNetId ;
    nSlaveAddr  : UINT;
    sPathName   : T_MaxString;
    dwPass      : DWORD := 0;
    eMode       : E_EcFoeMode := eFoeMode_Write;
    bExecute    : BOOL;
    tTimeout    : TIME := T#200s;
END_VAR

```

| 名称         | 类型                     | 描述                                                                            |
|------------|------------------------|-------------------------------------------------------------------------------|
| sNetId     | T_AmsNetId             | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。                                             |
| nSlaveAddr | UINT                   | 要上传或下载文件的 EtherCAT 从站的固定地址。                                                   |
| sPathName  | T_MaxString            | 包含要写入或读取的文件的路径和文件名。<br>(例如: 'C:\FOE_Test\EL6751\ECATFW__EL6751_C6_V0030.efw') |
| dwPass     | DWORD                  | 密码 (默认: 0)                                                                    |
| eMode      | E_EcFoeMode<br>[► 109] | “通过 File access over EtherCAT” 访问模式<br>(默认: 写入访问)                             |
| bExecute   | BOOL                   | 功能块由该输入的上升沿激活。                                                                |
| tTimeout   | TIME                   | 执行功能块时不得超过的最长时间 (默认: 200 秒)。                                                  |

 输出

```

VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbLoad      : UDINT;
    nProgress    : UDINT;
    sInfo       : T_MaxString;
END_VAR

```

| 名称        | 类型          | 描述                                                  |
|-----------|-------------|-----------------------------------------------------|
| bBusy     | BOOL        | 激活功能块时会设置该输出, 并保持设置状态, 直至收到反馈。                      |
| bError    | BOOL        | 在 bBusy 输出完成重置后, 在命令传输过程中出现错误时, 设置该输出。              |
| nErrId    | UDINT       | 如果已设置 bError 输出, 则会提供与最近执行的命令相关的 ADS 错误代码。          |
| cbLoad    | UDINT       | 成功写入或读取的数据字节数                                       |
| nProgress | UDINT       | 写入访问进度 (范围: 0 – 100%)。该变量目前不用于读取访问, 在读取情况下, 它始终为 0。 |
| sInfo     | T_MaxString | 字符串形式的附加命令信息 (保留)                                   |

## ST 中的示例:

bLoad 变量的上升沿通过 “File access over EtherCAT” 邮箱协议触发下载固件。

```

PROGRAM MAIN
VAR
    fbDownload : FB_EcFoeLoad := (
        sNetID      := '5.0.34.38.3.1',
        nSlaveAddr  := 1004,
        sPathName   := 'C:\FOE_Test\EL6751\ECATFW__EL6751_C6_V0030.efw',
        dwPass      := 0,
        eMode       := eFoeMode_Write );
    bLoad          : BOOL;
    bBusy          : BOOL;
    bError         : BOOL;
    nErrID         : UDINT;
    nBytesWritten  : UDINT;
    nPercent       : UDINT;
END_VAR

fbDownload( bExecute := bLoad,
            bBusy => bBusy,
            bError => bError,
            nErrId => nErrID,
            cbLoad => nBytesWritten,
            nProgress => nPercent );

```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

7.4 FB\_EcFoeOpen



该功能块用于为“File access over EtherCAT” 邮箱协议打开通信端口。

输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nPort       : UINT;
    sPathName   : T_MaxString;
    dwPass      : DWORD;
    eMode       : E_EcFoeMode;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称        | 类型                     | 描述                                                                  |
|-----------|------------------------|---------------------------------------------------------------------|
| sNetId    | T_AmsNetId             | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。                                   |
| nPort     | UINT                   | EtherCAT 设备的固定地址                                                    |
| sPathName | T_MaxString            | 文件路径名（例如：'c:\TwinCAT\FOE\Data.fwp'）<br>（有关 sPathName 的进一步解释，请参见下文。） |
| dwPass    | DWORD                  | 密码                                                                  |
| eMode     | E_EcFoeMode<br>[► 109] | 访问模式（写入/读取访问）                                                       |
| bExecute  | BOOL                   | 功能块由该输入的上升沿激活。                                                      |
| tTimeout  | TIME                   | 允许功能块执行的最长时间。                                                       |

sPathName

默认情况下，只会从输入的文件路径（例如 'c:\TwinCAT\FOE\Data.fwp'）中提取文件名（不含文件扩展名），并将其用作 FoE 协议的文件名（在我们的示例中为：“数据”）。从版本 3.3.12.0 的库开始，也可以使用包含文件扩展名的文件名（在我们的示例中为：'Data.fwp'）。

通过全局布尔变量

Tc2\_EtherCAT.bEcFoeOpenFileNameWithFileExt  
，可以在 FB\_EcFoeOpen 功能块的所有实例中启用或停用文件扩展名。默认情况下，变量的值为 FALSE（假）（无文件扩展名）。如果将该值设置为 TRUE（真），则会启用使用文件扩展名。  
注意，FoE 功能块最初用于不使用文件扩展名的固件更新。如果要更新固件，必须确保全局变量拥有原始默认值，即 FALSE（假）。

输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
    hFoe       : T_HFoe;
END_VAR
```

| 名称     | 类型             | 描述                                        |
|--------|----------------|-------------------------------------------|
| bBusy  | BOOL           | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL           | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT          | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| hFoe   | T_HFoe [► 118] | “File access over EtherCAT” 路径            |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

7.5 FB\_EcFoeReadFile



FB\_EcFoeReadFile 功能块可用于通过“File access over EtherCAT” 邮箱协议将文件从 EtherCAT 设备下载到本地数据载体。



文件路径只能指向计算机上的本地文件系统。这意味着在这里不能使用网络路径。

🔧 输入

```
VAR_INPUT
    sFSrvNetId      : T_AmsNetId := '';
    sFSrvPathName   : T_MaxString;
    sEcNetId        : T_AmsNetId;
    nSlaveAddr      : UINT;
    sFoEPathName    : T_MaxString;
    dwPass          : DWORD := 0;
    bExecute        : BOOL;
    tTimeout        : TIME := T#200s;
END_VAR
```

| 名称            | 类型          | 描述                                        |
|---------------|-------------|-------------------------------------------|
| sFSrvNetId    | T_AmsNetId  | 要写入已读取文件的计算机的 AMS 网络 ID。（默认：本地计算机）        |
| sFSrvPathName | T_MaxString | 包含要写入文件的路径和文件名（例如 'C:\Data\LogData.csv'）。 |
| sEcNetId      | T_AmsNetId  | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。         |
| nSlaveAddr    | UINT        | EtherCAT 从站的地址                            |
| sFoEPathName  | T_MaxString | EtherCAT 从站上的文件名（例如 'LogData'）            |
| dwPass        | DWORD       | 密码                                        |
| bExecute      | BOOL        | 功能块由该输入的上升沿激活。                            |
| tTimeout      | TIME        | 允许功能块执行的最长时间。（默认：200 秒。）                  |

输出

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
  cbRead     : UDINT;
  sInfo      : T_MaxString;
END_VAR
```

| 名称     | 类型          | 描述                                        |
|--------|-------------|-------------------------------------------|
| bBusy  | BOOL        | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL        | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT       | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| cbRead | UDINT       | 成功读取的数据字节数                                |
| sInfo  | T_MaxString | 附加 FoE 错误信息（保留）                           |

ST 中的示例：

bExecute 变量的上升沿会触发通过“File access over EtherCAT”邮箱协议读取指定文件。选定的 EtherCAT 从站 (sEcNetId & nSlaveAddr) 会读取 sFoEPathName 中命名的文件。文件将以 sFSrvPathName 中指定的名称存储在选定的计算机 (sFSrvNetID) 中。如果读取 EtherCAT 从站中的文件需要密码，可通过 dwPass 进行指定。

直到 bBusy 切换为 FALSE（假），读写操作才算完成。只有这样，才能评估读取错误信息或读取的字节数。

```
PROGRAM MAIN
VAR
  fbEcReadFile : FB_EcFoeReadFile := (
    sFSrvNetID      := '5.0.34.38.1.1', (* NetID for target file *)
    sFSrvPathName   := 'C:\Data\LogData.csv', (* Pathname for target file *)
    sEcNetId        := '5.0.34.38.3.1', (* NetID of EtherCAT master *)
    nSlaveAddr      := 1004, (* EtherCAT slave address *)
    sFoEPathName    := 'LogData', (* Name of source file *)
    dwPass          := 0
  );
  bExecute      : BOOL := TRUE;
  bBusy         : BOOL;
  bError        : BOOL;
  nErrID        : UDINT;
  nBytesRead    : UDINT;
END_VAR

fbEcReadFile (
  bExecute := bExecute,
  bBusy => bBusy,
  bError => bError,
  nErrID => nErrID
);
IF NOT bBusy THEN
  bExecute := FALSE;

  IF NOT bError THEN
    (* done, no error *)
    nBytesRead := fbEcReadFile.cbRead;
  ELSE
    (* evaluate error *)
    nBytesRead := 0;
  END_IF
  fbEcReadFile (bExecute := FALSE);
END_IF
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库              |
|----------------|------------------------|------------------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT >= 3.3.14 |

7.6 FB\_EcFoeWriteFile



功能块 FB\_EcFoeWriteFile 可用于通过“File access over EtherCAT” 邮箱协议将文件从本地数据载体写入 EtherCAT 设备。

**i** 文件路径只能指向计算机上的本地文件系统。这意味着在这里不能使用网络路径。

输入

```
VAR_INPUT
  sFSrvNetId      : T_AmsNetId := '';
  sFSrvPathName   : T_MaxString;
  sEcNetId        : T_AmsNetId;
  nSlaveAddr      : UINT;
  sFoEPathName    : T_MaxString;
  dwPass          : DWORD := 0;
  bExecute        : BOOL;
  tTimeout        : TIME := T#200s;
END_VAR
```

| 名称            | 类型          | 描述                                        |
|---------------|-------------|-------------------------------------------|
| sFSrvNetId    | T_AmsNetId  | 要读取要写入文件的计算机的 AMS 网络 ID。（默认：本地计算机）        |
| sFSrvPathName | T_MaxString | 包含要读取文件的路径和文件名（例如：'C:\Data\LogData.csv'）。 |
| sEcNetId      | T_AmsNetId  | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。         |
| nSlaveAddr    | UINT        | EtherCAT 从站的地址                            |
| sFoEPathName  | T_MaxString | EtherCAT 从站上的文件名（例如 'LogData'）            |
| dwPass        | DWORD       | 密码                                        |
| bExecute      | BOOL        | 功能块由该输入的上升沿激活。                            |
| tTimeout      | TIME        | 允许功能块执行的最长时间。（默认：200 秒。）                  |

输出

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
  cbWritten  : UDINT;
  nProgress  : UDINT;
  sInfo      : T_MaxString;
END_VAR
```



| 名称        | 类型          | 描述                                        |
|-----------|-------------|-------------------------------------------|
| bBusy     | BOOL        | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError    | BOOL        | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId    | UDINT       | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| cbWritten | UDINT       | 成功写入的数据字节数                                |
| nProgress | UDINT       | 写入访问进度（范围：0 – 100%）。                      |
| sInfo     | T_MaxString | 附加 FoE 错误信息（保留）                           |

### ST 中的示例：

bExecute 变量的上升沿通过“File access over EtherCAT”邮箱协议触发写入指定文件。从选定的计算机 (sFSrvNetID) 中读取 sFSrvPathName 中命名的文件。文件以 sFoEPathName 中指定的名称存储在选定的 EtherCAT 从站 (sEcNetId & nSlaveAddr) 中。如果需要密码才能将文件写入 EtherCAT 从站，可通过 dwPass 进行设定。

直到 bBusy 切换为 FALSE（假），读写操作才算完成。只有这样，才能评估读取错误信息或读取的字节数。

```

PROGRAM MAIN
VAR
    fbEcWriteFile : FB_EcFoeWriteFile := (
        sFSrvNetID      := '5.0.34.38.1.1', (* NetID for source file *)
        sFSrvPathName   := 'C:\Data\LogData.csv', (* Pathname for source file *)
        sEcNetId        := '5.0.34.38.3.1', (* NetID of EtherCAT master *)
        nSlaveAddr      := 1004, (* EtherCAT slave address *)
        sFoEPathName    := 'LogData', (* Name of target file *)
        dwPass          := 0
    );
    bExecute : BOOL := TRUE;
    bBusy    : BOOL;
    bError   : BOOL;
    nErrID   : UDINT;
    nBytesWritten : UDINT;
END_VAR

fbEcWriteFile (
    bExecute := bExecute,
    bBusy => bBusy,
    bError => bError,
    nErrId => nErrID
);
IF NOT bBusy THEN
    bExecute := FALSE;

    IF NOT bError THEN
        (* done, no error *)
        nBytesWritten := fbEcWriteFile.cbWritten;
    ELSE
        (* evaluate error *)
        nBytesWritten := 0;
    END IF
    fbEcWriteFile (bExecute := FALSE);
END_IF

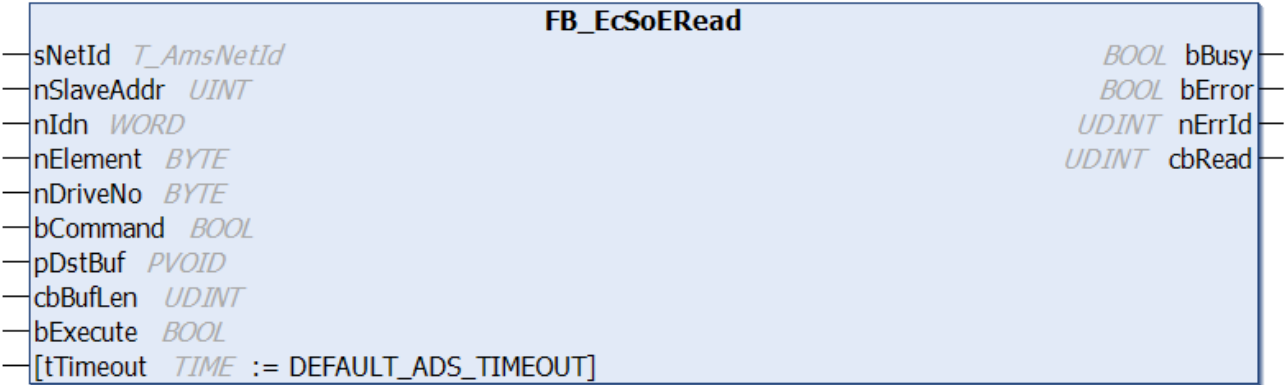
```

### 前提条件

| 开发环境                 | 目标平台                   | 需要的 PLC 库               |
|----------------------|------------------------|-------------------------|
| TwinCAT v3.1.4024.56 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT >= 3.5.1.0 |

## 8 SoE 接口

### 8.1 FB\_EcSoeRead



功能块 FB\_EcSoeRead 可用于通过“Sercos over EtherCAT (SoE)”协议读取驱动器参数。为此，从站必须有一个邮箱并支持 SoE 协议。要读取的驱动器参数由参数 nIdn（标识号）、nElement 和 nDriveNo 指定。

#### 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    nIdn        : WORD;
    nElement    : BYTE;
    nDriveNo    : BYTE;
    bCommand    : BOOL;
    pDstBuf     : PVOID;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 要向其发送 SoE 读取命令的 EtherCAT 从站的固定地址。                |
| nIdn       | WORD       | 要读取的参数的标识号                                       |
| nElement   | BYTE       | 要读取的参数的元素编号<br>(请参见 nElement)                    |
| nDriveNo   | BYTE       | 驱动器编号                                            |
| bCommand   | BOOL       | 如果要使用内部命令执行，则应设置此参数。                             |
| pDstBuf    | PVOID      | 接收缓冲器的地址（指针）                                     |
| cbBufLen   | UDINT      | 要读取的数据的最大可用缓冲器大小（以字节为单位）                         |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

#### nElement

要读取的参数的元素编号。允许使用以下值：

| 值    | 描述     |
|------|--------|
| 0x01 | 数据状态   |
| 0x02 | 名称（只读） |
| 0x04 | 属性     |
| 0x08 | 单位     |
| 0x10 | 最小     |
| 0x20 | 最大     |
| 0x40 | 值      |
| 0x80 | 默认值    |

输出

```
VAR_OUTPUT
  bBusy   : BOOL;
  bError  : BOOL;
  nErrId  : UDINT;
  cbRead  : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时设置该输出，并保持设置状态，直至收到反馈。               |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| cbRead | UDINT | 成功读取的数据字节数                                |

在 ST 中实施的示例：

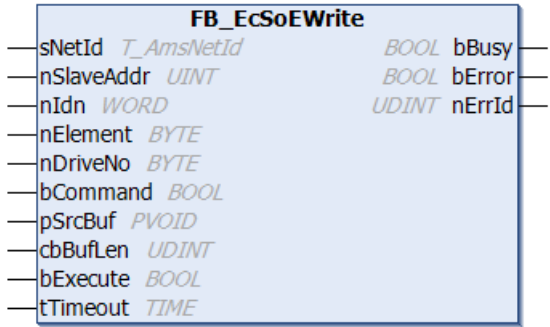
```
PROGRAM TEST_SoERead
VAR
  fbSoERead : FB_EcSoERead;
  sNetId    : T_AmsNetId := '172.16.2.131.2.1';
  bExecute  : BOOL;
  nSlaveAddr : UINT := 1006;
  nIdn      : WORD := 15;
  nElement  : BYTE := 0;
  nDriveNo  : BYTE := 0;
  bCommand  : BOOL := FALSE;
  val       : UDINT;
  bError    : BOOL;
  nErrId    : UDINT;
END_VAR

fbSoERead(sNetId:= sNetId,nSlaveAddr :=nSlaveAddr, nIdn := nIdn, nElement:=nElement, nDriveNo := nDriveNo, bCommand:=bCommand, pDstBuf:= ADR(val), cbBufLen:=SIZEOF(val),bExecute:=bExecute);
bError := fbSoERead.bError;
nErrId := fbSoERead.nErrId;
```

