



## EtherCAT 主站软件开发指南

发布版本：V1.7

发布日期：2023.8.14

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

### 注意

您购买的产品、服务或特性等应受商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 目录

### 目录

EtherCAT 主站软件开发指南 .....	1
目录 .....	2
一、EtherCAT 简介 .....	3
前言 .....	3
背景 .....	3
概述 .....	4
EtherCAT 协议原理 .....	7
Ethercat 应用场景 .....	8
二、EtherCAT 主站 .....	9
2.1 产品概述 .....	9
2.2 EtherCAT 主站整体框架 .....	11
2.3 EtherCAT 套件交互 .....	12
2.4 搭建二次开发环境 .....	12
2.5 接口函数说明 .....	17
1、FPGA 模块说明（如用不上请忽略） .....	17
2、FPGA 寄存器 .....	18
3、MCU 控制函数说明 .....	24
三、生成 XTI 操作步骤 .....	27
四、常见问题解答 .....	31

# 一、EtherCAT 简介

## 前言

本文档介绍 EtherCAT 主站二次开发的简介与要求、如何搭建二次开发环境，以及 API 函数、用例程序和常见问题处理方法。为用户对 EtherCAT 进行二次开发，提供参考与说明。

## 读者对象

本文档主要适用于以下工程师：

- 软件工程师
- 系统测试工程师
- FPGA 工程师

## 背景

EtherCAT（以太网控制自动化技术）是一个开放架构，基于以太网为基础的现场总线系统，其名称的 CAT 为控制自动化技术

（Control Automation Technology）字首的缩写。EtherCAT 是确定性的工业以太网，最早是由德国的 Beckhoff 公司研发。它于 2003 年被引入市场，于 2007 年成为国际标准，并于 2014 年成为中国国家标准。EtherCAT 的出现为系统的实时性能和拓扑的灵活性树立了新的标准。

自动化对通讯一般会要求较短的资料更新时间（或称为周期时间）、资料同步时的通讯抖动量低，而且硬件的成本要低，EtherCAT 开发的目的就是让以太网可以运用在自动化应用中。

## 概述

在工业自动化领域，伺服运动控制系统作为机器人及数控设备的核心组成部分，对其相关技术的研究尤为重要。目前各领域技术的进步，促使工业设备需求不断提高，传统伺服控制系统中使用的工业现场总线受限于老旧的协议技术及硬件规格，导致其数据传输速度、实时性等方面越来越难以满足逐渐提高的系统总线数据传输需求。

因此开发一种更加优良、性能更加符合标准的工业通信技术成为了自动化通信领域的人们迫切需求。

在众多通信技术中，曾用于计算机通信领域的以太网技术引起了人们的注意，Ethernet 拥有易于连接，连通方便的优点，并且价格也较为理想，传输速度可以达到工业控制所需要的标准，这些优点使它成为了用于开发工业网络的首选。通用的工业以太网定义为：技术上与商用以太网兼容，但在产品设计上必须满足工业现场对实时性、可靠性、可互操作性、抗干扰性、本质安全性、环境适应性等方面的需要，是继现场总线之后发展起来的、被广泛认同为颇具发展前景的一种工业通信网络。工业以太网采用 TCP /IP 协议，和 IEEE 802.3 标准兼容，但在应用层会加入各自特有的协议（通常为 IEEE 802.3/IEEE 802.3u）[2]。

相对于 IT 和办公应用中的硬件成本而言，工业自动化的硬件成本更加重要。标准以太网网络几乎无法满足以上需求的现场级应用。如果每个节点使用一个独立的以太网报文传输几个字节的周期性过程数据，那么有效数据利用率会明显下降。因为以太网报文的最短长度为 84 字节（包括帧间距），其中的 46 个字节可以用于过程数据。例如，一个驱动器发送 4 字节的实际位置和状态信息过程数据，同时接收 4 字节的目标位置和控制字信息数据，则发送/接收报文的有效数据利用率下降到 4.8%（4/84）。另外，驱动器通常在接收到目标值后触发传输实际值需要一定的响应时间。最终，100 Mbit/s 的带宽所剩无几。

而在 IT 领域通常使用的路由 (IP) 和连接 (TCP) 协议栈需要为每个节点使用附加的协议头, 会产生进一步的延时。

与商用以太网相比, 工业以太网的技术特征体现在对以下方面的特殊要求:

- ①具有高实时性与良好的时间确定性。
- ②传送信息多为短帧信息, 且信息交换频繁。
- ③容错能力强, 可靠性、安全性好。
- ④控制网络结构具有高度分散性。
- ⑤控制网络协议简单、实用, 工作效率高。
- ⑥控制设备的智能化与控制功能的自治性。
- ⑦与信息网络之间有高效率的通信, 易于实现与信息网络的集成。
- ⑧设备的可靠性与环境适应性。
- ⑨远距离传输。
- ⑩总线供电。