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

8.2 FB\_EcSoeWrite



功能块 FB\_EcSoeWrite 可用于通过 “Sercos over EtherCAT (SoE)” 协议写入驱动器参数。为此，从站必须有一个邮箱并支持 SoE 协议。要写入的驱动器参数由参数 nIdn（标识号）、nElement 和 nDriveNo 指定。

### 输入

```
VAR_INPUT
    sNetId      : T_AmsNetId;
    nSlaveAddr  : UINT;
    nIdn        : WORD;
    nElement    : BYTE;
    nDriveNo    : BYTE;
    pCommand    : BOOL;
    pSrcBuf     : PVOID;
    cbBufLen    : UDINT;
    bExecute    : BOOL;
    tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

| 名称         | 类型         | 描述                                               |
|------------|------------|--------------------------------------------------|
| sNetId     | T_AmsNetId | 包含 EtherCAT 主站设备的 AMS 网络 ID 的字符串。（类型：T_AmsNetId） |
| nSlaveAddr | UINT       | 要向其发送 SoE 写入命令的 EtherCAT 从站的固定地址。                |
| nIdn       | WORD       | 要写入的参数的标识号。                                      |
| nElement   | BYTE       | 要写入的参数的元素编号<br>(请参见 nElement)                    |
| nDriveNo   | BYTE       | 驱动器编号                                            |
| bCommand   | BOOL       | 如果要使用内部命令执行，则应设置此参数。                             |
| pSrcBuf    | PVOID      | 传输缓冲区的地址（指针）                                     |
| cbBufLen   | UDINT      | 以字节为单位的要发送的数据                                    |
| bExecute   | BOOL       | 功能块由该输入的上升沿激活。                                   |
| tTimeout   | TIME       | 允许功能块执行的最长时间。                                    |

### nElement

要写入的参数的元素编号。允许使用以下值：

| 值    | 描述     |
|------|--------|
| 0x01 | 数据状态   |
| 0x02 | 名称（只读） |
| 0x04 | 属性     |
| 0x08 | 单位     |
| 0x10 | 最小     |
| 0x20 | 最大     |
| 0x40 | 值      |
| 0x80 | 默认值    |

### 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrId     : UDINT;
END_VAR
```

| 名称     | 类型    | 描述                                        |
|--------|-------|-------------------------------------------|
| bBusy  | BOOL  | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| nErrId | UDINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |

在 ST 中实施的示例：

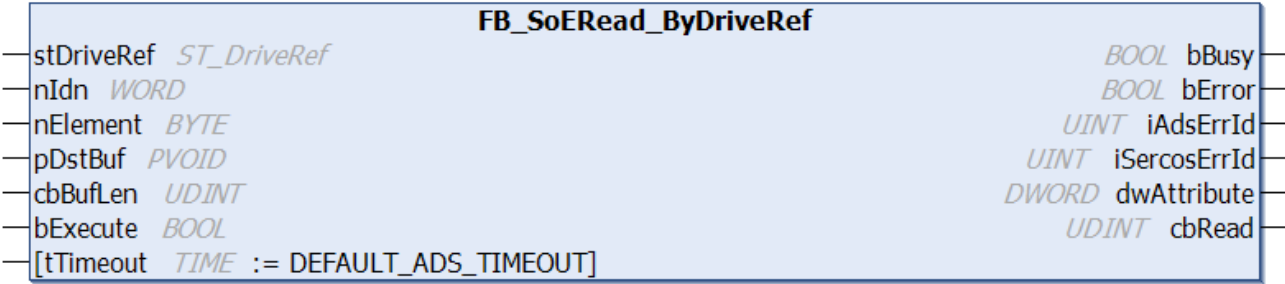
```
PROGRAM TEST_SoEWrite
VAR
    fbSoeWrite : FB_EcSoEWrite;
    sNetId      : T_AmsNetId:= '172.16.2.131.2.1';
    bExecute    : BOOL;
    nSlaveAddr  : UINT := 1006;
    nIdn        : WORD := 15;
    nElement    : BYTE := 0;
    nDriveNo    : BYTE := 0;
    bCommand    : BOOL := FALSE;
    val         : UINT;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR

fbSoEWrite(sNetId:= sNetId,nSlaveAddr :=nSlaveAddr, nIdn := nIdn, nElement:=nElement, nDriveNo := nDriveNo,bCommand:=bCommand, pSrcBuf:= ADR(val), cbBufLen:=SIZEOF(val),bExecute:=bExecute);
bError := fbSoEWrite.bError;
nErrId := fbSoEWrite.nErrId;
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

8.3 FB\_SoERead\_ByDriveRef



FB\_SoeRead\_ByRef 功能块可用于通过“Sercos over EtherCAT (SoE)”协议读取驱动器参数。为此，从站必须有一个邮箱并支持 SoE 协议。要读取的驱动器参数由参数 nIdn（标识号）、nElement 和 stDriveRef 指定。

Tc2\_EtherCAT 库中的全局变量 bSeqReadDrvAttrAndValue := TRUE 可用于强制执行对属性和值的循序访问。该变量的默认值为 FALSE（假）。AX5xxx 系列的设备可以并行和循序访问属性和数值。对于第三方设备，可能有必要将对属性和值的访问分开，这样总体上可能访问速度减慢几个周期。

📁 输入

```
VAR_INPUT
    stDriveRef : ST_DriveRef; (* contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo of EcDrive, either preset or read from NC *)
    nIdn       : WORD; (* SoE IDN: e.g. "S_0-IDN + 1" for S-0-0001 or "P_0-IDN + 23" for P-0-0023*)
    nElement   : BYTE; (* SoE element.*)
    pDstBuf    : PVOID; (* Contains the address of the buffer for the received data. *)
    cbBufLen   : UDINT; (* Contains the max. number of bytes to be received. *)
    bExecute   : BOOL; (* Function block execution is triggered by a rising edge at this input. *)
    tTimeout   : TIME := DEFAULT_ADS_TIMEOUT; (* States the time before the function is cancelled. *)
END_VAR
```

| 名称         | 类型          | 描述                                                                       |
|------------|-------------|--------------------------------------------------------------------------|
| stDriveRef | ST_DriveRef | 对驱动器的引用可直接链接到系统管理器中的 PLC。为此，必须使用 ST_PlcDriveRef 实例，并将字节数组的 NetID 转换为字符串。 |
| nIdn       | WORD        | 要读取的参数的标识号                                                               |
| nElement   | BYTE        | 要读取的参数的元素编号<br>(请参见 nElement)                                            |
| pDstBuf    | PVOID       | 读取缓冲器的地址 (指针)                                                            |
| cbBufLen   | UDINT       | 要读取的数据的最大可用缓冲器大小 (以字节为单位)                                                |
| bExecute   | BOOL        | 功能块由该输入的上升沿激活。                                                           |
| tTimeout   | TIME        | 允许功能块执行的最长时间。                                                            |

## nElement

要读取的参数的元素编号。允许使用以下值：

| 值    | 描述      |
|------|---------|
| 0x01 | 数据状态    |
| 0x02 | 名称 (只读) |
| 0x04 | 属性      |
| 0x08 | 单位      |
| 0x10 | 最小      |
| 0x20 | 最大      |
| 0x40 | 值       |
| 0x80 | 默认值     |

## 输出

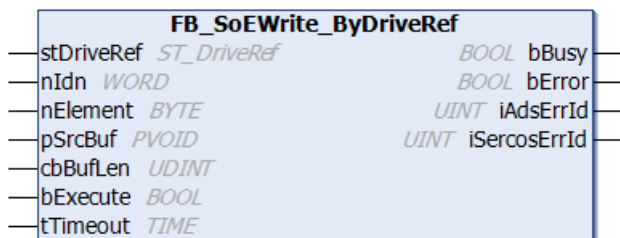
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iAdsErrId  : UINT;
  iSercosErrId : UINT;
  dwAttribute : DWORD;
  cbRead     : UDINT;
END_VAR
```

| 名称           | 类型    | 描述                                     |
|--------------|-------|----------------------------------------|
| bBusy        | BOOL  | 激活功能块时设置该输出，并保持设置状态，直至收到反馈。            |
| bError       | BOOL  | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。   |
| iAdsErrId    | UINT  | 当已设置 bError 输出时，返回最后执行命令的 ADS 错误代码。    |
| iSercosErrId | UINT  | 在设置 bError 输出的情况下，返回最后执行命令的 Sercos 错误。 |
| dwAttribute  | DWORD | 返回 Sercos 参数的属性。                       |
| cbRead       | UDINT | 成功读取的数据字节数                             |

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 8.4 FB\_SoEWrite\_ByDriveRef



功能块 `FB_SoeWrite_ByRef` 可用于通过“Sercos over EtherCAT (SoE)”协议写入驱动器参数。为此，从站必须有一个邮箱并支持 SoE 协议。要写入的驱动器参数由参数 `nIdn`（标识号）、`nElement` 和 `stDriveRef` 指定。

`Tc2_EtherCAT` 库中的全局变量 `bSeqReadDrvAttrAndValue := TRUE` 可用于强制执行对属性和值的循序访问。该变量的默认值为 `FALSE`（假）。AX5xxx 系列设备支持对属性和值的并行和顺序访问。对于第三方设备，可能需要将属性和值的分开访问，这样总体上可能访问速度减慢几个周期。

### 输入

```

VAR_INPUT
    stDriveRef : ST_DriveRef; (* contains sNetID of EcMaster, nSlaveAddr of EcDrive, nDriveNo of EcDrive, either preset or read from NC *)
    nIdn       : WORD; (* SoE IDN: e.g. "S_0_IDN + 1" for S-0-0001 or "P_0_IDN + 23" for P-0-0023 *)
    nElement   : BYTE; (* SoE element. *)
    pSrcBuf    : PVOID; (* Contains the address of the buffer containing the data to be send. *)
    cbBufLen   : UDINT; (* Contains the max. number of bytes to be received. *)
    bExecute   : BOOL; (* Function block execution is triggered by a rising edge at this input. *)
    tTimeout   : TIME := DEFAULT_ADS_TIMEOUT; (* States the time before the function is cancelled. *)
END_VAR

```

| 名称         | 类型          | 描述                                                                                                 |
|------------|-------------|----------------------------------------------------------------------------------------------------|
| stDriveRef | ST_DriveRef | 对驱动器的引用可直接链接到系统管理器中的 PLC。为此，必须使用 <code>ST_PlcDriveRef</code> 实例，并将字节数组的 <code>NetID</code> 转换为字符串。 |
| nIdn       | WORD        | 要读取的参数的标识号                                                                                         |
| nElement   | BYTE        | 要读取的参数的元素编号<br>(请参见 <code>nElement</code> )                                                        |
| pSrcBuf    |             | 发送缓冲器的地址（指针）                                                                                       |
| cbBufLen   |             | 要读取的数据的最大可用缓冲器大小（以字节为单位）                                                                           |
| bExecute   | BOOL        | 功能块由该输入的上升沿激活。                                                                                     |
| tTimeout   | TIME        | 允许功能块执行的最长时间。                                                                                      |

### nElement

要读取的参数的元素编号。允许使用以下值：

| 值    | 描述     |
|------|--------|
| 0x01 | 数据状态   |
| 0x02 | 名称（只读） |
| 0x04 | 属性     |
| 0x08 | 单位     |
| 0x10 | 最小     |
| 0x20 | 最大     |
| 0x40 | 值      |
| 0x80 | 默认值    |

 输出

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    iAdsErrId  : UINT;
    iSercosErrId : UINT;
END_VAR
```

| 名称           | 类型   | 描述                                        |
|--------------|------|-------------------------------------------|
| bBusy        | BOOL | 激活功能块时会设置该输出，并保持设置状态，直至收到反馈。              |
| bError       | BOOL | 在 bBusy 输出完成重置后，在命令传输过程中出现错误时，设置该输出。      |
| iAdsErrId    | UINT | 如果已设置 bError 输出，则会提供与最近执行的命令相关的 ADS 错误代码。 |
| iSercosErrId | UINT | 当已设置 bError 输出时，会返回最后执行命令的 Sercos 错误。     |

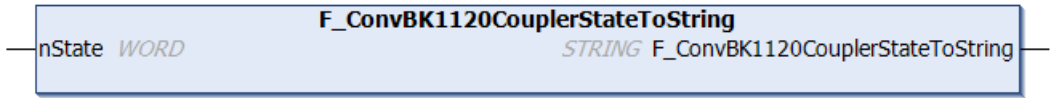
前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |



9 转换功能

9.1 F\_ConvBK1120CouplerStateToString



功能 F\_ConvBK1120CouplerStateToString 会以字符串形式返回 BK1120/BK1150/BK1250 的耦合器状态。nState = 0 时，返回“无错误”，否则返回“K-bus 错误”，如 nState = 1。如果有多个错误待处理，则用逗号分隔。

🔧 输入

```
VAR_INPUT
    nState : WORD;
END_VAR
```

| 名称     | 类型   | 描述                                                                                                                                                                                                                                                                             |
|--------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nState | WORD | 耦合器状态；可在系统管理器中将 BK1120/BK1250 的输入链接到 PLC。<br>0x0000 = 'No error'<br>0x0001 = 'K-Bus error'<br>0x0002 = 'Configuration error'<br>0x0010 = 'Outputs disabled'<br>0x0020 = 'K-Bus overrun'<br>0x0040 = 'Communication error (Inputs)'<br>0x0080 = 'Communication error (Outputs)' |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

9.2 F\_ConvMasterDevStateToString



功能 F\_ConvMasterDevStateToString 可将 EtherCAT 主站的设备状态转换为字符串。

nState = 0 时，返回“确定”，否则返回“不确定 – 链接错误”，例如 nState = 1。如果有多个错误待处理，则用连字符分隔。

🔧 输入

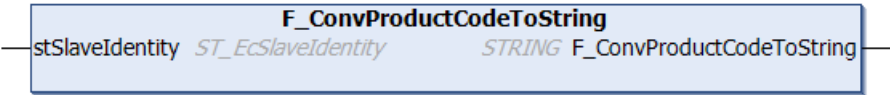
```
VAR_INPUT
    nState : WORD;
END_VAR
```

| 名称     | 类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nState | WORD | <p>EtherCAT 主站的设备状态；可在系统管理器中作为 DevState 从 EtherCAT 主站的输入链接到 PLC。</p> <p>0x0001 = 'Link error'<br/>                     0x0002 = 'I/O locked after link error (I/O reset required)'<br/>                     0x0004 = 'Link error (redundancy adapter)'<br/>                     0x0008 = 'Missing one frame (redundancy mode)'<br/>                     0x0010 = 'Out of send resources (I/O reset required)'<br/>                     0x0020 = 'Watchdog triggered'<br/>                     0x0040 = 'Ethernet driver (miniport) not found'<br/>                     0x0080 = 'I/O reset active'<br/>                     0x0100 = 'At least one device in 'INIT' state'<br/>                     0x0200 = 'At least one device in 'PRE-OP' state'<br/>                     0x0400 = 'At least one device in 'SAFE-OP' state'<br/>                     0x0800 = 'At least one device indicates an error state'<br/>                     0x1000 = 'DC not in sync'</p> |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 9.3 F\_ConvProductCodeToString



功能 F\_ConvProductToString 会以字符串形式返回产品代码，如 'EL6731-0000-0017'。从版本 3.3.8.0 的 Tc2\_EtherCAT 库开始，该功能还支持 ELM 和 EPP 系列从站，如 'EPP4374-0002-0018' 和 'ELM3704-0001-0016'。

输入

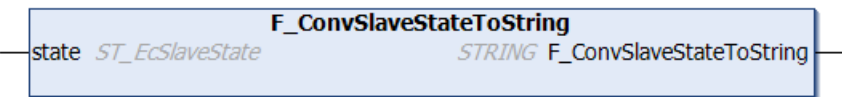
```
VAR_INPUT
    stSlaveIdentity : ST_EcSlaveIdentity;
END_VAR
```

| 名称              | 类型                 | 描述                                                        |
|-----------------|--------------------|-----------------------------------------------------------|
| stSlaveIdentity | ST_EcSlaveIdentity | 可通过 <a href="#">FB_EcGetSlaveIdentity</a> [► 29] 读取的从站标识。 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 9.4 F\_ConvSlaveStateToString



功能 F\_ConvSlaveStateToString 会以字符串形式返回 EtherCAT 从站状态。关于字符串的转换，请参见 [F\\_ConvStateToString](#) [► 76]。

输入

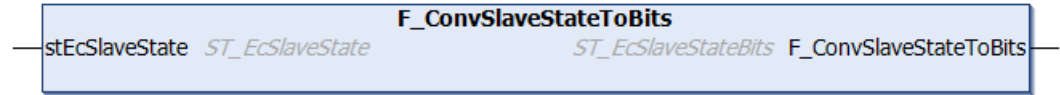
```
VAR_INPUT
    state : ST_EcSlaveState;
END_VAR
```

| 名称    | 类型              | 描述                                                       |
|-------|-----------------|----------------------------------------------------------|
| state | ST_EcSlaveState | EtherCAT 从站状态结构体（包括：deviceState: BYTE; linkState: BYTE;） |

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

9.5 F\_ConvSlaveStateToBits



功能 F\_ConvSlaveStateToBits 会以 TYPE ST\_EcSlaveStateBits [► 115] 结构体 形式返回 EtherCAT 从站状态。

🔧 输入

```
VAR_INPUT
    stEcSlaveState : ST_EcSlaveState;
END_VAR
```

| 名称             | 类型              | 描述                                                       |
|----------------|-----------------|----------------------------------------------------------|
| stEcSlaveState | ST_EcSlaveState | EtherCAT 从站状态结构体（包括：deviceState: BYTE; linkState: BYTE;） |

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

9.6 F\_ConvSlaveStateToBitsEx



功能 F\_ConvSlaveStateToBitsEx 会以 ST\_EcSlaveStateBitsEx [► 115] 结构体 形式返回 EtherCAT 从站状态。

🔧 输入

```
VAR_INPUT
    stEcSlaveState : ST_EcSlaveState;
END_VAR
```

| 名称             | 类型              | 描述                                                       |
|----------------|-----------------|----------------------------------------------------------|
| stEcSlaveState | ST_EcSlaveState | EtherCAT 从站状态结构体（包括：deviceState: BYTE; linkState: BYTE;） |

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

## 9.7 F\_ConvStateToString



功能 F\_ConvStateToString 会以字符串形式返回 EtherCAT 从站状态。nState = 0 时，返回 ‘ ’，否则返回 ‘INIT ’，例如 nState = 1。如果有多条消息待处理，则用空格隔开。

### 输入

```
VAR_INPUT
    nState : WORD;
END_VAR
```

| 名称     | 类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nState | WORD | EtherCAT 从站状态为 WORD<br>0x_1 = 'INIT'<br>0x_2 = 'PREOP'<br>0x_3 = 'BOOT'<br>0x_4 = 'SAFEOP'<br>0x_8 = 'OP'<br>0x001_ = 'Slave signals error'<br>0x002_ = 'Invalid vendorId, productCode... read'<br>0x004_ = 'Initialization error occurred'<br>0x008_ = 'Slave disabled'<br>0x010_ = 'Slave not present'<br>0x020_ = 'Slave signals link error'<br>0x040_ = 'Slave signals missing link'<br>0x080_ = 'Slave Slave signals unexpected link'<br>0x100_ = 'Communication port A'<br>0x200_ = 'Communication port B'<br>0x400_ = 'Communication port C'<br>0x800_ = 'Communication port D' |

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

# 10 分布式时钟

## 10.1 DCTIME32

### 10.1.1 ConvertDcTimeToPos



该功能块可将 T\_DCTIME32 [► 117] 类型的 32 位分布式时钟系统时间变量转换为相应的 NC 轴位置（即精确到此时的 NC 轴位置）。

#### 输入

```
VAR_INPUT
    nAxisId : UDINT;
    nSubIdx : UDINT;
    dcTime : T_DCTIME32; (* 32 bit distributed clock time *)
END_VAR
```

| 名称      | 类型         | 描述                                                                                                                                                                                                                                                                |
|---------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nAxisId | UDINT      | NC 轴的 ID                                                                                                                                                                                                                                                          |
| nSubIdx | UDINT      | 该 32 位输入变量包含两个不同的信息项，因此被分为两个 16 位值： <ul style="list-style-type: none"><li>LowWord（最低有效 16 位）包含子索引，用于在轴上对编码器子元素进行相对寻址。子索引从 0 开始向上计数。对于只有一个编码器的典型轴，空子索引是正确的。</li><li>HighWord（最高有效 16 位）包含一个控制字（位掩码），会影响计算位置的方式（如插值或外推法）。如果位掩码值为 0x0001，则表示计算中要包括轴的设定加速度。</li></ul> |
| dcTime  | T_DCTIME32 | 32 位“分布式时钟系统时间”变量。该输入变量转换为相应的 NC 轴位置。                                                                                                                                                                                                                             |

**i** 32 位时间只能在当前系统时间 ± 2,147 秒的狭窄 范围内使用，以确保其准确无误。在功能块内无法检查该输入 条件。

#### 输出

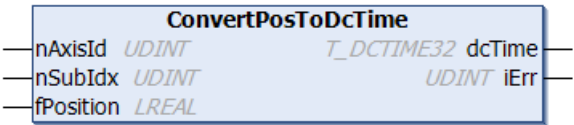
```
VAR_OUTPUT
    fPosition : LREAL;
    iErr : UDINT;
END_VAR
```

| 名称        | 类型    | 描述                                                         |
|-----------|-------|------------------------------------------------------------|
| fPosition | LREAL | 提供与 dcTime 相对应的 NC 轴位置。这是一个经过缩放并带有偏移量的 NC 轴位置，例如以度或毫米为 单位。 |
| iErr      | UDINT | 如果出现错误，则会返回错误编号，例如错误 0x4012（不允许使用轴 ID，或在系统中不存在轴）。          |

#### 前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

10.1.2 ConvertPosToDcTime



该功能块可将 NC 轴位置转换为相应的 T\_DCTIME32 [► 117] 类型的 32 位分布式时钟系统时间变量（即精确到达或即将到达该 NC 轴位置的时间）。

输入

```
VAR_INPUT
    nAxisId      : UDINT;
    nSubIdx      : UDINT;
    fPosition    : LREAL;
END_VAR
```

| 名称        | 类型    | 描述                                                                                                                                                                                                                                                                 |
|-----------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nAxisId   | UDINT | NC 轴的 ID                                                                                                                                                                                                                                                           |
| nSubIdx   | UDINT | 该 32 位输入变量 由两个不同的信息项组成，并被分为两个 16 位值： <ul style="list-style-type: none"><li>LowWord（最低有效 16 位）包含子索引，用于在轴上对编码器子元素进行相对寻址。子索引从 0 开始向上计数。对于只有一个编码器的典型轴，空子索引是正确的。</li><li>HighWord（最高有效 16 位）包含一个控制字（位掩码），会影响计算位置的方式（如插值或外推法）。如果位掩码值为 0x0001，则表示计算中要包括轴的设定加速度。</li></ul> |
| fPosition | LREAL | 转换为相应 32 位“分布式时钟系统时间”变量的 NC 轴位置。<br>如果与位置相关的“分布式时钟系统时间”超出预期时间 范围 ± 2.147 秒，则该转换会被拒绝，并显示错误编号。                                                                                                                                                                       |

输出

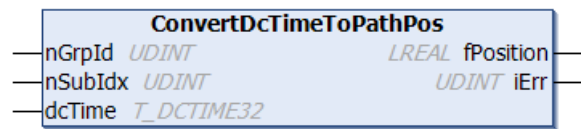
```
VAR_OUTPUT
    dcTime : T_DCTIME32; (* 32 bit distributed clock time *)
    iErr   : UDINT;
END_VAR
```

| 名称     | 类型         | 描述                                                                                                                                                                                                 |
|--------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dcTime | T_DCTIME32 | 返回与输入 fPosition 相对应的 32 位“分布式时钟系统时间”变量。                                                                                                                                                            |
| iErr   | UDINT      | 如果出现错误，则会提供错误编号，例如 <ul style="list-style-type: none"><li>错误 0x4012：不允许使用轴 ID，或在系统中不存在轴。</li><li>错误 0x4361：超出时间范围（大余最大时间）</li><li>错误 0x4362：超出时间范围（小于最小时间）</li><li>错误 0x4363：无法用数学方法确定位置。</li></ul> |

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

10.1.3 ConvertDcTimeToPathPos



该功能块可将 T\_DCTIME32 [► 117] 类型的 32 位分布式时钟系统时间变量转换为当前激活的 Nci 程序轮廓上的相对 Nci 路径距离（即该功能块会根据时序返回正或负的相对间隔）。

输入

```
VAR_INPUT
    nGrpId   : UDINT;
    nSubIdx  : UDINT;
    dcTime   : T_DCTIME32; (* 32 bit distributed clock time *)
END_VAR
```

| 名称      | 类型         | 描述                                                                                                                                                                                                                                                                                                                    |
|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nGrpId  | UDINT      | 相应 Nci 通道的组 ID                                                                                                                                                                                                                                                                                                        |
| nSubIdx | UDINT      | 该 32 位输入变量包含两个不同的信息项，因此被分为两个 16 位值： <ul style="list-style-type: none"><li>LowWord（最低有效 16 位）包含子索引，用于在轴上对编码器子元素进行相对寻址。子索引从 0 开始向上计数。对于只有一个编码器的典型轴，空子索引是正确的。</li><li>HighWord（最高有效 16 位）包含一个控制字（位掩码），会影响位置计算的方式（例如内插法或外插法）。<br/>位掩码 0x0001 表示计算中要包含的轴的设定加速度。<br/>位掩码 0x0010 表示计算是相对的，目前是强制性的。否则，调用会被拒绝，并显示错误。</li></ul> |
| dcTime  | T_DCTIME32 | 32 位“分布式时钟系统时间”变量。该输入变量被转换为轮廓上相应的相对 Nci 路径距离。                                                                                                                                                                                                                                                                         |

**i** 32 位时间只能在当前系统时间 ± 2,147 秒的狭窄 范围内使用，以确保其准确无误。在功能块内无法检查该输入 条件。

输出

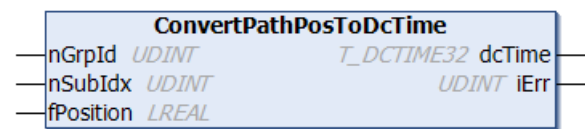
```
VAR_OUTPUT
    fPosition : LREAL;
    iErr      : UDINT;
END_VAR
```

| 名称        | 类型    | 描述                           |
|-----------|-------|------------------------------|
| fPosition | LREAL | 返回轮廓上与 dcTime 相对应的 Nci 路径距离。 |
| iErr      | UDINT | 出现错误时，会返回错误编号                |

前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

10.1.4 ConvertPathPosToDcTime



该功能块可将相对 Nci 路径距离转换为相应的 T\_DCTIME32 [► 117] 类型的 32 位分布式时钟系统时间变量（即与相对 Nci 路径距离相对应或曾经相对应的时间）。

输入

```
VAR_INPUT
    nGrpId      : UDINT;
    nSubIdx     : UDINT;
    fPosition   : LREAL;
END_VAR
```

| 名称        | 类型    | 描述                                                                                                                                                                                                                                                                                                                    |
|-----------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nGrpId    | UDINT | 相应 Nci 通道的组 ID                                                                                                                                                                                                                                                                                                        |
| nSubIdx   | UDINT | 该 32 位输入变量包含两个不同的信息项，因此被分为两个 16 位值： <ul style="list-style-type: none"><li>LowWord（最低有效 16 位）包含子索引，用于在轴上对编码器子元素进行相对寻址。子索引从 0 开始向上计数。对于只有一个编码器的典型轴，空子索引是正确的。</li><li>HighWord（最高有效 16 位）包含一个控制字（位掩码），会影响位置计算的方式（例如内插法或外插法）。<br/>位掩码 0x0001 表示计算中要包含的轴的设定加速度。<br/>位掩码 0x0010 表示计算是相对的，目前是强制性的。否则，调用会被拒绝，并显示错误。</li></ul> |
| fPosition | LREAL | 将相对 Nci 路径距离转换为相应的 32 位“分布式时钟系统时间”。<br>如果与相对 Nci 路径距离相关的“分布式时钟系统时间”超出预期时间窗口 ± 2.147 秒，则该转换会被拒绝，并显示错误编号。                                                                                                                                                                                                               |

输出

```
VAR_OUTPUT
    dcTime : T_DCTIME32; (* 32 bit distributed clock time *)
    iErr   : UDINT;
END_VAR
```

| 名称     | 类型         | 描述                                                                                                                                                         |
|--------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dcTime | T_DCTIME32 | 返回与输入 fPosition 相对应的 32 位“分布式时钟系统时间”变量。                                                                                                                    |
| iErr   | UDINT      | 如果出现错误，则会提供错误编号，例如 <ul style="list-style-type: none"><li>错误 0x4361：超出时间范围（大余最大时间）</li><li>错误 0x4362：超出时间范围（小于最小时间）</li><li>错误 0x4363：无法用数学方法确定位置</li></ul> |

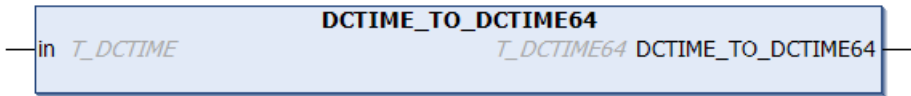
前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |



10.2 DCTIME64

10.2.1 DCTIME\_TO\_DCTIME64



该功能可将 T\_DCTIME [► 118] 类型的分布式时钟系统时间变量转换为 T\_DCTIME64 [► 117] 类型的 64 位分布式时钟系统时间变量。

功能 DCTIME\_TO\_DCTIME64: T\_DCTIME64

🔧 输入

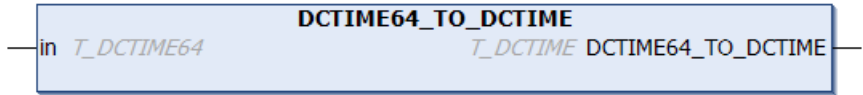
```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

| 名称 | 类型       | 描述                |
|----|----------|-------------------|
| in | T_DCTIME | 要转换的“分布式时钟系统时间”变量 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.2.2 DCTIME64\_TO\_DCTIME



该功能可将 T\_DCTIME [► 117] 类型的分布式时钟系统时间变量转换为 T\_DCTIME64 [► 118] 类型的 64 位分布式时钟系统时间变量。

功能 DCTIME64\_TO\_DCTIME: T\_DCTIME

🔧 输入

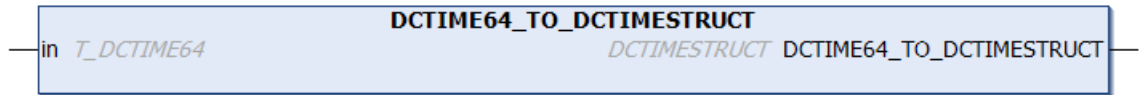
```
VAR_INPUT
    in : T_DCTIME64;
END_VAR
```

| 名称 | 类型         | 描述                |
|----|------------|-------------------|
| in | T_DCTIME64 | 要转换的“分布式时钟系统时间”变量 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.2.3 DCTIME64\_TO\_DCTIMESTRUCT



该功能可将 `T_DCTIME64` [► 117] 类型的 64 位分布式时钟系统时间变量转换为 `DCTIMESTRUCT` [► 116] 类型的结构化变量。

### 功能 `DCTIME64_TO_DCTIMESTRUCT`

#### 🔧 输入

```
VAR_INPUT
    in : T_DCTIME64;
END_VAR
```

| 名称 | 类型         | 描述                |
|----|------------|-------------------|
| in | T_DCTIME64 | 要转换的“分布式时钟系统时间”变量 |

#### 示例：

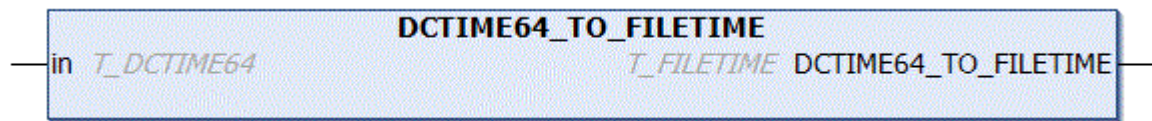
```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT;
    dcTime : T_DCTIME64;
END_VAR

dcTime := F_GetCurDcTickTime64();
dcStruct := DCTIME64_TO_DCTIMESTRUCT (dcTime);
```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.2.4 DCTIME64\_TO\_FILETIME64



该功能可将 `T_DCTIME64` [► 117] 类型的 64 位“分布式时钟系统时间”变量转换为 `T_FILETIME64` 类型的 64 位“Windows File Time”变量。

### 功能 `DCTIME64_TO_FILETIME64`: `T_FILETIME64`

#### 🔧 输入

```
VAR_INPUT
    in : T_DCTIME64;
END_VAR;
```

| 名称 | 类型         | 描述                |
|----|------------|-------------------|
| in | T_DCTIME64 | 要转换的“分布式时钟系统时间”变量 |

#### 示例：

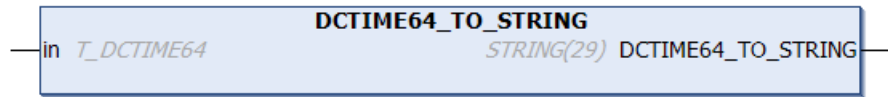
```
PROGRAM P_TEST
VAR
    ft : T_FILETIME64;
    dct : T_DCTIME64;
END_VAR

dct := F_GetCurDcTickTime64();
ft := DCTIME64_TO_FILETIME64 (dct);
```

#### 前提条件

| 开发环境              | 目标平台                   | 需要的 PLC 库                |
|-------------------|------------------------|--------------------------|
| TwinCAT v3.1.4024 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT >= 3.3.16.0 |

10.2.5 DCTIME64\_TO\_STRING



该功能可将 T\_DCTIME64 [► 117] 类型的 64 位分布式时钟系统时间变量转换为字符串。

转换后的字符串格式如下： ‘YYYY-MM-DD-hh:mm:ss.nnnnnnnnnn ‘

- YYYY: 年;
- MM: 月;
- DD: 日;
- hh: 小时;
- mm: 分钟;
- ss: 秒;
- nnnnnnnnnn: 纳秒

功能 DCTIME64\_TO\_STRING: 字符串 (29)

🔧 输入

```
VAR_INPUT
    in : T_DCTIME64; (*Distributed Clock Time*)
END_VAR
```

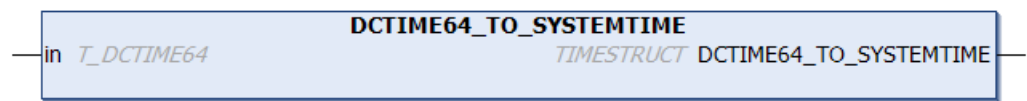
| 名称 | 类型         | 描述                |
|----|------------|-------------------|
| in | T_DCTIME64 | 要转换的“分布式时钟系统时间”变量 |

示例：  
参见功能说明： F\_GetCurDcTickTime64 [► 90].