工业以太网有三种实现方式: TCP/IP 方式、以太网方式、修改以太网方式。

其中, TCP/IP 的方式仍然采用传统的 TCP/IP 协议栈进行通信, 通过上层合理调度减少数据传输过程中的不确定性, 使用这种方式有 Modbus/TCP 和 Ethernet/IP 等协议这种方式的数据传输实时性不高; 以太网的方式采用标准的以太网设备, 传输普通的以太网数据仍然可以使用 TCP/IP 协议, 而用于传输控制信号的过程数据则使用专门的协议传输, 使用这种方式的有 Ethernet Powerlink、PROFINet RT 和中国的 EPA (Ethernet for Plant Automation) 等协议, 可以实现较高的实时性; 修改以太网的方式采用经过修改的以太网协议传输数据, 而使用专门的硬件处理数据, 使得响应时间小于 1ms, 它的实时数据和非实时数据也是分开传输的, 彻底避免数据报文冲突, 使用这种方式的有 SERCOS-III、PROFINet IRT 和 EtherCAT 等协议,

**EtherCAT** 技术是由德国 BECKHOFF 自动化公司提出并实现的工业以太网技

术，是目前最快的工业以太网解决方案。

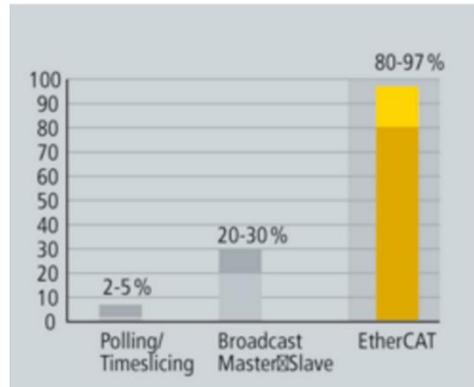


图 2

一个完整的 EtherCAT 系统可以分为主站和从站两个部分，主站使用标准以太网设备传输数据，从站使用专用的控制芯片处理数据，支持线形、树形或星形等多种拓扑结构，具有超高的性能、灵活性和成本优势，EtherCAT 协议与普通的现场总线协议相比有以下特点：

(1) EtherCAT 具有广泛的适应性，无论是简单的 16 位微处理器还是复杂的 PC 系统，只要控制单元带有普通的以太网控制器都可以构成 EtherCAT 主站。

(2) EtherCAT 是一种特殊的以太网协议，使用该协议的数据可以使用常用的以太网设备，可以节省设备更新的成本。

(3) EtherCAT 可以灵活选择从站类型，无论是带有微处理器的复杂节点还是只有 2 位 I/O 的简单节点可以用作 EtherCAT 从站。

(4) EtherCAT 数据传输速率高，由 (2) 知该协议数据符合标准的以太网协议，可以通过标准的以太网进行数据传输，可以充分利用以太网带宽进行用户数据的传输。

(5) EtherCAT 数据刷新周期短，使用专门的硬件处理数据，数据的刷新周期很小，低于  $100 \mu s$ ，可用于响应要求高的设备。

(6) EtherCAT 具有良好的同步性能，设备之间的同步由专用硬件的相关寄存器完成，各个从站设备之间的时钟同步精度可以控制在  $1 \mu s$  以内。

于是以 EtherCAT 为典型的基于 Ethernet 面向工业传输的各种各样的工业

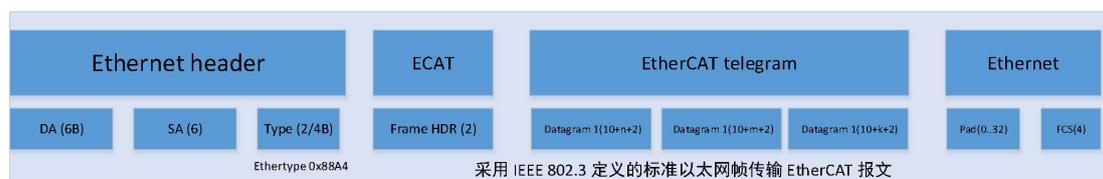
以太网技术被研发出来，是当今最流行的工业以太网标准之一。

## EtherCAT 协议原理

一个 EtherCAT 数据帧足以完成所有节点控制数据的发送和接收，这种高性能的运行模式克服了前面章节描述的各种问题。EtherCAT 主站发送一个报文，报文经过所有节点。EtherCAT 从站设备高速动态地（on the fly）读取寻址到该节点的数据，并在数据帧继续传输的同时插入数据。这样，数据帧的传输延时只取决于硬件传输延时。当某一网段或分支上的最后一个节点检测到开放端口（无下一个从站）时，利用以太网技术的全双工特性，将报文返回给主站。