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.2.6 DCTIME64\_TO\_SYSTEMTIME



该功能可将 T\_DCTIME64 [► 117] 类型的 64 位分布式时钟系统时间变量转换为 Timestruct 类型的结构化 Windows 系统时间变量。

DCTIME64\_TO\_SYSTEMTIME: Timestruct

🔧 输入

```
VAR_INPUT
    in : T_DCTIME64;
END_VAR
```

| 名称 | 类型         | 描述                |
|----|------------|-------------------|
| in | T_DCTIME64 | 要转换的“分布式时钟系统时间”变量 |

示例：

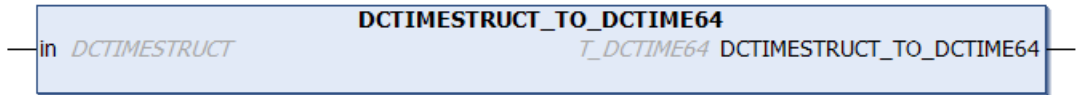
```
PROGRAM P_TEST
VAR
    syst : TIMESTRUCT
END_VAR

syst := DCTIME64_TO_SYSTEMTIME ( F_GetCurDcTickTime64() )
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.2.7 DCTIMESTRUCT\_TO\_DCTIME64



该功能可将 DCTIMESTRUCT [► 116] 类型的结构化变量转换为 64 位分布式时钟系统时间变量 T\_DCTIME64 [► 117]。在转换过程中，wDayofWeek 结构变量会被忽略。结构变量 wYear 必须大于或等于 2000，并小于 2584。对于结构变量出现无效值，功能会返回零值。

功能 DCTIMESTRUCT\_TO\_DCTIME64: T\_DCTIME64

🔧 输入

```
VAR_INPUT
    in : DCTIMESTRUCT;
END_VAR
```

| 名称 | 类型           | 描述        |
|----|--------------|-----------|
| in | DCTIMESTRUCT | 要转换的结构化变量 |

示例:

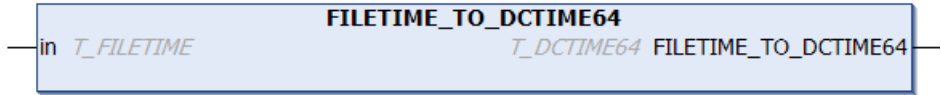
```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT := ( wYear := 2008, wMonth := 3, wDay := 13,
                                wHour := 1, wMinute := 2, wSecond := 3,
                                wMilliseconds := 123, wMicroseconds := 456, wNanoseconds := 789 );
    dc64 : T_DCTIME64;
END_VAR

dc64 := DCTIMESTRUCT_TO_DCTIME64( dcStruct );
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.2.8 FILETIME64\_TO\_DCTIME64



该功能可将 T\_FILETIME64 类型的 64 位“Windows 文件时间”变量转换为 T\_DCTIME64 [► 117] 类型的 64 位“分布式时钟系统时间”变量。如果出现转换错误，则功能会返回0。

功能 FILETIME64\_TO\_DCTIME64: T\_DCTIME64

🔧 输入

```
VAR_INPUT
    in : T_FILETIME64;
END_VAR
```

| 名称 | 类型           | 描述                         |
|----|--------------|----------------------------|
| in | T_FILETIME64 | 要转换的“Windows File Time” 变量 |

示例：

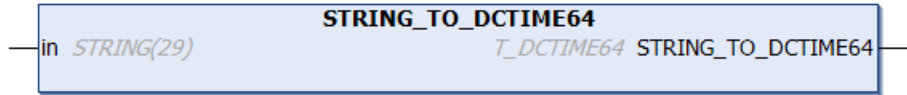
```
PROGRAM P_TEST
VAR
    ft : T_FILETIME64;
    dct : T_DCTIME64;
END_VAR

ft := F_GetSystemTime();
dct := FILETIME64_TO_DCTIME64(ft);
```

前提条件

| 开发环境              | 目标平台                   | 需要的 PLC 库                |
|-------------------|------------------------|--------------------------|
| TwinCAT v3.1.4024 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT >= 3.3.16.0 |

10.2.9      STRING\_TO\_DCTIME64



该功能可将字符串转换为 T\_DCTIME64 [► 117] 类型的分布式时钟系统时间变量。

功能 STRING\_TO\_DCTIME64: T\_DCTIME64

📌 输入

```
VAR_INPUT
    in : STRING(29);
END_VAR
```

| 名称 | 类型     | 描述                                                                                                                                                                                                                          |
|----|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | STRING | 要转换的字符串<br>字符串必须具有以下格式：'YYYY-MM-DD-hh:mm:ss:nnnnnnnnnn' <ul style="list-style-type: none"><li>• YYYY：年；</li><li>• MM：月；</li><li>• DD：日；</li><li>• hh：小时；</li><li>• mm：分钟；</li><li>• ss：秒；</li><li>• nnnnnnnnnn：纳秒</li></ul> |

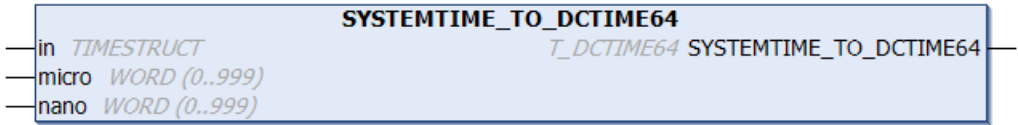
示例：

请参见功能 F\_GetCurDcTickTime64 [► 90] 的说明。

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.2.10     SYSTEMTIME\_TO\_DCTIME64



该功能可将 Timestruct 类型的结构化 Windows 系统时间变量转换为 T\_DCTIME64 [► 117] 类型的 64 位分布式时钟系统时间变量。如果出现转换错误，则功能会返回 0。

功能 SYSTEMTIME\_TO\_DCTIME64: T\_DCTIME64

🔴 输入

```
VAR_INPUT
    in      : Timestruct;
    micro   : word(0..999); (* Microseconds: 0..999 *)
    nano    : word(0..999); (* Nanoseconds: 0..999 *)
END_VAR
```

| 名称    | 类型         | 描述                          |
|-------|------------|-----------------------------|
| in    | Timestruct | 要转换的“Windows System Time”变量 |
| micro | word       | 微秒                          |
| nano  | word       | 纳秒                          |

示例:

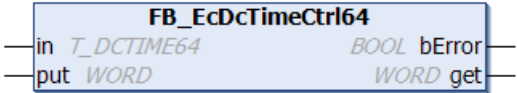
```
PROGRAM P_TEST
VAR
    syst : Timestruct := ( wYear := 2009, wMonth := 9, wDay := 16, wHour := 12, wMinute := 22, wSecond := 44, wMilliseconds := 123 );
END_VAR

dct := SYSTEMTIME_TO_DCTIME64( syst, 456, 789 );
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.2.11     FB\_EcDcTimeCtrl64



该功能块可用于读取 T\_DCTIME64 [► 117] 类型的 64 位“分布式时钟系统时间”变量中的年、月、日等各个部分。该功能块有多个 A\_GETXYZ 操作。调用所需操作之后，“get”输出变量中就会出现 XYZ 组件的值。目前未使用“put”输入变量。

该功能块具有以下任务：

- A\_GetYear
- A\_GetMonth
- A\_GetDay
- A\_GetDayOfWeek
- A\_GetHour
- A\_GetMinute
- A\_GetSecond
- A\_GetMilli
- A\_GetMicro

- A\_GetNano

### 输入

```
VAR_IN_OUT
    put : WORD;
END_VAR
```

| 名称  | 类型   | 描述          |
|-----|------|-------------|
| put | WORD | 输入参数（目前未使用） |

### 输入/输出

```
VAR_IN_OUT
    in : T_DCTIME64;
END_VAR
```

| 名称 | 类型         | 描述                |
|----|------------|-------------------|
| in | T_DCTIME64 | 要转换的“分布式时钟系统时间”变量 |

### 输出

```
VAR_IN_OUT
    bError : BOOL;
    get : WORD;
END_VAR
```

| 名称     | 类型   | 描述                     |
|--------|------|------------------------|
| bError | BOOL | 如果操作调用过程中出现错误，则会设置此输出。 |
| get    | WORD | 输出参数（年、月、日等）           |

### 在 ST 中实施的示例：

```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT;
    dcTime : T_DCTIME64;
    fbCtrl : FB_EcDcTimeCtrl;

    wYear : WORD;
    wMonth : WORD;
    wDay : WORD;
    wDayOfWeek : WORD;
    wHour : WORD;
    wMinute : WORD;
    wSecond : WORD;
    wMilli : WORD;
    wMicro : WORD;
    wNano : WORD;
END_VAR

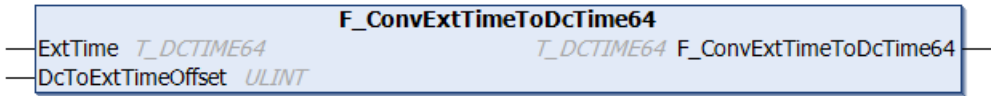
dcTime := F_GetCurDcTickTime64();
fbCtrl.A_GetYear( in := dcTime, get => wYear );
fbCtrl.A_GetMonth( in := dcTime, get => wMonth );
fbCtrl.A_GetDay( in := dcTime, get => wDay );
fbCtrl.A_GetDayOfWeek( in := dcTime, get => wDayOfWeek );
fbCtrl.A_GetHour( in := dcTime, get => wHour );
fbCtrl.A_GetMinute( in := dcTime, get => wMinute );
fbCtrl.A_GetSecond( in := dcTime, get => wSecond );
fbCtrl.A_GetMilli( in := dcTime, get => wMilli );
fbCtrl.A_GetMicro( in := dcTime, get => wMicro );
fbCtrl.A_GetNano( in := dcTime, get => wNano );
```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.3 DCTIME64 和 ULINT

### 10.3.1 F\_ConvExtTimeToDcTime64



功能 F\_ConvExtTimeToDcTime64 可将外部时间转换为 TwinCAT 分布式时钟系统时间。

功能 F\_ConvExtTimeToDcTime64: T\_DCTIME64

#### 输入

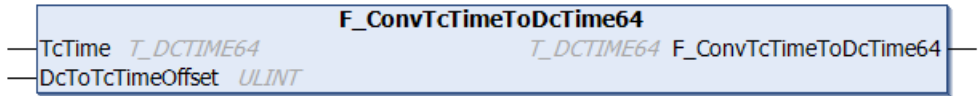
```
VAR_INPUT
    ExtTime      : T_DCTIME64;
    DcToExtTimeOffset : ULINT;
END_VAR
```

| 名称                 | 类型         | 描述                                |
|--------------------|------------|-----------------------------------|
| ExtTime            | T_DCTIME64 | 采用 TwinCAT “分布式时钟” 系统时间格式的外部时间    |
| DcToExtTime Offset | ULINT      | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量 |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 10.3.2 F\_ConvTcTimeToDcTime64



功能 F\_ConvTcTimeToDcTime64 可将 TwinCAT 系统时间转换为 TwinCAT 分布式时钟系统时间。

功能 F\_ConvTcTimeToDcTime64: T\_DCTIME64

#### 输入

```
VAR_INPUT
    TcTime      : T_DCTIME64;
    DcToTcTimeOffset : ULINT;
END_VAR
```

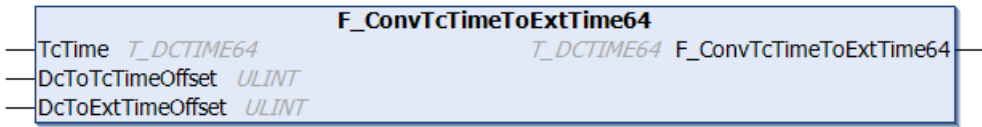
| 名称               | 类型         | 描述                                         |
|------------------|------------|--------------------------------------------|
| TcTime           | T_DCTIME64 | 采用 TwinCAT “分布式时钟” 系统时间格式的 TwinCAT 系统时间    |
| DcToTcTimeOffset | ULINT      | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |



10.3.3 F\_ConvTcTimeToExtTime64



功能 F\_ConvTcTimeToExtTime64 可将 TwinCAT 分布式时钟系统时间转换为外部时间。

功能 F\_ConvTcTimeToExtTime64: T\_DCTIME64

输入

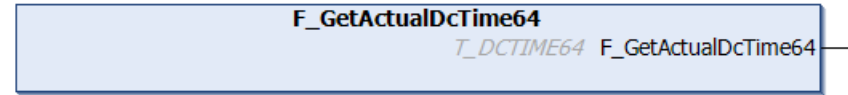
```
VAR_INPUT
    TcTime          : T_DCTIME64;
    DcToTcTimeOffset : ULINT;
    DcToExtTimeOffset : ULINT;
END_VAR
```

| 名称                | 类型         | 描述                                        |
|-------------------|------------|-------------------------------------------|
| TcTime            | T_DCTIME64 | 采用“分布式时钟”格式的 TwinCAT 系统时间                 |
| DcToTcTimeOffset  | ULINT      | TwinCAT “分布式时钟”系统时间与 TwinCAT 系统时间之间的时间偏移量 |
| DcToExtTimeOffset | ULINT      | TwinCAT “分布式时钟”系统时间与外部时间之间的时间偏移量          |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.3.4 F\_GetActualDcTime64



该功能以 TwinCAT 分布式时钟系统时间格式(T\_DCTIME64 [► 117]) 返回当前时间。

功能 F\_GetActaulDcTime: T\_DCTIME64

输入

```
VAR_INPUT
    (*none*)
END_VAR
```

ST 中的示例:

```
PROGRAM MAIN
VAR
    actDC : T_DCTIME64;
    sAct  : STRING;
END_VAR

actDC := F_GetActualDcTime64();
sAct  := DCTIME64_TO_STRING( actDC );
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 10.3.5 F\_GetCurDcTaskTime64

**F\_GetCurDcTaskTime64**  
T\_DCTIME64 F\_GetCurDcTaskTime64

该功能以 TwinCAT “分布式时钟” 系统时间格式 (T\_DCTIME64 [► 117]) 返回任务开始时间，即任务应开始的时间。该功能始终会返回调用时任务的开始时间。

**功能 F\_GetCurDcTaskTime64: T\_DCTIME64**

#### 输入

```
VAR_INPUT
(*none*)
END_VAR
```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 10.3.6 F\_GetCurDcTickTime64

**F\_GetCurDcTickTime64**  
T\_DCTIME64 F\_GetCurDcTickTime64

该功能以 TwinCAT 分布式时钟系统时间格式 (T\_DCTIME64 [► 117]) 返回当前（最后）工作的时间。

**功能 F\_GetCurDcTickTime64: T\_DCTIME64**

#### 输入

```
VAR_INPUT
(*none*)
END_VAR
```

#### 示例:

```
PROGRAM MAIN
VAR
    tDC : T_DCTIME64;
    sDC : STRING;
    tDCBack : T_DCTIME64;

    sDCZero : STRING;(* DCTIME64 = zero time starts on 01.01.2000 *)
    tDCBackFromZero : T_DCTIME64;

    tDCFromString : T_DCTIME64;
    sDCBackFromString : STRING;
END_VAR

tDC := F_GetCurDcTickTime64();
sDC := DCTIME64_TO_STRING( tDC );
tDCBack := STRING_TO_DCTIME64( sDC );

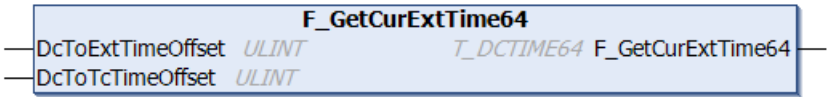
sDCZero := DCTIME64_TO_STRING( ULARGE_INTEGER(0, 0) );
tDCBackFromZero := STRING_TO_DCTIME64( sDCZero );

tDCFromString := STRING_TO_DCTIME64( '2007-03-09-11:31:09.223456789' );
sDCBackFromString := DCTIME64_TO_STRING( tDCFromString );
```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.3.7 F\_GetCurExtTime64



该功能以 TwinCAT 分布式时钟系统时间格式 (T\_DCTIME64 [► 117]) 返回外部时间。

功能 F\_GetCurExtTime64: T\_DCTIME64

输入

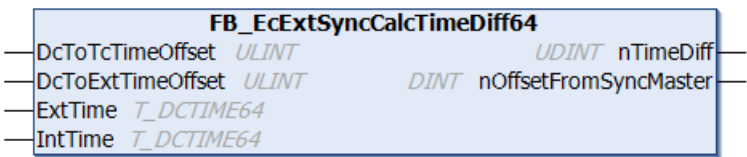
```
VAR_INPUT
    DcToExtTimeOffset : ULINT;
    DcToTcTimeOffset : ULINT;
END_VAR
```

| 名称                | 类型    | 描述                                         |
|-------------------|-------|--------------------------------------------|
| DcToExtTimeOffset | ULINT | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量          |
| DcToTcTimeOffset  | ULINT | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.3.8 FB\_EcExtSyncCalcTimeDiff64



功能块 FB\_EcExtSyncCalcTimeDiff64 会在考虑时间偏移量的情况下，计算外部时间和内部时间之间的差值。

输入/输出

```
VAR_IN_OUT
    DcToTcTimeOffset : ULINT;
    DcToExtTimeOffset : ULINT;
    ExtTime          : T_DCTIME64;
    IntTime          : T_DCTIME64;
END_VAR
```

| 名称                | 类型         | 描述                                         |
|-------------------|------------|--------------------------------------------|
| DcToTcTimeOffset  | ULINT      | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |
| DcToExtTimeOffset | ULINT      | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量          |
| ExtTime           | T_DCTIME64 | 采用 TwinCAT “分布式时钟” 系统时间格式的外部时间             |
| IntTime           | T_DCTIME64 | 采用 TwinCAT “分布式时钟” 系统时间格式的内部时间             |

 输出

```

VAR_OUTPUT
    nTimeDiff      : UDINT; (*with difference greater than 32 bit timeDiff = 0xffffffff*)
    nOffsetFromSyncMaster : DINT; (*less than 32 bit int Offset = 0x80000000, greater than 32 bit int
Offset = 0x7FFFFFFF*)
END_VAR

```

| 名称                    | 类型    | 描述                                                                                                             |
|-----------------------|-------|----------------------------------------------------------------------------------------------------------------|
| nTimeDiff             | UDINT | 如果差值小于 32 位，则会返回时间差。如果差值大于 32 位，则会返回 16#FFFFFFFF。                                                              |
| nOffsetFromSyncMaster | DINT  | 如果差值大于 32 位，且内部时间与 DC 时间的偏移量小于 32 位，则会返回 16#80000000。<br>如果差值大于 32 位，且内部时间与 DC 时间的偏移量大于 32 位，则会返回 16#7FFFFFFF。 |