EtherCAT 报文的最大有效数据利用率达 90% 以上，而由于采用全双工特性，有效数据利用率理论上高于 100 MBit/s。EtherCAT 主站是网段内唯一能够主动发送 EtherCAT 数据帧的节点，其他节点仅传送数据帧。这一设想是为了避免不可预知的延时，从而保证 EtherCAT 的实时性能。EtherCAT 主站采用标准的以太网介质访问控制器（MAC），无需额外的通信处理器。因此，任何集成了以太网接口的硬件平台都可以实现 EtherCAT 主站，而与所使用的实时操作系统或应用软件无关。EtherCAT 从站设备的 EtherCAT 从站控制器（ESC）负责在硬件中高速动态地（on the fly）处理 EtherCAT 数据帧，不仅使网络性能可预测，而且其性能独立于具体的从站设备实施方式。

EtherCAT 将其报文嵌入到标准的以太网数据帧中（形成 EtherCAT 数据帧）。设备通过帧类型 0x88A4 识别 EtherCAT 数据帧。由于 EtherCAT 协议被优化为适用于短周期性的过程数据，因此无需庞大的协议堆栈，例如 TCP/IP 或 UDP/IP。



•

为了保证节点之间的 IT 通信，TCP/IP 可选择性地通过邮箱通道传输，从而不影响实时数据的传输。在启动期间，EtherCAT 主站设备为从站设备配置并映射过程数据。主站与从站之间交换的数据量可以各不相同，从一个位到几个字节，甚至是几 KB。EtherCAT 数据帧包含一个或多个 EtherCAT 子报文，子报文头标明了主站设备的访问

方式：

读，写，或读-写

通过直接寻址访问指定的从站设备，或通过逻辑寻址访问多个从站设备（隐式寻址）逻辑寻址方式主要用于周期性交换的过程数据。每个报文定位到 EtherCAT 网段中过程映像的具体位置，过程映像具有 4GB 的地址空间。网络启动阶段，在全局地址空间中，为每个从站分配一个或多个地址。如果多个从站设备被分配到了相同的地址域，那么可通过单个报文对其进行寻址。由于报文中包含了所有的数据访问相关信息，因此主站可决定何时对哪些数据进行访问。例如，主站设备可以使用短循环周期刷新驱动器中数据，长循环周期采样 I/O 端口，固定的过程数据结构不是必要的。这也使得 EtherCAT 主站设备相较于传统的现场总线系统减轻了负担，在传统的现场总线系统中，主站需要单独读取每个节点

## Ethercat 应用场景

### 1.运动控制器/运动控制卡

脉冲型升级为 EtherCAT 总线系统，提高性能和降低成本。

应用在多轴控制系统中，大幅度降低成本和简约电气布线。

分布式从站节点，搭配自由方便灵活。

### 2.数控系统 CNC

脉冲型升级为 EtherCAT 总线系统，增加系统稳定性、提高性能和降低成本。

摆脱由于传统脉冲频率的限制，体现高精高速的性能需求。

### 3.机器人/机械手臂等多轴高精度控制系统

降低 CPU 使用率，让 CPU 更加出色做高阶算法。

电气简单，控制稳定。

分布式从站节点，搭配自由方便灵活。

### 4.工控主板

支持目前主流现场工业以太网总线，提高性能，增加其附加值。

### 5.现场工业总线协议转换

EtherCAT 协议转换到其他协议总线方便快捷。

## 二、EtherCAT 主站

### 2.1 产品概述

使用紫光 FPGA 逻辑实现 EtherCAT 协议，实现主站 DC 功能。更加突出了 EtherCAT 现场总线的同步性能及高效性，大量减轻处理器的负担，CPU 只需要专注于计算，无需关注 ethercat 发包。只需要将数据通过 FSMC 协议写入 FPGA。FPGA 会把数据打包成 EtherCAT 格式。通过网络发送给从站。

如图所示是 EtherCAT 主站样卡图 2.1.1 是正面图，图 2.1.2 是反面图。



图 2.1.1

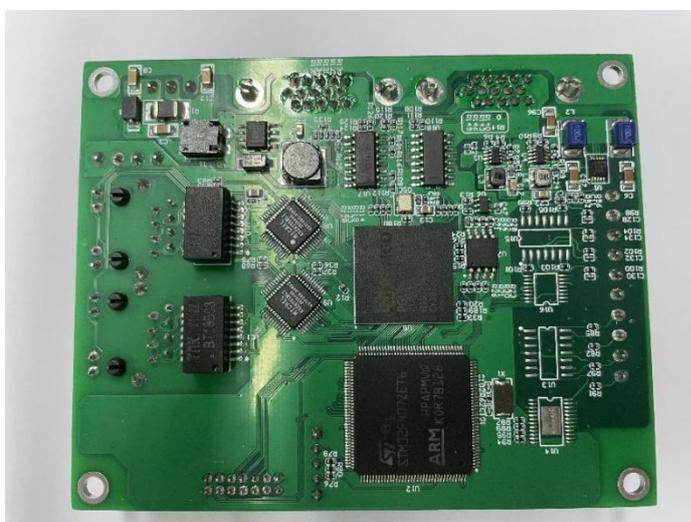


图 2.1.2

EtherCAT 主站实现了标准的 EtherCAT 协议，采用通用的 Memory 接口，可以连接任何 CPU 对 IP 进行控制。可用于任何标准的 EtherCAT 电机、IO 控制完全按照 EtherCAT 协议规范进行开发，整个报文的组装和解析完全在 FPGA 内进行。

对于 cpu 来说，完全可以把 FPGA 看成一个 SRAM 设备。Cpu 和 IP 进行数据的通信，会把这些报文按照一定的要求组装成 EtherCAT 报文发送出去。这样可以大大简化 cpu 侧软件的开发。DC 同步以及重试、线缆冗余等功能也直接在 FPGA 内完成。这样可以减轻 CPU 侧的负担，CPU 只需专注于上层软件的开发，无需再关心底层的通信。

我们在 FPGA 主站中加入了专用的同步模块，依赖与 FPGA 精确的计时和该专用同步模块，可以保证主站和从站之间的精确同步。目前主从之间的同步可以做到远小于 1us。

对于一个安全性要求高的系统，冗余功能是必须的。这样可以保证即使两个从站之间发生断线，整个系统也可以继续通信工作。相对于软主站冗余功能，FPGA 主站可以做到更快的通信恢复，而且可做到断线后所有的从站继续保持同步。

#### **产品特点：**

- 1.精确的主站 DC，从站同步输出达到 40 纳秒的同步精度。
2. 周期通信发送抖动不受处理器中断性能的影响，40 纳秒低抖动周期通信使处于 free run 模式下的从站更加稳定可靠。
3. 实现和缓存一个周期的数据，减轻处理器负担，让处理器有更多的时间使用在更高层次的算法等应用上或使用同等的处理器开发出更高性能的产品。
4. 同步性和实时性不受操作系统性能影响，甚至没操作系统的限制，更可无操作系统的支持，适用性更加广泛和易用。
5. 用户应用场景需要 FPGA 时，可在 FPGA 加入主站功能（主站只占用 4K 逻辑资源），多余资源可以用作所需业务控制

## 2.2 EtherCAT 主站整体框架

采用紫光 FPGA 和 MCU 配合完成 EtherCAT 主站功能，如图 2.2.1 所示。MCU 通过 FSMC 协议和 FPGA 进行交互，可以把 FPGA 看成是一个 SRAM 设备。MCU 读写内存等时，会把内存相应数据通过 FSMC 协议发给 FPGA。

FPGA 拿到数据后对数据进行 EtherCAT 协议组包、解包，并把数据通过网口发给 EtherCAT 从站。和传统主站相比 MCU 无需关心 EtherCAT 协议，MCU 只需要关注业务运算。无需关心 EtherCAT 协议组包和解析。