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.3.9 FB\_EcExtSyncCheck64



功能块 FB\_EcExtSyncCheck64 可检查内部时钟和外部时钟是否同步。请参见功能块 FB\_EcExtSyncCalcTimeDiff64 [► 91]。

 输入

```

VAR_INPUT
    nSyncWindow : UDINT;
    bNotConnected : BOOL;
END_VAR

```

| 名称            | 类型    | 描述                     |
|---------------|-------|------------------------|
| nSyncWindow   | UDINT | 内部时钟和外部时钟被视为同步的时间窗口。   |
| bNotConnected | BOOL  | TRUE (真) = 与外部时钟的连接中断。 |

 输入/输出

```

VAR_IN_OUT
    DcToTcTimeOffset : T_LARGE_INTEGER;
    DcToExtTimeOffset : T_LARGE_INTEGER;
    ExtTime          : T_DCTIME64;
    IntTime          : T_DCTIME64;
END_VAR

```

| 名称                | 类型              | 描述                                         |
|-------------------|-----------------|--------------------------------------------|
| DcToTcTimeOffset  | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |
| DcToExtTimeOffset | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量          |
| ExtTime           | T_DCTIME64      | 采用 TwinCAT “分布式时钟” 系统时间格式的外部时间             |
| IntTime           | T_DCTIME64      | 采用 TwinCAT “分布式时钟” 系统时间格式的内部时间             |

输出

```
VAR_OUTPUT
    bSynchronized      : BOOL;
    nTimeDiff          : UDINT;
    nOffsetFromSyncMaster : DINT;
END_VAR
```

| 名称                    | 类型    | 描述                    |
|-----------------------|-------|-----------------------|
| bSynchronized         | BOOL  | TRUE（真） = 外部时钟和内部时钟同步 |
| nTimeDiff             | UDINT | 两个时钟之间的当前时差           |
| nOffsetFromSyncMaster | DINT  | 同步主站的偏移量              |

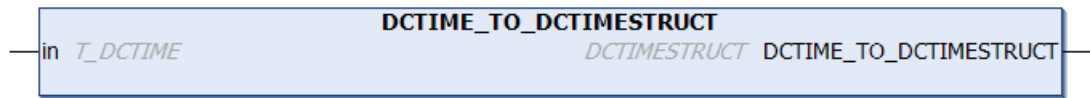
前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

10.4 [废弃]

10.4.1 [过时的 DCTIME]

10.4.1.1 DCTIME\_TO\_DCTIMESTRUCT



**过时的功能**

该功能已过时。使用功能 `DCTIME64_TO_DCTIMESTRUCT` [► 81] 代替。

该功能可将 `T_DCTIME` [► 118] 类型的 64 位分布式时钟系统时间变量转换为 `DCTIMESTRUCT` [► 116] 类型的结构化变量。

功能 DCTIME\_TO\_DCTIMESTRUCT: DCTIMESTRUCT

输入

```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

| 名称 | 类型       | 描述                 |
|----|----------|--------------------|
| in | T_DCTIME | 要转换的“分布式时钟系统时间”变量。 |

示例:

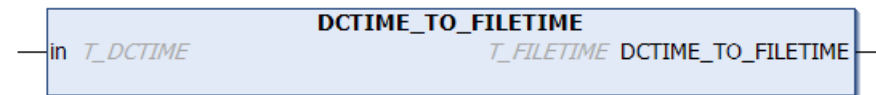
```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT;
    dcTime   : T_DCTIME;
END_VAR

dcTime := F_GetCurDcTickTime();
dcStruct := DCTIME_TO_DCTIMESTRUCT(dcTime);
```

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.1.2 DCTIME\_TO\_FILETIME



## ● 过时的功能

**i** 该功能已过时。使用功能 [DCTIME64\\_TO\\_FILETIME](#) [► 106] 代替。

该功能可将 [T\\_DCTIME](#) [► 118] 类型的 64 位分布式时钟系统时间变量转换为 [T\\_FILETIME](#) 类型的 64 位 Windows 文件时间变量。

## 功能 DCTIME\_TO\_FILETIME: T\_FILETIME

## 🚩 输入

```
VAR_INPUT
    in : T_DCTIME;
END_VAR
```

| 名称 | 类型       | 描述                |
|----|----------|-------------------|
| in | T_DCTIME | 要转换的“分布式时钟系统时间”变量 |

## 示例:

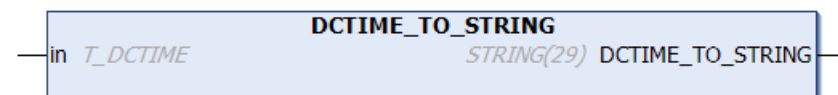
```
PROGRAM P_TEST
VAR
    ft : T_FILETIME;
    dct : T_DCTIME;
END_VAR

dct := F_GetCurDcTickTime();
ft := DCTIME_TO_FILETIME(dct);
```

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.1.3 DCTIME\_TO\_STRING



## ● 过时的功能

**i** 该功能已过时。使用功能 [DCTIME64\\_TO\\_STRING](#) [► 83] 代替。

该功能可将字符串转换为 [T\\_DCTIME](#) [► 118] 类型的分布式时钟系统时间变量。

转换后的字符串格式如下: 'YYYY-MM-DD-hh:mm:ss.nnnnnnnnn'

- YYYY: 年;
- MM: 月;

- DD: 日;
- hh: 小时;
- mm: 分钟;
- ss: 秒;
- nnnnnnnnn: 纳秒;

功能 DCTIME\_TO\_STRING: 字符串 (29)

输入

```
VAR_INPUT
  in : T_DCTIME;
END_VAR
```

| 名称 | 类型       | 描述                |
|----|----------|-------------------|
| in | T_DCTIME | 要转换的“分布式时钟系统时间”变量 |

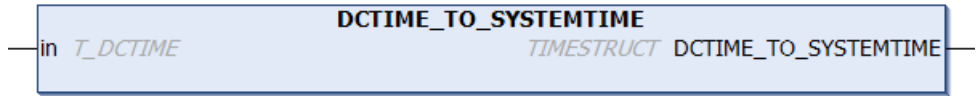
示例:

请参见功能: [F\\_GetCurDcTickTime \[► 102\]](#) 的说明。

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.4.1.4 DCTIME\_TO\_SYSTEMTIME



过时的功能

该功能已过时。使用功能 [DCTIME64\\_TO\\_SYSTEMTIME \[► 83\]](#) 代替。

该功能可将 T\_DCTIME [► 118] 类型的 64 位分布式时钟系统时间变量转换为 Timestruct 类型的结构化 Windows 系统时间变量。

功能 DCTIME\_TO\_SYSTEMTIME: Timestruct

输入

```
VAR_INPUT
  in : T_DCTIME;
END_VAR
```

| 名称 | 类型       | 描述                |
|----|----------|-------------------|
| in | T_DCTIME | 要转换的“分布式时钟系统时间”变量 |

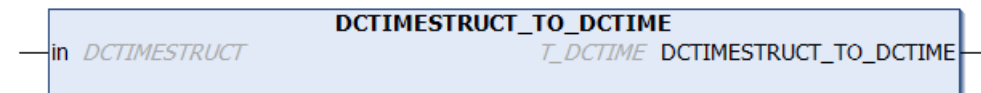
示例:

```
PROGRAM P_TEST
VAR
  syst : Timestruct;
END_VAR
syst := DCTIME_TO_SYSTEMTIME ( F_GetCurDcTickTime() );
```

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.1.5 DCTIMESTRUCT\_TO\_DCTIME



## 过时的功能

该功能已过时。使用功能 [DCTIMESTRUCT\\_TO\\_DCTIME64](#) [► 84] 代替。

该功能可将 [DCTIMESTRUCT](#) [► 116] 类型的结构化变量转换为 [T\\_DCTIME](#) [► 118] 类型的 64 位分布式时钟系统时间变量。

在转换过程中，结构变量 wDayOfWeek 会被忽略。结构变量 wYear 必须大于或等于 2000，并小于 2584。对于结构变量出现无效值，功能会返回零值。

## 功能 DCTIMESTRUCT\_TO\_DCTIME: T\_DCTIME

## 输入

```
VAR_INPUT
    in : DCTIMESTRUCT;
END_VAR
```

| 名称 | 类型           | 描述        |
|----|--------------|-----------|
| in | DCTIMESTRUCT | 要转换的结构化变量 |

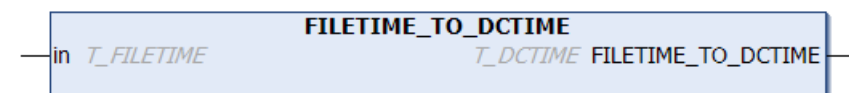
## 示例:

```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT := ( wYear := 2008, wMonth := 3, wDay := 13,
                                wHour := 1, wMinute := 2, wSecond := 3,
                                wMilliseconds := 123, wMicroseconds := 456, wNanoseconds := 789 );
    dc64 : T_DCTIME;
END_VAR
dc64 := DCTIMESTRUCT_TO_DCTIME( dcStruct );
```

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.1.6 FILETIME\_TO\_DCTIME



## 过时的功能

该功能已过时。使用功能 [FILETIME\\_TO\\_DCTIME64](#) [► 106] 代替。

该功能可将 T\_FILETIME 类型的 64 位 Windows 文件时间变量转换为 [T\\_DCTIME](#) [► 118] 类型的 64 位分布式时钟系统时间变量。如果出现转换错误，则功能会返回 0。



功能 FILETIME\_TO\_DCTIME: T\_DCTIME

🔧 输入

```
VAR_INPUT
    in : T_FILETIME;
END_VAR
```

| 名称 | 类型         | 描述                         |
|----|------------|----------------------------|
| in | T_FILETIME | 要转换的“Windows File Time” 变量 |

示例:

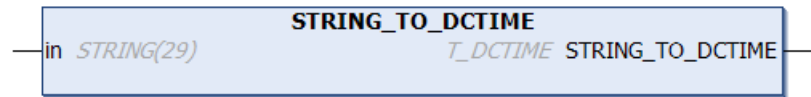
```
PROGRAM P_TEST
VAR
    fbSysFileTime : GETSYSTEMTIME;
    ft : T_FILETIME;
    dct : T_DCTIME;
END_VAR

fbSysFileTime(timeLoDW=>ft.dwLowDateTime, timeHiDW=>ft.dwHighDateTime);
dct := FILETIME_TO_DCTIME(ft);
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.4.1.7 STRING\_TO\_DCTIME





**过时的功能**

该功能已过时。使用功能 [STRING\\_TO\\_DCTIME64 \[► 85\]](#) 代替。

该功能可将字符串转换为 [T\\_DCTIME \[► 118\]](#) 类型的分布式时钟系统时间变量。

功能 STRING\_TO\_DCTIME: T\_DCTIME

🔧 输入

```
VAR_INPUT
    in : STRING(29);
END_VAR
```

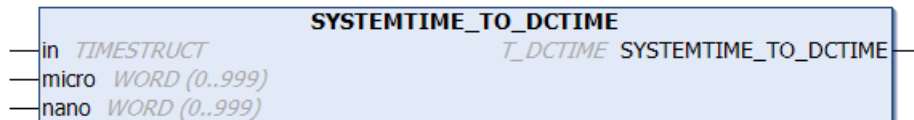
| 名称 | 类型     | 描述                                                                                                                                                                                                                                   |
|----|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| in | STRING | 要转换的字符串<br>字符串必须具有以下格式: 'YYYY-MM-DD-hh:mm:ss:nnnnnnnnnn' <ul style="list-style-type: none"><li>• YYYY: 年;</li><li>• MM: 月;</li><li>• DD: 日;</li><li>• hh: 小时;</li><li>• mm: 分钟;</li><li>• ss: 秒;</li><li>• nnnnnnnnnn: 纳秒;</li></ul> |

请参见功能: [F\\_GetCurDcTickTime \[► 102\]](#) 的说明。

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.1.8 SYSTEMTIME\_TO\_DCTIME



## 过时的功能

该功能已过时。使用功能 [SYSTEMTIME\\_TO\\_DCTIME64](#) [► 86] 代替。

该功能可将 TIMESTRUCT 类型的结构化 Windows 系统时间变量转换为 [T\\_DCTIME](#) [► 118] 类型的 64 位分布式时钟系统时间变量。如果出现转换错误，则功能会返回 0。

## 功能 SYSTEMTIME\_TO\_DCTIME: T\_DCTIME

## 输入

```
VAR_INPUT
    in      : TIMESTRUCT;
    micro   : WORD(0..999); (* Microseconds: 0..999 *)
    nano    : WORD(0..999); (* Nanoseconds: 0..999 *)
END_VAR
```

| 名称    | 类型         | 描述                          |
|-------|------------|-----------------------------|
| in    | TIMESTRUCT | 要转换的“Windows System Time”变量 |
| Micro | WORD       | 微秒                          |
| nano  | WORD       | 纳秒                          |

## 示例:

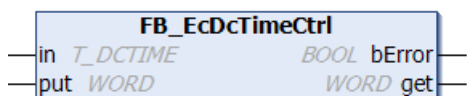
```
PROGRAM P_TEST
VAR
    syst : TIMESTRUCT := ( wYear := 2009, wMonth := 9, wDay := 16, wHour := 12, wMinute := 22, wSeco
nd := 44, wMilliseconds := 123 );
END_VAR

dct := SYSTEMTIME_TO_DCTIME( syst, 456, 789 );
```

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.1.9 FB\_EcDcTimeCtrl



## 过时的功能

该功能已过时。使用功能块 [FB\\_EcDcTimeCtrl64](#) [► 86] 代替。

该功能块可用于读取 T\_DCTIME [► 118] 类型的 64 位“分布式时钟系统时间”变量中的年、月、日等各个组件。该功能块有多个 A\_GetXYZ 操作。调用所需操作之后，“get”输出变量中就会出现 XYZ 组件的值。目前未使用“put”输入变量。

该功能块目前有以下操作：

- A\_GetYear;
- A\_GetMonth;
- A\_GetDay;
- A\_GetDayOfWeek;
- A\_GetHour;
- A\_GetMinute;
- A\_GetSecond;
- A\_GetMilli;
- A\_GetMicro;
- A\_GetNano;

#### 输入

```
VAR_INPUT
    put : WORD;
END_VAR
```

| 名称  | 类型   | 描述          |
|-----|------|-------------|
| put | WORD | 输入参数（目前未使用） |

#### 输入/输出

```
VAR_IN_OUT
    in : T_DCTIME;
END_VAR
```

| 名称 | 类型       | 描述                    |
|----|----------|-----------------------|
| in | T_DCTIME | TwinCAT “分布式时钟系统时间”变量 |

#### 输出

```
VAR_OUTPUT
    bError : BOOL;
    get    : WORD;
END_VAR
```

| 名称     | 类型   | 描述                     |
|--------|------|------------------------|
| bError | BOOL | 如果操作调用过程中出现错误，则会设置此输出。 |
| get    | WORD | 输出参数（年、月、日等）           |

#### 在 ST 中实施的示例：

```
PROGRAM P_TEST
VAR
    dcStruct : DCTIMESTRUCT;
    dcTime   : T_DCTIME;
    fbCtrl   : FB_EcDcTimeCtrl;

    wYear    : WORD;
    wMonth   : WORD;
    wDay     : WORD;
    wDayOfWeek : WORD;
    wHour    : WORD;
    wMinute  : WORD;
    wSecond  : WORD;
    wMilli   : WORD;
    wMicro   : WORD;
    wNano    : WORD;
END_VAR

dcTime := F_GetCurDcTickTime();
```

```

fbCtrl.A_GetYear( in := dcTime, get => wYear );
fbCtrl.A_GetMonth( in := dcTime, get => wMonth );
fbCtrl.A_GetDay( in := dcTime, get => wDay );
fbCtrl.A_GetDayOfWeek( in := dcTime, get => wDayOfWeek );
fbCtrl.A_GetHour( in := dcTime, get => wHour );
fbCtrl.A_GetMinute( in := dcTime, get => wMinute );
fbCtrl.A_GetSecond( in := dcTime, get => wSecond );
fbCtrl.A_GetMilli( in := dcTime, get => wMilli );
fbCtrl.A_GetMicro( in := dcTime, get => wMicro );
fbCtrl.A_GetNano( in := dcTime, get => wNano );

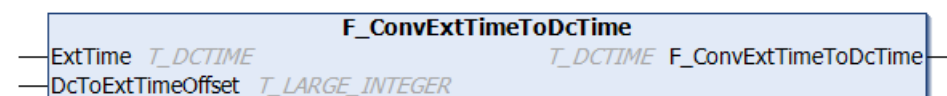
```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.2 [过期的 DCTIME 和 T\_LARGE\_INTEGER]

### 10.4.2.1 F\_ConvExtTimeToDcTime



#### 过时的功能

该功能已过时。使用功能 [F\\_ConvExtTimeToDcTime64](#) [► 88] 代替。

功能 [F\\_ConvExtTimeToDcTime](#) 可将外部时间转换为 TwinCAT 分布式时钟系统时间。

#### 功能 [F\\_ConvExtTimeToDcTime](#): T\_DCTIME

#### 输入

```

VAR_INPUT
    ExtTime      : T_DCTIME;
    DcToExtTimeOffset : T_LARGE_INTEGER;
END_VAR

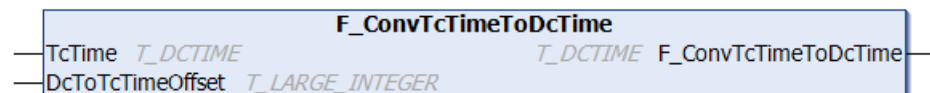
```

| 名称                 | 类型              | 描述                                |
|--------------------|-----------------|-----------------------------------|
| ExtTime            | T_DCTIME        | 采用 TwinCAT “分布式时钟” 系统时间格式的外部时间    |
| DcToExtTime Offset | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量 |

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 10.4.2.2 F\_ConvTcTimeToDcTime



#### 过时的功能

该功能已过时。使用功能 [F\\_ConvTcTimeToDcTime64](#) [► 88] 代替。

功能 [F\\_ConvTcTimeToDcTime64](#) 可将 TwinCAT 系统时间转换为 TwinCAT 分布式时钟系统时间。

功能 F\_ConvTcTimeToDcTime: T\_DCTIME

输入

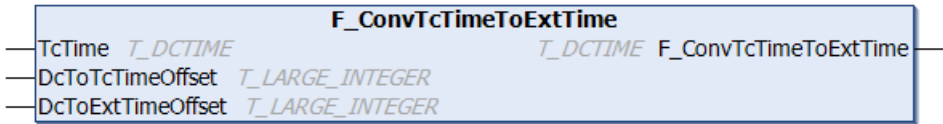
```
VAR_INPUT
    TcTime      : T_DCTIME;
    DcToTcTimeOffset : T_LARGE_INTEGER;
END_VAR
```

| 名称               | 类型              | 描述                                         |
|------------------|-----------------|--------------------------------------------|
| TcTime           | T_DCTIME        | 采用 TwinCAT “分布式时钟” 系统时间格式的 TwinCAT 系统时间    |
| DcToTcTimeOffset | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.4.2.3 F\_ConvTcTimeToExtTime



**过时的功能**  
该功能已过时。使用功能 [F\\_ConvTcTimeToExtTime64](#) [► 89] 代替。

功能 F\_ConcTcTimeToExtTime 可将 TwinCAT 分布式时钟系统时间转换为外部时间。

功能 F\_ConvTcTimeToExtTime: T\_DCTIME

输入

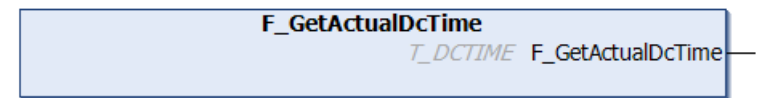
```
VAR_INPUT
    TcTime      : T_DCTIME;
    DcToTcTimeOffset : T_LARGE_INTEGER;
    DcToExtTimeOffset : T_LARGE_INTEGER;
END_VAR
```

| 名称                | 类型              | 描述                                         |
|-------------------|-----------------|--------------------------------------------|
| TcTime            | T_DCTIME        | 采用 “分布式时钟” 格式的 TwinCAT 系统时间                |
| DcToTcTimeOffset  | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |
| DcToExtTimeOffset | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量          |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.4.2.4 F\_GetActualDcTime





### 过时的功能

该功能已过时。使用功能 [F\\_GetActualDcTime64 \[► 89\]](#) 代替。

该功能以 TwinCAT 分布式时钟系统时间格式([T\\_DCTIME \[► 118\]](#)) 返回当前时间。

#### 功能 F\_GetActualDcTime: T\_DCTIME

##### 输入

```
VAR_INPUT
(*none*)
END_VAR
```

##### 示例:

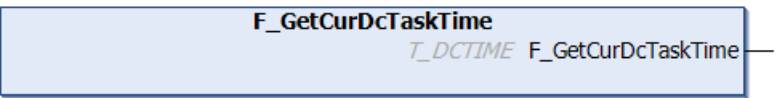
```
PROGRAM MAIN
VAR
    actDC : T_DCTIME;
    sAct : STRING;
END_VAR

actDC := F_GetActualDcTime();
sAct := DCTIME_TO_STRING( actDC );
```

##### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.2.5 F\_GetCurDcTaskTime



### 过时的功能

该功能已过时。使用功能 [F\\_GetCurDcTaskTime64 \[► 90\]](#) 代替。

该功能以 TwinCAT 分布式时钟系统时间格式 ([T\\_DCTIME \[► 118\]](#)) 返回任务开始时间（任务应开始的时间）。该功能始终会返回调用时任务的开始时间。

#### 功能 F\_GetCurDcTaskTime: T\_DCTIME

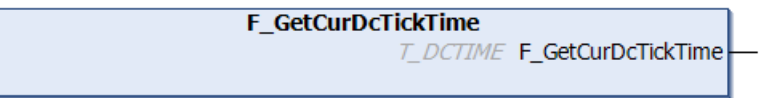
##### 输入

```
VAR_INPUT
(*none*)
END_VAR
```

##### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 10.4.2.6 F\_GetCurDcTickTime





过时的功能

该功能已过时。使用功能 [F\\_GetCurDcTickTime64](#) [► 90] 代替。

该功能以 TwinCAT 分布式时钟系统时间格式 ([T\\_DCTIME](#) [► 118]) 返回当前（最后）工作的时间。

功能 [F\\_GetCurDcTickTime](#): [T\\_DCTIME](#)

输入

```
VAR_INPUT
(*none*)
END_VAR
```

示例:

```
PROGRAM MAIN
VAR
    tDC : T_DCTIME;
    sDC : STRING;
    tDCBack : T_DCTIME;

    sDCZero : STRING;(* DCTIME = zero time starts on 01.01.2000 *)
    tDCBackFromZero : T_DCTIME;

    tDCFromString : T_DCTIME;
    sDCBackFromString : STRING;
END_VAR

tDC := F_GetCurDcTickTime();
sDC := DCTIME_TO_STRING( tDC );
tDCBack := STRING_TO_DCTIME( sDC );

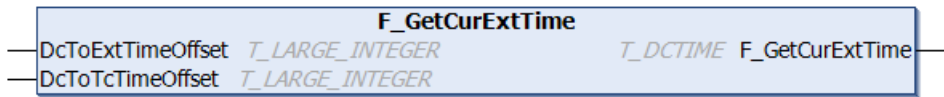
sDCZero := DCTIME_TO_STRING( ULARGE_INTEGER(0, 0) );
tDCBackFromZero := STRING_TO_DCTIME( sDCZero );

tDCFromString := STRING_TO_DCTIME( '2007-03-09-11:31:09.223456789' );
sDCBackFromString := DCTIME_TO_STRING( tDCFromString );
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.4.2.7 [F\\_GetCurExtTime](#)



过时的功能

该功能已过时。使用功能 [F\\_GetCurExtTime64](#) [► 91] 代替。

该功能以 TwinCAT 分布式时钟系统时间格式 ([T\\_DCTIME](#) [► 118]) 返回外部时间。

功能 [F\\_GetCurExtTime](#): [T\\_DCTIME](#)

输入

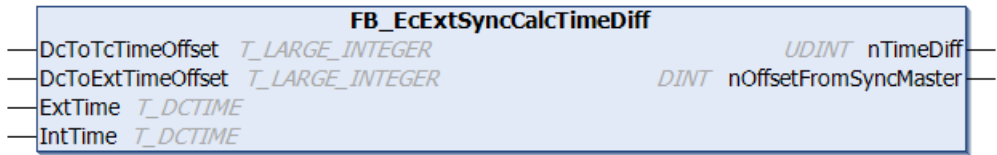
```
VAR_INPUT
    DcToExtTimeOffset : T_LARGE_INTEGER;
    DcToTcTimeOffset : T_LARGE_INTEGER;
END_VAR
```

| 名称                    | 类型                  | 描述                                         |
|-----------------------|---------------------|--------------------------------------------|
| DcToExtTime<br>Offset | T_LARGE_<br>INTEGER | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量          |
| DcToTcTime<br>Offset  | T_LARGE_<br>INTEGER | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 10.4.2.8 FB\_EcExtSyncCalcTimeDiff



**过时的功能块**

该功能块已过时。使用功能块 [FB\\_EcExtSyncCalcTimeDiff64 \[► 91\]](#) 代替。

功能块 `FB_EcExtSyncCalcTimeDiff` 会在考虑时间偏移量的情况下，计算外部时间和内部时间之间的差值。

**输入/输出**

```
VAR_IN_OUT
  DcToTcTimeOffset : T_LARGE_INTEGER;
  DcToExtTimeOffset : T_LARGE_INTEGER;
  ExtTime          : T_DCTIME;
  IntTime          : T_DCTIME;
END_VAR
```

| 名称                    | 类型                  | 描述                                         |
|-----------------------|---------------------|--------------------------------------------|
| DcToTcTime<br>Offset  | T_LARGE_<br>INTEGER | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |
| DcToExtTime<br>Offset | T_LARGE_<br>INTEGER | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量          |
| ExtTime               | T_DCTIME            | 采用 TwinCAT “分布式时钟” 系统时间格式的外部时间             |
| IntTime               | T_DCTIME            | 采用 TwinCAT “分布式时钟” 系统时间格式的内部时间             |

**输出**

```
VAR_OUTPUT
  nTimeDiff      : UDINT; (*with difference greater than 32 bit timeDiff = 0xffffffff*)
  nOffsetFromSyncMaster : DINT; (*less than 32 bit int Offset = 0x80000000, greater than 32 bit
  int Offset = 0x7FFFFFFF*)
END_VAR
```

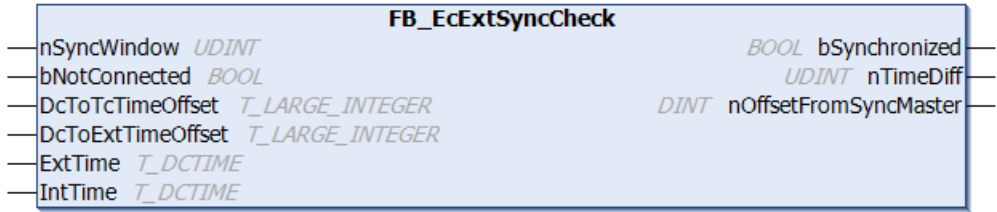
| 名称                        | 类型    | 描述                                                                                                             |
|---------------------------|-------|----------------------------------------------------------------------------------------------------------------|
| nTimeDiff                 | UDINT | 如果差值小于 32 位，则会返回时间差。如果差值大于 32 位，则会返回 16#FFFFFFFF。                                                              |
| nOffsetFrom<br>SyncMaster | DINT  | 如果差值大于 32 位，且内部时间与 DC 时间的偏移量小于 32 位，则会返回 16#80000000。<br>如果差值大于 32 位，且内部时间与 DC 时间的偏移量大于 32 位，则会返回 16#7FFFFFFF。 |



前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

10.4.2.9 FB\_EcExtSyncCheck



**过时的功能块**  
该功能块已过时。使用功能块 [FB\\_EcExtSyncCheck64](#) [► 92] 代替。

功能块 [FB\\_EcExtSyncCheck](#) 可检查内部时钟和外部时钟是否同步。请参见功能块 [FB\\_EcExtSyncCalcTimeDiff](#) [► 104]。

输入

```
VAR_INPUT
    nSyncWindow      : UDINT;
    bNotConnected    : BOOL;
END_VAR
```

| 名称            | 类型    | 描述                    |
|---------------|-------|-----------------------|
| nSyncWindow   | UDINT | 内部时钟和外部时钟被视为同步的时间窗口。  |
| bNotConnected | BOOL  | TRUE（真） = 与外部时钟的连接中断。 |

输入/输出

```
VAR_IN_OUT
    DcToTcTimeOffset : T_LARGE_INTEGER;
    DcToExtTimeOffset : T_LARGE_INTEGER;
    ExtTime           : T_DCTIME;
    IntTime           : T_DCTIME;
END_VAR
```

| 名称                | 类型              | 描述                                         |
|-------------------|-----------------|--------------------------------------------|
| DcToTcTimeOffset  | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与 TwinCAT 系统时间之间的时间偏移量 |
| DcToExtTimeOffset | T_LARGE_INTEGER | TwinCAT “分布式时钟” 系统时间与外部时间之间的时间偏移量          |
| ExtTime           | T_DCTIME        | 采用 TwinCAT “分布式时钟” 系统时间格式的外部时间             |
| IntTime           | T_DCTIME        | 采用 TwinCAT “分布式时钟” 系统时间格式的内部时间             |

输出

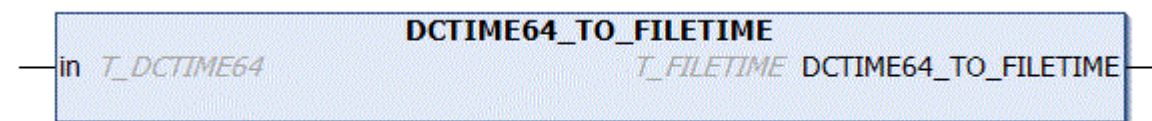
```
VAR_OUTPUT
    bSynchronized      : BOOL;
    nTimeDiff           : UDINT;
    nOffsetFromSyncMaster : DINT;
END_VAR
```

| 名称                     | 类型    | 描述                   |
|------------------------|-------|----------------------|
| bSynchronized          | BOOL  | TRUE（真）= 外部时钟和内部时钟同步 |
| nTimeDiff              | UDINT | 两个时钟之间的当前时差          |
| nOffsetFrom SyncMaster | DINT  | 同步主站的偏移量             |

#### 前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

### 10.4.3 DCTIME64\_TO\_FILETIME



该功能可将 T\_DCTIME64 [► 117] 类型的 64 位分布式时钟系统时间变量转换为 T\_FILETIME 类型的 64 位 Windows 文件时间变量。

#### 功能 DCTIME64\_TO\_FILETIME: T\_FILETIME

##### 输入

```
VAR_INPUT
    in : T_DCTIME64;
END_VAR;
```

| 名称 | 类型         | 描述                |
|----|------------|-------------------|
| in | T_DCTIME64 | 要转换的“分布式时钟系统时间”变量 |

#### 示例:

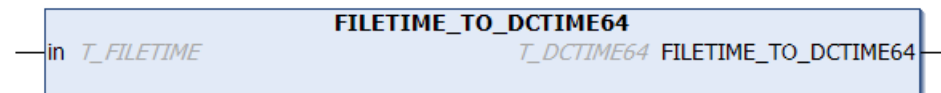
```
PROGRAM P_TEST
VAR
    ft : T_FILETIME;
    dct : T_DCTIME64;
END_VAR

dct := F_GetCurDcTickTime64();
ft := DCTIME64_TO_FILETIME(dct);
```

#### 前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

### 10.4.4 FILETIME\_TO\_DCTIME64



该功能可将 T\_FILETIME 类型的 64 位 Windows 文件时间变量转换为 T\_DCTIME64 [► 117] 类型的 64 位分布式时钟系统时间变量。如果出现转换错误，则功能会返回 0。

功能 FILETIME\_TO\_DCTIME64: T\_DCTIME64

 输入

```
VAR_INPUT
    in : T_FILETIME;
END_VAR
```

| 名称 | 类型         | 描述                        |
|----|------------|---------------------------|
| in | T_FILETIME | 要转换的“Windows File Time”变量 |

示例:

```
PROGRAM P_TEST
VAR
    fbSysFileTime : GETSYSTEMTIME;
    ft : T_FILETIME;
    dct : T_DCTIME64;
END_VAR

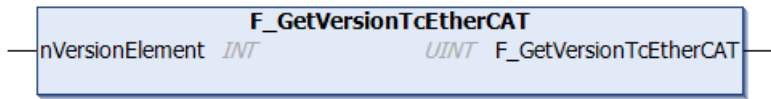
fbSysFileTime(timeLoDW=>ft.dwLowDateTime, timeHiDW=>ft.dwHighDateTime);
dct := FILETIME_TO_DCTIME64(ft);
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

# 11 [废弃]

## 11.1 F\_GetVersionTcEtherCAT



**过时的功能**  
该功能已过时。使用全局结构实例 stLibVersion\_Tc2\_EtherCAT 代替

该功能可用于读取 PLC 库的版本信息。

功能 F\_GetVersionTcEtherCAT: UINT

 输入

```

VAR_INPUT
    nVersionElement : INT;
END_VAR

```

| 名称              | 类型  | 描述                                                                                                                |
|-----------------|-----|-------------------------------------------------------------------------------------------------------------------|
| nVersionElement | INT | 要读取的版本元素。可能的参数： <ul style="list-style-type: none"><li>• 1: 主要编号；</li><li>• 2: 次要编号；</li><li>• 3: 修订版本号；</li></ul> |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12 数据类型

### 12.1 E\_EcAdressingType

EtherCAT 中的寻址要么取决于位置 (eAdressingType\_AutoInc), 要么基于固定的配置地址 (eAdressingType\_Fixed), 要么适用于所有从站 (eAdressingType\_Broadcast)。

```
TYPE E_EcAdressingType :
(
  eAdressingType_AutoInc:=1, (* Adress slave by it's position. (adp = 1-
position, 1.Slave = 0, 2.Slave = 0xffff(-1) etc) *)
    (* EtherCAT commands: APRD, APWR, APRW *)
  eAdressingType_Fixed, (* Adress slave by configured ethercat slave address (adp = configured address
) *)
    (* EtherCAT commands: FPRD, FPWR, FPRW *)
  eAdressingType_Broadcast (* Adress all slaves. *)
    (* EtherCAT commands: BRD, BWR, BRW *)
);
END_TYPE
```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 12.2 E\_EcFoeMode

“File access over EtherCAT” 邮箱协议的访问模式。

```
TYPE E_EcFoeMode :
(
  eFoeMode_Write := 1,
  eFoeMode_Read
);
END_TYPE
```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 12.3 E\_EcMbxProtType

支持的 EtherCAT 邮箱协议类型。

```
TYPE E_EcMbxProtType:
(
  eEcMbxProt_CoE := 3, (* CANopen over EtherCAT *)
  eEcMbxProt_FoE := 4, (* File over EtherCAT *)
  eEcMbxProt_SoE := 5 (* Servo Drive Profile over EtherCAT *)
);
END_TYPE
```