由 MCU 设置 FPGA 参数，让 FPGA 周期性出发中断告知 MCU 进行业务处理并且把数据通过 FSMC 协议和 FPGA 进行交互。

对于协议处理而言 FPGA 并行处理架构更加适用。大大提高了 MCU 运行效率。而对于同步实时性 FPGA 更加有保证。

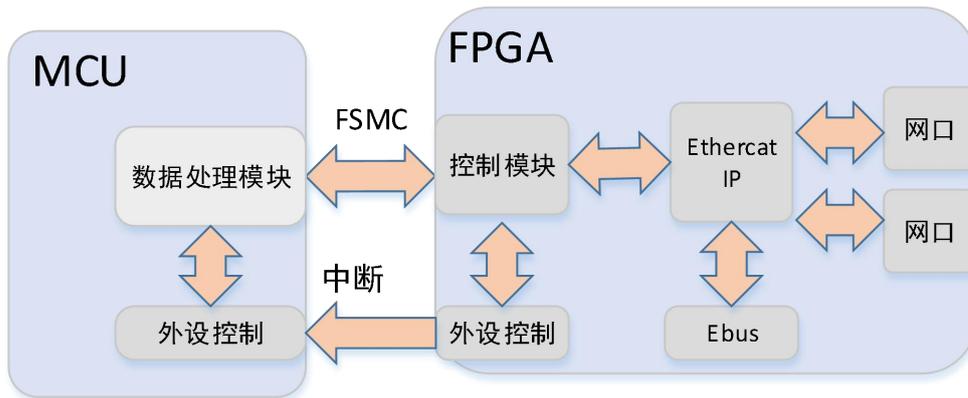


图 2.2.1

EtherCAT 主站接入网络连线图如图 2.2.2 所示，一个 EtherCAT 主站可以控制多个从站。只需要把运算好数据根据接口函数写入 FPGA 中。FPGA 会把数据通过网口同步给各个从站，同时也会把从站反馈的数据进行解包。存放在 FPGA 内部 RAM 当中。软件只需通过给到函数接口读取即可。

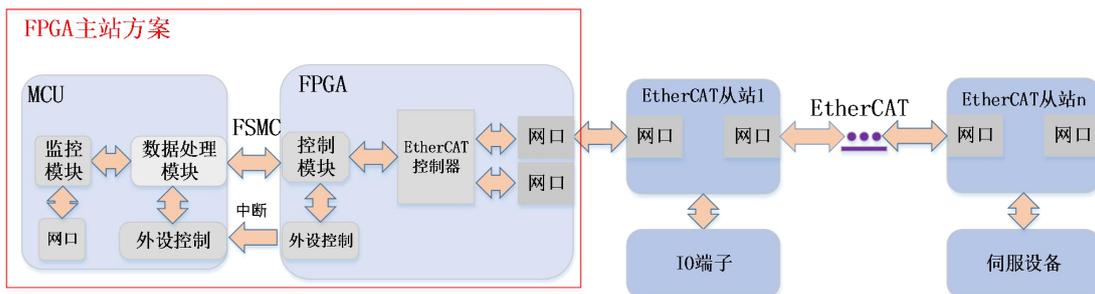


图 2.2.2

在 EtherCAT 主站中 FPGA 主要工作如下：

PDO 实时发送、 DC 同步功能、 EtherCAT 协议组包、发包、解包

## 2.3 EtherCAT 套件交互

**硬件：**

1. 主站开发板 PGL25G（27K LUT4）一块
2. 主站参考原理图及 PCB 一份

**软件：**

- 1.MCU 参考代码一份
2. EtherCAT Master IP 网表文件
- 3.FPGA 参考工程

**文档：**

EtherCAT 主站软件开发指南一份

## 2.4 搭建二次开发环境

### 1、运行环境搭建

**前提条件**

24V 电源、 网线、 EtherCAT 从站、 win7 系统以上电脑一台

**注意事项**

网口有两个，实际业务只需要一个网口，另外一个网口是预留，切勿接错

## 操作步骤

如下图 2.3.1 所示：

1. 首先将 EtherCAT 主站数据输出网口通过网线连接到从站模块
- 2 图中 24V 电源接口处，接入 24V 电源。EtherCAT 从站根据对应电源链路接入
- 3.上电即可完成整个运行环境搭建