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

### 12.4 ST\_EcCrcError

包含 EtherCAT 从站设备各个端口 (A、B 和 C) 的 CRC 错误计数器的结构。

```
TYPE ST_EcCrcError :
STRUCT
  portA : UDINT;
  portB : UDINT;
```

```

    portC : UDINT;
END_STRUCT
END_TYPE

```

| 名称    | 类型    | 描述               |
|-------|-------|------------------|
| portA | UDINT | 端口 A 的 CRC 错误计数器 |
| portB | UDINT | 端口 B 的 CRC 错误计数器 |
| portC | UDINT | 端口 C 的 CRC 错误计数器 |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.5 ST\_EcCrcErrorEx

包含 EtherCAT 从站设备各个端口（A、B、C 和 D）的 CRC 错误计数器的结构。

```

TYPE ST_EcCrcErrorEx :
STRUCT
    portA : UDINT;
    portB : UDINT;
    portC : UDINT;
    portD : UDINT;
END_STRUCT
END_TYPE

```

| 名称    | 类型    | 描述               |
|-------|-------|------------------|
| portA | UDINT | 端口 A 的 CRC 错误计数器 |
| portB | UDINT | 端口 B 的 CRC 错误计数器 |
| portC | UDINT | 端口 C 的 CRC 错误计数器 |
| portD | UDINT | 端口 D 的 CRC 错误计数器 |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.6 ST\_EcLastProtErrInfo

结构 ST\_EcLastProtErrInfo 包含与最近 EtherCAT 邮箱协议错误相关的附加错误信息。

```

TYPE ST_EcSlaveState:
STRUCT
    ownAddr : ST_AmsAddr;
    orgAddr : ST_AmsAddr;
    errCode : UDINT;
    binDesc : ARRAY[0..MAX_STRING_LENGTH] OF BYTE;
END_STRUCT
END_TYPE

```

| 名称      | 类型                                  | 描述                                            |
|---------|-------------------------------------|-----------------------------------------------|
| ownAddr | ST_AmsAddr                          | 自有的 AMS 地址（用于查询错误信息的通信设备地址）                   |
| orgAddr | ST_AmsAddr                          | 错误发起者的 AMS 地址（已触发或导致协议错误的通信设备地址）              |
| errCode | UDINT                               | 邮箱协议错误编号 [► 121]（SoE、CoE、FoE 错误代码）            |
| binDesc | ARRAY[0..MAX_STRING_LENGTH] OF BYTE | 采用二进制数据形式的附加错误信息。附加错误信息针对特定设备，可以包括字符串或二进制数据等。 |

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.7 ST\_EcMasterStatistic

```

TYPE ST_EcMasterStatistic :
STRUCT
    nSysTime          : UDINT;
    nCycFrameCnt      : UDINT;
    nCycFrameMissedCnt : UDINT;
    nQueuedFrameCnt   : UDINT;
    nQueuedFrameMissedCnt : UDINT;
END_STRUCT
END_TYPE

```

| 名称                    | 类型    | 描述                 |
|-----------------------|-------|--------------------|
| nSysTime              | UDINT | 系统时间 (μs)          |
| nCycFrameCnt          | UDINT | 周期 EtherCAT 帧数     |
| nCycFrameMissedCnt    | UDINT | 丢失的循环 EtherCAT 帧数  |
| nQueuedFrameCnt       | UDINT | 非周期 EtherCAT 帧数    |
| nQueuedFrameMissedCnt | UDINT | 丢失的非周期 EtherCAT 帧数 |

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.8 ST\_EcSlaveConfigData

结构 ST\_EcSlaveConfigData 包含 EtherCAT 从站设备的 EtherCAT 配置数据。

```

TYPE ST_EcSlaveConfigData:
STRUCT
    nEntries          : WORD;
    nAddr             : WORD;
    sType             : STRING[15];
    sName             : STRING[31];
    nDevType          : DWORD;
    stSlaveIdentity   : ST_EcSlaveIdentity;
    nMailboxOutSize   : WORD;
    nMailboxInSize    : WORD;
    nLinkStatus       : BYTE;
END_STRUCT
END_TYPE

```

| 名称              | 类型                 | 描述                                                               |
|-----------------|--------------------|------------------------------------------------------------------|
| nEntries        | WORD               | 内部使用                                                             |
| nAddr           | WORD               | EtherCAT 从站地址                                                    |
| sType           | STRING             | 从站的 EtherCAT 类型                                                  |
| sName           | STRING             | EtherCAT 从站的名称                                                   |
| nDevType        | DWORD              | 从站的 EtherCAT 设备类型                                                |
| stSlaveIdentity | ST_EcSlaveIdentity | EtherCAT 从站的标识 (请参见 <a href="#">ST_EcSlaveIdentity [► 112]</a> ) |
| nMailboxOutSize | WORD               | EtherCAT 从站的邮箱大小过大                                               |
| nMailboxInSize  | WORD               | EtherCAT 从站的邮箱大小过小                                               |
| nLinkStatus     | BYTE               | EtherCAT 从站的链接状态                                                 |

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.9 ST\_EcSlaveIdentity

结构 `ST_EcSlaveIdentity` 包含 EtherCAT 从站设备的 EtherCAT 标识数据。

```

TYPE ST_EcSlaveIdentity :
STRUCT
    vendorId      : UDINT;
    productCode   : UDINT;
    revisionNo    : UDINT;
    serialNo      : UDINT;
END_STRUCT
END_TYPE

```

| 名称          | 类型    | 描述            |
|-------------|-------|---------------|
| vendorId    | UDINT | 从站设备供应商 ID    |
| productCode | UDINT | 从站设备产品代码      |
| revisionNo  | UDINT | 表示从站设备的修订版本号。 |
| serialNo    | UDINT | 表示从站设备的序列号。   |

## 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.10 ST\_EcSlaveScannedData

结构 `ST_EcSlaveScannedData` 包含已扫描的 EtherCAT 从站设备的 EtherCAT 配置数据。

```

TYPE EcSlaveScannedData:
STRUCT
    nEntries      : WORD;
    nAddr         : WORD;
    stSlaveIdentity : ST_EcSlaveIdentity;
    ndlStatusReg  : WORD;
END_STRUCT
END_TYPE

```

| 名称              | 类型                 | 描述                                                                                                                                                                                                                      |
|-----------------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nEntries        | WORD               | 内部使用                                                                                                                                                                                                                    |
| nAddr           | WORD               | EtherCAT 从站地址                                                                                                                                                                                                           |
| stSlaveIdentity | ST_EcSlaveIdentity | EtherCAT 从站的标识（请参见 <code>ST_EcSlaveIdentity</code> [► 112]）                                                                                                                                                             |
| ndlStatusReg    | WORD               | 来自 ESC 寄存器 0110/0111 <sub>hex</sub> 或 272/273 <sub>dec</sub> 的 EtherCAT 从站链接状态。如果从站不可用/脱机，则显示状态 0。<br>“端口号 <=> 接口/Ebus 触点”的分配可以在相关设备文档中找到。除非另有说明，Port0 为 EL/ES 终端或 EP 盒 RJ45 接口的左侧 Ebus 触点，Port1 为右侧传出 Ebus 触点/RJ45 接口。 |

ndlStatusReg 的位含义如下：



| 位  | 含义                                 |
|----|------------------------------------|
| 1  | 内部使用                               |
| 2  | 内部使用                               |
| 3  | 内部使用                               |
| 4  | 端口 0 上的物理链接<br>0: 无链接,<br>1: 检测到链接 |
| 5  | 端口 1 上的物理链接<br>0: 无链接,<br>1: 检测到链接 |
| 6  | 端口 2 上的物理链接<br>0: 无链接,<br>1: 检测到链接 |
| 7  | 端口 3 上的物理链接<br>0: 无链接,<br>1: 检测到链接 |
| 8  | 环路端口 0<br>0: 打开,<br>1: 关闭          |
| 9  | 端口 0 上的通信<br>0: 无稳定通信,<br>1: 通信已建立 |
| 10 | 环路端口 1<br>0: 打开,<br>1: 关闭          |
| 11 | 端口 1 上的通信<br>0: 无稳定通信,<br>1: 通信已建立 |
| 12 | 环路端口 2<br>0: 打开,<br>1: 关闭          |
| 13 | 端口 2 上的通信<br>0: 无稳定通信<br>1: 通信已建立  |
| 14 | 环路端口 3<br>0: 打开,<br>1: 关闭          |
| 15 | 端口 3 上的通信<br>0: 无稳定通信,<br>1: 通信已建立 |

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

12.11 ST\_EcSlaveState

结构 ST\_EcSlaveState 包含 EtherCAT 状态和 EtherCAT 从站设备的链接状态。

```
TYPE ST_EcSlaveState:
STRUCT
    deviceState :BYTE;
    linkState   :BYTE;
END_STRUCT
END_TYPE
```

| 名称          | 类型   | 描述                                   |
|-------------|------|--------------------------------------|
| deviceState | BYTE | 从站的 EtherCAT 状态<br>(请参见 deviceState) |
| linkState   | BYTE | EtherCAT 从站的链接状态<br>(请参见 linkState)  |

### deviceState

从站的 EtherCAT 状态。该状态可以采用以下值之一：

| 常量                        | 值    | 描述            |
|---------------------------|------|---------------|
| EC_DEVICE_STATE_INIT      | 0x01 | Init (初始化) 状态 |
| EC_DEVICE_STATE_PREOP     | 0x02 | 准备运行状态        |
| EC_DEVICE_STATE_BOOTSTRAP | 0x03 | Bootstrap 状态  |
| EC_DEVICE_STATE_SAFEOP    | 0x04 | 安全运行状态        |
| EC_DEVICE_STATE_OP        | 0x08 | 运行状态          |

此外，还可以设置以下位：

| 常量                            | 值    | 描述                      |
|-------------------------------|------|-------------------------|
| EC_DEVICE_STATE_ERROR         | 0x10 | EtherCAT 从站内的状态机错误      |
| EC_DEVICE_STATE_INVALID_VPRS  | 0x20 | 供应商 ID、产品代码、修订版本号或序列号无效 |
| EC_DEVICE_STATE_INITCMD_ERROR | 0x40 | 发送初始化命令时出错。             |
| EC_DEVICE_STATE_DISABLED      | 0x80 | 从站已禁用                   |

### linkState

EtherCAT 从站的链接状态。链接状态可以由以下位的 ORing 组成：

| 常量                              | 值    | 描述                                                              |
|---------------------------------|------|-----------------------------------------------------------------|
| EC_LINK_STATE_OK                | 0x00 |                                                                 |
| EC_LINK_STATE_NOT_PRESENT       | 0x01 | 无法与 EtherCAT 从站进行 EtherCAT 通信                                   |
| EC_LINK_STATE_LINK_WITHOUT_COMM | 0x02 | 端口 X 处出错 (通过 EC_LINK_STATE_PORT_A/B/C/D 指定)。端口有链接，但无法通过该端口进行通信。 |
| EC_LINK_STATE_MISSING_LINK      | 0x04 | 端口 X 处链接缺失 (通过 EC_LINK_STATE_PORT_A/B/C/D 指定)。                  |
| EC_LINK_STATE_ADDITIONAL_LINK   | 0x08 | 端口 X 处附加链接 (通过 EC_LINK_STATE_PORT_A/B/C/D 指定)。                  |
| EC_LINK_STATE_PORT_A            | 0x10 | 端口 0                                                            |
| EC_LINK_STATE_PORT_B            | 0x20 | 端口 1                                                            |
| EC_LINK_STATE_PORT_C            | 0x40 | 端口 2                                                            |
| EC_LINK_STATE_PORT_D            | 0x80 | 端口 3                                                            |

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.12 ST\_EcSlaveStateBits

结构 ST\_EcSlaveStateBits 包含 EtherCAT 状态和 EtherCAT 从站设备的链接状态。

```

TYPE ST_EcSlaveStateBits:
STRUCT
    bInit          : BOOL;
    bPreop         : BOOL;
    bBootStrap     : BOOL;
    bSafeOp        : BOOL;
    bOp            : BOOL;
    bError         : BOOL;
    bInvVPRS       : BOOL;
    bInitCmdError  : BOOL;
    bLinkNotPresent : BOOL;
    bLinkWithoutComm : BOOL;
    bLinkMissing   : BOOL;
    bAdditionalLink : BOOL;
    bPortA         : BOOL;
    bPortB         : BOOL;
    bPortC         : BOOL;
    bPortD         : BOOL;
END_STRUCT
END_TYPE

```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.13 ST\_EcSlaveStateBitsEx

结构 ST\_EcSlaveStateBitsEx 包含 EtherCAT 状态和 EtherCAT 从站设备的链接状态。

```

TYPE ST_EcSlaveStateBitsEx:
STRUCT
    bInit          : BOOL;
    bPreop         : BOOL;
    bBootStrap     : BOOL;
    bSafeOp        : BOOL;
    bOp            : BOOL;
    bError         : BOOL;
    bInvVPRS       : BOOL;
    bInitCmdError  : BOOL;
    bDisabled      : BOOL;
    bLinkNotPresent : BOOL;
    bLinkWithoutComm : BOOL;
    bLinkMissing   : BOOL;
    bAdditionalLink : BOOL;
    bPortA         : BOOL;
    bPortB         : BOOL;
    bPortC         : BOOL;
    bPortD         : BOOL;
END_STRUCT
END_TYPE

```

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.14 ST\_PortAddr

结构 ST\_PortAddr 包含 EtherCAT 从站设备的 EtherCAT 拓扑信息。EtherCAT 从站设备通常有 2 至 4 个端口。

```

TYPE ST_PortAddr:
STRUCT
    portA : UINT;
    portB : UINT;
    portC : UINT;
    portD : UINT;
END_STRUCT
END_TYPE

```

| 名称    | 类型   | 描述                                       |
|-------|------|------------------------------------------|
| portA | UINT | 当前 EtherCAT 从站端口 A 处先前 EtherCAT 从站的地址    |
| portB | UINT | 当前 EtherCAT 从站端口 B 处可选的后续 EtherCAT 从站的地址 |
| portC | UINT | 当前 EtherCAT 从站端口 C 处可选的后续 EtherCAT 从站的地址 |
| portD | UINT | 当前 EtherCAT 从站端口 D 处可选的后续 EtherCAT 从站的地址 |

## 12.15 ST\_TopologyDataEx

结构 ST\_TopologyDataEx 包含有关 EtherCAT 拓扑和热连接组的信息。

```

TYPE ST_TopologyDataEx:
STRUCT
  nOwnPhysicalAddr : UINT;
  nOwnAutoIncAddr : UINT;
  stPhysicalAddr : ST_PortAddr;
  stAutoIncAddr : ST_PortAddr;
  aReserved1 : ARRAY [0..3] OF UDINT;
  nStatusBits : DWORD;
  nHCSlaveCountCfg : UINT; (*nStatusBits.0 = TRUE: DcSupprt;.1 = TRUE: DC64Supprt;.2=TRUE: Slave
Present following hot connect info requires runtime >= TC 2.11 R3 B2246 nStatusBits.3 = TRUE: HotCon
nectGroupStart;.4 = HotConnectSlave;.5 = TRUE: HotConnectInvalidB;.6 = TRUE: HotConnectInvalidC;
.7 = TRUE: HotConnectInvalidD*)
  nHCSlaveCountAct : UINT;
  aReserved2 : ARRAY [0..4] OF UDINT;
END_STRUCT
END_TYPE

```

| 名称               | 类型                    | 描述                                                                                                                                                                                                                                                                                             |
|------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nOwnPhysicalAddr | UINT                  | EtherCAT 从站设备的专用物理 EtherCAT 地址                                                                                                                                                                                                                                                                 |
| nOwnAutoIncAddr  | UINT                  | EtherCAT 从站设备的专用自动递增 EtherCAT 地址                                                                                                                                                                                                                                                               |
| stPhysicalAddr   | ST_PortAddr           | 端口 A...D 处 EtherCAT 从站设备的物理地址信息                                                                                                                                                                                                                                                                |
| stAutoIncAddr    | ST_PortAddr           | 端口 A...D 处 EtherCAT 从站设备的自动递增地址信息                                                                                                                                                                                                                                                              |
| aReserved1       | ARRAY [0..3] OF UDINT | 预留                                                                                                                                                                                                                                                                                             |
| nStatusBits      | DWORD                 | nStatusBits.0 = TRUE: 支持 DC 分布式时钟<br>nStatusBits.1 = TRUE: 支持分布式时钟 (64 位)<br>nStatusBits.2 = TRUE: 从站存在<br>nStatusBits.3 = TRUE: 从站是热连接组的起始节点<br>nStatusBits.4 = TRUE: 从站在热连接组中<br>nStatusBits.5 = TRUE: 热连接在端口 B 无效<br>nStatusBits.6 = TRUE: 热连接在端口 C 无效<br>nStatusBits.7 = TRUE: 热连接在端口 D 无效 |
| nHCSlaveCountCfg | UINT                  | 热连接组设备的配置数量                                                                                                                                                                                                                                                                                    |
| nHCSlaveCountAct | UINT                  | 找到的热连接组设备的数量                                                                                                                                                                                                                                                                                   |
| aReserved2       | ARRAY [0..4] OF UDINT | 预留                                                                                                                                                                                                                                                                                             |

### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.16 DCTIMESTRUCT

结构化 TwinCAT “分布式时钟系统时间” 格式。最小单位为纳秒。该数据类型表示自 2000 年 1 月 1 日（格林尼治标准时间）起的纳秒数。

```

TYPE DCTIMESTRUCT :
STRUCT
  wYear : WORD;
  wMonth : WORD;
  wDayOfWeek : WORD;
  wDay : WORD;

```

```

wHour      : WORD;
wMinute    : WORD;
wSecond    : WORD;
wMilliseconds : WORD;
wMicroseconds : WORD;
wNanoseconds : WORD;
END_STRUCT
END_TYPE