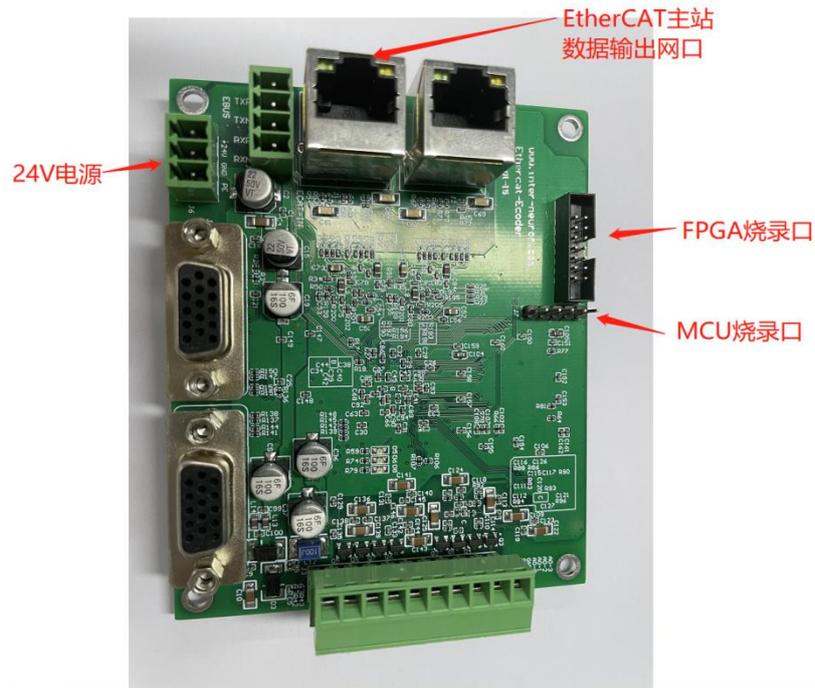


图 2.4.1

## 2、开发环境搭建

EtherCAT 主站出厂自带 MCU 和 FPGA 程序，当前 MCU 里面内置汇川 PLC、松下伺服控制。用户也可以直接接入两者可直接运行起来。

## 1.MCU 软件运行环境搭建

下面是百度云链接，安装 keil 软件及 MCU 补丁包。安装完成后才可以打开 MCU 工程。可以在提供参考工程上进行业务开发。

链接：<https://pan.baidu.com/s/1EXdQivAb30STZp6WEc04SQ?pwd=1234>

提取码：1234

## 2.FPGA 开发软件安装

由于主站使用的是紫光同创 FPGA，当用户需要对 FPGA 进行业务开发时，涉及使用紫光同创 FPGA 软件及 license。请联系人员。申请 FPGA 开发软件的 license 及软件使用支持。

出厂板卡 FPGA 中内置 EtherCAT 主站程序，如果用户不涉及 FPGA 或者不需要修改 FPGA 程序，请忽略此项。

### MCU 程序目录架构

当拿到 EtherCAT 主站程序后，进行解压 MCU 控制程序后得到如下目录。

**MCU 硬件相关目录：** CORE、 FWLIB、 HARDWARE、 SYSTEM

**工程目录：** PRG

**业务开发目录：** EtherCat、 USER

用户开发目录代码主要是**业务开发目录**里面的代码，在里面根据所需业务功能进行修改即可

## FPGA 程序目录框架（如不涉及 FPGA 使用，请忽略此项）

用户拿到 FPGA 工程后，FPGA 代码文件主要如下所示。当需要对 FPGA 进行开发，可以参考我们样例，对 FPGA 程序进行增加，我们参考工程调用网表部分无需要变动

主站网表：EC\_MasterPro\_syn.vm

主站应用程序：EC\_MasterFpga.v

## 2.5 EtherCAT 主站概要框图

### 1、FPGA 和 PHY 连接框图

如图 2.5.1 所示，FPGA master IP 在 IP 内部引出控制线直接跟 PHY 连接。通过 PHY0 控制 EtherCAT 设备。

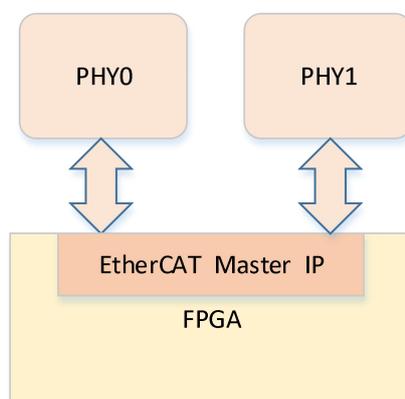


图 2.5.1

如图 2.5.2 所示，EtherCAT IP，会输出 ET\_CLK 25MHz 时钟给到 PHY 芯片，PHY 芯片需要选用支持 MII 协议。PHY 完成自适应后，在 ET\_RX\_CLK、ET\_TX\_CLK 会给到相应时钟，需要注意 25MHz 才是正确时钟，如果是 2.5MHz 说明网口跟 IP 自适应可能失败。

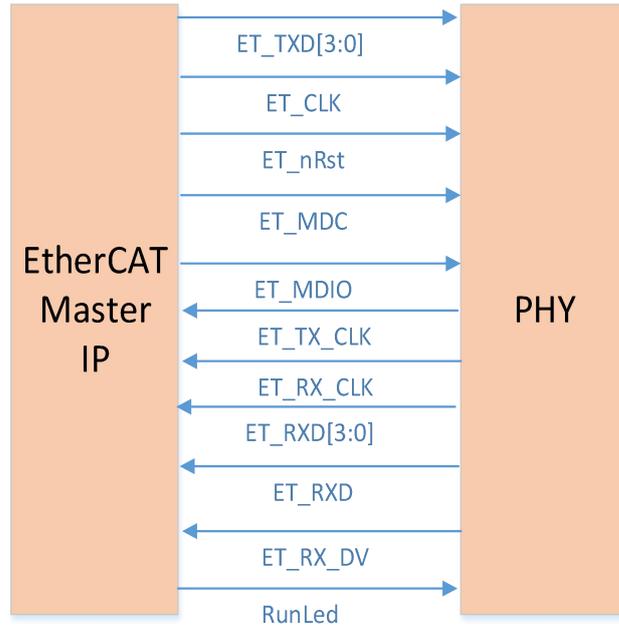


图 2.5.2

## 2、MCU 和 FPGA 连接框图

如图 2.5.3 所示，MCU 通过 FSMC 总线跟 FPGA 连接，MCU 只需要将数据根据 MCU 地址写入 fpga，fpga 会将数据打包通过 PHY 发送出去。当 phy 收到数据后，fpga 会解析存到 fpga 内部，MCU 通过 fsmc 总线读取 fpga 内部 ram。

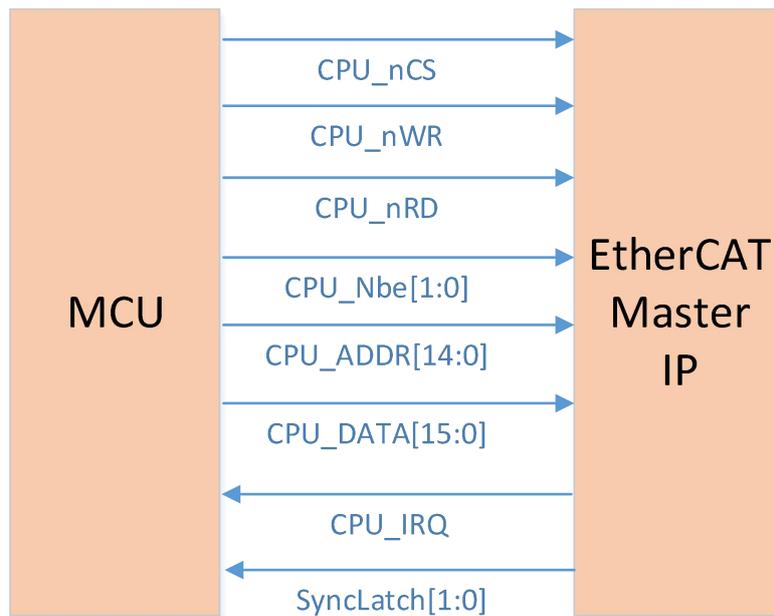


图 2.5.3

## 2.6 相关接口说明

### 1、FPGA EtherCAT master 模块说明（如用不上请忽略）

EtherCAT 主站 IP 描述如下

```
EC_MasterPro ec(  
    .Clk_100(Clk_Sys),  
    .nRst(nReset),  
    .nCs(nCsEtherCat),  
    .nWR(nMcuWen),  
    .nRD(nMcuRen),  
    .ADDR(McuAddr[16:1]),  
    .iDATA(iMcuData),  
    .oDATA(EtherCatReadData),  
    .nBEN(inMcuBen),  
    .INT(Ec_int),  
    .iMDIO(ET_MDI),  
    .oMDIO(ET_MDO),  
    .MDT(ET_MDT),  
    .MDC(ET_MDC),  
    .PHY_nRst(ET_nRst),  
    .Link(1'b1),  
    .TxClk(ET_TX_CLK),  
    .TxEn(ET_TX_EN),  
    .TxData(ET_TXD),  
    .RxClk(ET_RX_CLK),  
    .RxDv(ET_RX_DV),  
    .RxData(ET_RXD)  
);
```

提供一个 EtherCAT 主站参考 demo, 在参考 EC\_MasterFpga.v 中将对变量描述如下

变量名	i/o 方向	变量描述
Clk_25	i	25M 输入时钟
ET_CLK	o	PHY 控制时钟
McuAddr[17:0]	i	MCU 的 FSMC 控制地址

McuData[15:0]	io	MCU 的 FSMC 数据
nMcuBen[1:0]	i	MCU 的 FSMC 数据有效位
nMcuRen	i	MCU 的 FSMC 读信号
nMcuWen	i	MCU 的 FSMC 写信号
nMcuCS	i	MCU 的 FSMC 片选信号
McuInt	o	MCU 的 FSMC 初始化信号
ET_nRst	o	PHY 复位信号
ET_MDIO	io	PHY MDIO
ET_MDC	o	PHY MDC
ET_RXD[3:0]	i	PHY 输出数据
ET_RX_DV	i	PHY 输出有效信号
ET_RX_CLK	i	PHY 数据输出时钟
ET_TX_CLK	i	PHY 数据发送时钟
ET_TX_EN	o	PHY 数据发送使能信号
ET_TXD[3:0]	o	PHY 数据发送线
RunLed	o	PHY 数据状态运行灯