```

| 名称            | 类型   | 描述                               |
|---------------|------|----------------------------------|
| wYear         | WORD | 年份：2000 ~ 2584                   |
| wMonth        | WORD | 月份：1 ~ 12（一月 = 1，二月 = 2 等）       |
| wDayOfWeek    | WORD | 星期中的某一天：0 ~ 6（星期日 = 0，星期一 = 1 等） |
| wDay          | WORD | 月份中的某一天：1 ~ 31                   |
| wHour         | WORD | 小时：0 ~ 23                        |
| wMinute       | WORD | 分钟：0 ~ 59                        |
| wSecond       | WORD | 秒：0 ~ 59                         |
| wMilliseconds | WORD | 毫秒：0 ~ 999                       |
| wMicroseconds | WORD | 微秒：0 ~ 999                       |
| wNanoseconds  | WORD | 纳秒：0 ~ 999                       |

#### 前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

## 12.17 T\_DCTIME32

32 位 TwinCAT 分布式时钟系统时间格式。最小单位为纳秒。

该 32 位 DC 系统时间由完整的绝对 64 位 DC 系统时间 (T\_DCTIME [▶ 118]) 构成，只使用最低阶的 32 位。这意味着已失去绝对唯一时间的特性，并且假定该 32 位时间仅在当前系统时间周围 ± 2.147 秒的狭窄时间窗口内使用，以确保准确无误。在许多应用中，这一假设均可行。

如果违反这一假设，那么对该时间的解释和进一步处理可能会出现错误。

```

TYPE T_DCTIME32 : UDINT;
END_TYPE

```

#### 前提条件

| 开发环境           | 目标平台                  | 需要的 PLC 库    |
|----------------|-----------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX（x86、x64、Arm®） | Tc2_EtherCAT |

## 12.18 T\_DCTIME64

数据类型 T\_DCTIME64 可将分布式时钟系统时间（简称 DC 时间）表示为线性 64 位整数值。该时间以纳秒为单位，从 UTC 2000 年 1 月 1 日开始计算。最小单位为纳秒。

```

TYPE T_DCTIME64 : ULINT;
END_TYPE

```

| 有用的分布式时钟系统时间常量             | 描述                                                                                 |
|----------------------------|------------------------------------------------------------------------------------|
| EC_DCTIME_DELTA_OFFSET64   | 1601 年 1 月 1 日和 2000 年 1 月 1 日之间的 100 纳秒刻度数。<br>此为“Windows 文件时间”和“分布式时钟系统时间”之间的差值。 |
| EC_DCTIME_DATEDELTA_OFFSET | 零年到 2000 年 1 月 1 日之间经过的天数                                                          |
| EC_DCTIME_TICKSPERMSEC64   | 每毫秒分布式时钟系统时间纳秒数                                                                    |
| EC_DCTIME_TICKSPERSEC64    | 每秒分布式时钟系统时间纳秒数                                                                     |
| EC_DCTIME_TICKSPERDAY64    | 每天分布式时钟系统时间纳秒数                                                                     |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.19 T\_DCTIME



### 过时的数据类型

该数据类型已经过时。使用数据类型 [T\\_DCTIME64 \[► 117\]](#) 代替。

数据类型 `T_DCTIME` 可将分布式时钟系统时间（简称 DC 时间）表示为线性 64 位整数值。时间用自 UTC 2000 年 1 月 1 日起的纳秒表示。

数据类型用两个 32 位 DWORD 变量表示，以便于在 PLC 中进行处理。可使用 Tc2\_Uilities 库中的 ui64 功能执行运算（时间加法和时间减法）。

```
TYPE T_DCTIME : T_ULARGE_INTEGER;
END_TYPE
```

| 有用的分布式时钟系统时间常量             | 描述                                                                                 |
|----------------------------|------------------------------------------------------------------------------------|
| EC_DCTIME_DELTA_OFFSET     | 1601 年 1 月 1 日和 2000 年 1 月 1 日之间以 100 纳秒为单位的数值。<br>此为 Windows 文件时间与分布式时钟系统时间之间的差值。 |
| EC_DCTIME_DATEDELTA_OFFSET | 零年到 2000 年 1 月 1 日之间经过的天数                                                          |
| EC_DCTIME_TICKSPERMSEC     | 每毫秒分布式时钟系统时间纳秒数                                                                    |
| EC_DCTIME_TICKSPERSEC      | 每秒分布式时钟系统时间纳秒数                                                                     |
| EC_DCTIME_TICKSPERDAY      | 每天分布式时钟系统时间纳秒数                                                                     |

#### 前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 12.20 T\_HFoe

“File access over EtherCAT” 路径。在使用该途径之前，必须使用功能块 [FB\\_EcFoeOpen \[► 61\]](#) 对其进行一次初始化。不得直接写入这种结构化类型的变量。

```
TYPE T_HFoe :
STRUCT
  sNetID : T_AmsNetId := '';
  nPort : T_AmsPort := 0;
  handle : UDINT := 0;
  eMode : E_EcFoeMode := eFoeMode_Write;
END_STRUCT
END_TYPE
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 13 常量

### 13.1 全局常量

#### VAR\_GLOBAL 常量

```

EC_AMSPORT_MASTER :UINT :=16#FFFF;
EC_MAX_SLAVES      :UINT :=16#FFFF;

(*ethercat commands*)
EC_CMD_TYPE_APRD :BYTE :=1;
EC_CMD_TYPE_APWR :BYTE :=2;
EC_CMD_TYPE_APRW :BYTE :=3;
EC_CMD_TYPE_FPRD :BYTE :=4;
EC_CMD_TYPE_FPWR :BYTE :=5;
EC_CMD_TYPE_FPRW :BYTE :=6;
EC_CMD_TYPE_BRD  :BYTE :=7;
EC_CMD_TYPE_BWR  :BYTE :=8;
EC_CMD_TYPE_BRW  :BYTE :=9;
EC_CMD_TYPE_LRD  :BYTE :=10;
EC_CMD_TYPE_LWR  :BYTE :=11;
EC_CMD_TYPE_LRW  :BYTE :=12;

(* Device states *)
EC_DEVICE_STATE_MASK :BYTE :=16#0F;
EC_DEVICE_STATE_INIT :BYTE :=16#01;
EC_DEVICE_STATE_PREOP :BYTE :=16#02;
EC_DEVICE_STATE_BOOTSTRAP :BYTE :=16#03;
EC_DEVICE_STATE_SAFEOP :BYTE :=16#04;
EC_DEVICE_STATE_OP :BYTE :=16#08;
EC_DEVICE_STATE_ERROR :BYTE :=16#10;
EC_DEVICE_STATE_INVALID_VPRS :BYTE :=16#20;
EC_DEVICE_STATE_INITCMD_ERROR :BYTE :=16#40;

(* Link states *)
EC_LINK_STATE_OK :BYTE :=16#00;
EC_LINK_STATE_NOT_PRESENT :BYTE :=16#01;
EC_LINK_STATE_LINK_WITHOUT_COMM :BYTE :=16#02;
EC_LINK_STATE_MISSING_LINK :BYTE :=16#04;
EC_LINK_STATE_ADDITIONAL_LINK :BYTE :=16#08;
EC_LINK_STATE_PORT_A :BYTE :=16#10;
EC_LINK_STATE_PORT_B :BYTE :=16#20;
EC_LINK_STATE_PORT_C :BYTE :=16#40;
EC_LINK_STATE_PORT_D :BYTE :=16#80;

(* Device/Link state IG/IO *)
EC_ADS_IGRP_MASTER_STATEMACHINE :UDINT :=16#00000003;
EC_ADS_IOFFS_MASTER_CURSTATE :UDINT :=16#00000100;
EC_ADS_IOFFS_MASTER_REQSTATE :UDINT :=16#00000101;
EC_ADS_IOFFS_MASTER_INTERNALSTATE :UDINT :=16#00000102;

EC_ADS_IGRP_MASTER_COUNT_SLAVE :UDINT :=16#00000006;
EC_ADS_IOFFS_MASTER_COUNT_SLAVE :UDINT :=16#00000000;
EC_ADS_IOFFS_MASTER_COUNT_PORT :UDINT :=16#00000001;
EC_ADS_IOFFS_MASTER_COUNT_ROUTER :UDINT :=16#00000002;

EC_ADS_IGRP_MASTER_SLAVE_ADDRESSES :UDINT :=16#00000007;
EC_ADS_IGRP_MASTER_SENDCMD :UDINT :=16#00000008;
EC_ADS_IGRP_SLAVE_STATEMACHINE :UDINT :=16#00000009;
EC_ADS_IGRP_MASTER_SLAVE_IDENTITY :UDINT :=16#00000011;
EC_ADS_IGRP_MASTER_SLAVE_CRC :UDINT :=16#00000012;
EC_ADS_IGRP_MASTER_SLAVE_ABNORMAL_STATE_CHANGES :UDINT :=16#00000013;
EC_ADS_IGRP_MASTER_SLAVE_SETPRESENT_CHANGES :UDINT :=16#00000016;
EC_ADS_IGRP_MASTER_DEVICESTATE :UDINT :=16#00000045;
EC_ADS_IGRP_MASTER_COUNT_FRAME :UDINT :=16#00000048;

(* SoE IG/IO *)
EC_ADS_IGRP_ECAT_SOE :UDINT :=16#0000F420;
EC_ADS_IGRP_ECAT_SOE_LASTERROR :UDINT :=16#0000F421;

EC_SOE_ELEMENT_DATASTATE :BYTE :=16#01;
EC_SOE_ELEMENT_NAME :BYTE :=16#02;
EC_SOE_ELEMENT_ATTRIBUTE :BYTE :=16#04;
EC_SOE_ELEMENT_UNIT :BYTE :=16#08;
EC_SOE_ELEMENT_MIN :BYTE :=16#10;
EC_SOE_ELEMENT_MAX :BYTE :=16#20;
EC_SOE_ELEMENT_VALUE :BYTE :=16#40;
EC_SOE_ELEMENT_DEFAULT :BYTE :=16#80;

(* FoE IG/IO *)
EC_ADS_IGRP_FOE_FOPENREAD :UDINT :=16#0000F401;
EC_ADS_IGRP_FOE_FOPENWRITE :UDINT :=16#0000F402;
EC_ADS_IGRP_FOE_FCLOSE :UDINT :=16#0000F403;
EC_ADS_IGRP_FOE_FREAD :UDINT :=16#0000F404;
EC_ADS_IGRP_FOE_FWRITE :UDINT :=16#0000F405;
EC_ADS_IGRP_FOE_PROGRESSINFO :UDINT :=16#0000F406;
EC_ADS_IGRP_FOE_LASTERROR :UDINT :=16#0000F407;

```



```
(* CoE IG/IO *)
EC_ADS_IGRP_CANOPEN_SDO :UDINT :=16#0000F302;
EC_ADS_IGRP_CANOPEN_SDO_LASTERROR :UDINT :=16#0000F303;

EC_DCTIME_DATEDELTA_OFFSET : DWORD := 730120; (* Number of past days since year zero until 1 January 2000 *)
EC_DCTIME_DELTA_OFFSET : T_ULARGE_INTEGER := ( dwHighPart := 16#01BF53EB, dwLowPart := 16#256D4000 )
; (* Number of 100ns ticks between 1.1.1601 and 1.1.2000 *)
EC_DCTIME_TICKSPERMSEC : T_ULARGE_INTEGER := ( dwHighPart := 16#00000000, dwLowPart := 16#000F4240);
(* Number of nanosecond ticks per millisecond *)
EC_DCTIME_TICKSPERSEC : T_ULARGE_INTEGER := ( dwHighPart := 16#00000000, dwLowPart := 16#3B9ACA00);
(* Number of nanosecond ticks per second *)
EC_DCTIME_TICKSPERDAY : T_ULARGE_INTEGER := ( dwHighPart := 16#00004E94, dwLowPart := 16#914F0000);
(* Number of nanosecond ticks per day *)

EC_DCTIME_DELTA_OFFSET64 : ULINT := ULINT#16#01BF53EB_256D4000;
(* Number of 100ns ticks between 1.1.1601 and 1.1.2000 *)
EC_DCTIME_TICKSPERMSEC64 : ULINT := ULINT#16#00000000_000F4240;
(* Number of nanosecond ticks per millisecond *)
EC_DCTIME_TICKSPERSEC64 : ULINT := ULINT#16#00000000_3B9ACA00;
(* Number of nanosecond ticks per second *)
EC_DCTIME_TICKSPERDAY64 : ULINT := ULINT#16#00004E94_914F0000;
(* Number of nanosecond ticks per day *)

bSeqReadDrvAttrAndValue : BOOL := FALSE;
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

13.2 库版本

所有库都有特定版本。例如，版本会在 PLC 库中显示。全局常量包含有关库版本的信息：

Global\_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_EtherCAT : ST_LibVersion;
END_VAR
```

**stLibVersion\_Tc2\_EtherCAT**：Tc2\_EtherCAT 库的版本信息（类型：ST\_LibVersion）

要检查您拥有的版本是否是所需的版本，请使用功能 F\_CmpLibVersion（在 Tc2\_System 库中定义）。



您可能从 TwinCAT 2 中了解到的用于比较库版本的所有其他选项都已过时！

13.3 EtherCAT 邮箱协议错误代码

VAR\_GLOBAL 常量

```
(* FoE mailbox protocol error codes *)
EC_FOE_PROTERR_NOTDEFINED : UDINT := 0;
EC_FOE_PROTERR_NOTFOUND : UDINT := 1;
EC_FOE_PROTERR_ACCESS : UDINT := 2;
EC_FOE_PROTERR_DISKFULL : UDINT := 3;
EC_FOE_PROTERR_ILLEAGAL : UDINT := 4;
EC_FOE_PROTERR_PACKENO : UDINT := 5;
EC_FOE_PROTERR_EXISTS : UDINT := 6;
EC_FOE_PROTERR_NOUSER : UDINT := 7;
EC_FOE_PROTERR_BOOTSTRAPONLY : UDINT := 8;
EC_FOE_PROTERR_NOTINBOOTSTRAP : UDINT := 9;
EC_FOE_PROTERR_INVALIDPASSWORD : UDINT := 10;

(* CoE mailbox protocol error codes *)
EC_COE_PROTERR_TOGGLE : UDINT := 16#05030000; (* Toggle bit not alternated. *)
EC_COE_PROTERR_TIMEOUT : UDINT := 16#05040000; (* SDO protocol timed out. *)
EC_COE_PROTERR_CCS_SCS : UDINT := 16#05040001; (* Client/
server command specifier not valid or unknown. *)
EC_COE_PROTERR_BLK_SIZE : UDINT := 16#05040002; (* Invalid block size (block mode only). *)
EC_COE_PROTERR_SEQNO : UDINT := 16#05040003; (* Invalid sequence number (block mode only). *)
EC_COE_PROTERR_CRC : UDINT := 16#05040004; (* CRC error (block mode only). *)
EC_COE_PROTERR_MEMORY : UDINT := 16#05040005; (* Out of memory. *)
EC_COE_PROTERR_ACCESS : UDINT := 16#06010000; (* Unsupported access to an object. *)
```

```
EC_COE_PROTERR_WRITEONLY : UDINT := 16#06010001; (* Attempt to read a write only object. *)
EC_COE_PROTERR_READONLY : UDINT := 16#06010002; (* Attempt to write a read only object. *)
EC_COE_PROTERR_INDEX : UDINT := 16#06020000; (* Object does not exist in the object dictionary. *)
EC_COE_PROTERR_PDO_MAP : UDINT := 16#06040041; (* Object cannot be mapped to the PDO. *)
EC_COE_PROTERR_PDO_LEN : UDINT := 16#06040042; (* The number and length of the objects to be mapped would exceed PDO length. *)
EC_COE_PROTERR_P_INCOMP : UDINT := 16#06040043; (* General parameter incompatibility reason. *)
EC_COE_PROTERR_I_INCOMP : UDINT := 16#06040047; (* General internal incompatibility in the device. *)
EC_COE_PROTERR_HARDWARE : UDINT := 16#06060000; (* Access failed due to an hardware error. *)
EC_COE_PROTERR_DATA_SIZE : UDINT := 16#06070010; (* Data type does not match, length of service parameter does not match *)
EC_COE_PROTERR_DATA_SIZE1 : UDINT := 16#06070012; (* Data type does not match, length of service parameter too high *)
EC_COE_PROTERR_DATA_SIZE2 : UDINT := 16#06070013; (* Data type does not match, length of service parameter too low *)
EC_COE_PROTERR_OFFSET : UDINT := 16#06090011; (* Sub-index does not exist. *)
EC_COE_PROTERR_DATA_RANGE : UDINT := 16#06090030; (* Value range of parameter exceeded (only for write access). *)
EC_COE_PROTERR_DATA_RANGE1 : UDINT := 16#06090031; (* Value of parameter written too high. *)
EC_COE_PROTERR_DATA_RANGE2 : UDINT := 16#06090032; (* Value of parameter written too low. *)
EC_COE_PROTERR_MINMAX : UDINT := 16#06090036; (* Maximum value is less than minimum value. *)
EC_COE_PROTERR_GENERAL : UDINT := 16#08000000; (* general error *)
EC_COE_PROTERR_TRANSFER : UDINT := 16#08000020; (* Data cannot be transferred or stored to the application. *)
EC_COE_PROTERR_TRANSFER1 : UDINT := 16#08000021; (* Data cannot be transferred or stored to the application because of local control. *)
EC_COE_PROTERR_TRANSFER2 : UDINT := 16#08000022; (* Data cannot be transferred or stored to the application because of the present device state. *)
EC_COE_PROTERR_DICTIONARY : UDINT := 16#08000023; (* Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error). *)
```

前提条件

| 开发环境           | 目标平台                   | 需要的 PLC 库    |
|----------------|------------------------|--------------|
| TwinCAT v3.1.0 | PC 或 CX (x86、x64、Arm®) | Tc2_EtherCAT |

## 14 示例

用于诊断的示例项目和示例配置

请参见 [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc2\\_EtherCAT/Resources/2364613387.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_EtherCAT/Resources/2364613387.zip)

## Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar®, and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

## Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
电话号码: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)