## 2、FPGA 寄存器

### 1. Type (0x0000)

Bit	Description	R/W	ResetValue
7:0	Typeof EtherCAT Master	RO	0X20

2. Revision (0x0001)

Bit	Description	R/W	ResetValue
7:0	Revision of EtherCAT Master	RO	0X0B

3. Build (0x0002:0x0003)

Bit	Description	R/W	ResetValue
7:0	Build of EtherCAT Master	RO	0X07E2 (2018)

4. DL1Control (0x0004:0x0007)

Bit	Description	R/W	ResetValue
15:0	PHY Write Data	R/W	0x0
20:16	PHY Register Address	R/W	0x0
23:21	PHYAddress	R/W	0x0
25:24	Command Register 00:No Command 01:Read 10:Write 11:No Command	R/W	0x0
31:26	Reserved	Ro	0x0

5. DL1Status (0x0008:0x000B)

Bit	Description	R/W	ResetValue

5:0	PHY Read Data	RO	0x0
16	Busy 0:MI control state machineisidle 1:MI control state machineisActive	RO	0x0
31:17	Reserved	RO	0x0

#### 6. DL2Control (0x000C:0x000F)

BIT	Description	R/W	ResetValue
15:0	PHY Write Data	R/W	0x0
20:16	PHY Register Address	R/W	0x0
23:21	PHY Address	R/W	0x0
25:24	Command Register 00:No Command 01:Read 10:Write 11:No Command	R/W	0x0
31:26	Reserved	RO	0x0

#### 7. DL2Status (0x0010:0x0013)

BIT	Description	R/W	ResetValue
15:0	PHY Read Data	RO	0x0

16	Busy 0:MI control state Machine is idle 1:MI control state Machine is Active	RO	0x0
31:17	Reserved	RO	0x0

#### 8. Destination Address (0X0014:0X0019)

Bit	Description	R/W	ResetValue
47:0	Destination Address	R/W	0xFFFFFFFF

#### 9. Source Address (0X001C:0X0021)

Bit	Description	R/W	ResetValue
47:0	Source Address	R/W	0X0

#### 10. DC Slaves Count (0X0024:0X0025)

Bit	Description	R/W	ResetValue
15:0	DC Slaves Count	R/W	0X0

#### 11. DC StartTime (0X0028:0X002F)

Bit	描述 (单位ns)	R/W	ResetValue
63:0	DC Start Time	R/W	0X0

12. Cyc Time (0X0030:0X0033)

Bit	描述 (单位ns)	R/W	ResetValue
63:0	Cyc Time	R/W	0X0

13. EtherCat Op Status (0x0034)

Bit	Description	R/W	ResetValue
3:0	EtherCat Op Status	R/W	0X0
相关值解析	EC_STATE_INIT = 0x01, /** Pre-operational. */ EC_STATE_PRE_OP = 0x02, /** Boot state*/ EC_STATE_BOOT = 0x03,/** Safe-operational. */ EC_STATE_SAFE_OP = 0x04,/** Operational */ EC_STATE_OPERATIONAL = 0x08, /** Error or ACK error */ EC_STATE_ACK = 0x10, EC_STATE_ERROR = 0x10		

14. Acyc Send Busy (0x0038)

Bit	Description	R/W	ResetValue
0	Acyc Send Busy 0:idle 1:busy	RO	0X0

15. Acyc Send Req (0x003c)

Bit	Description	R/W	ResetValue
0	Acyc Send Req 0:No Command 1:Req Send	RO	0X0

16. Port Send Time (0X0040:0X0047)

Bit	Description	R/W	ResetValue
63:0	Port Send Time	RO	0X0

17. Port Receive Time (0X0048:0X004f)

Bit	Description	R/W	ResetValue
63:0	Port Receive Time	RO	0X0

18. System Time (0X0050:0X0057)

Bit	描述 (单位ns)	R/W	ResetValue
63:0	System Time	RO	0X0

19. Line Status (0x0064) ResetValue

Bit	Description	R/W	ResetValue
0	Line Have broken 0:linegood 1:broken	R/W	0x0
1	Port1 Valid Sign 0:No link 1:Link100M	R/W	0x1

20. Port2 Shift Time (0X0068:0X006f)

Bit	Description	R/W	ResetValue
63:0	Port2 Shift Time	RO	0X0

### SDO数据映射地址范围

SENDSODORAMADDRESS :0x10000~0x103ff;

RECEIVESODORAMADDRESS:0x20000~0x203ff;

### PDO数据映射地址范围

SENDPDORAMADDRESS :0x18000~0x18fff;

RECEIVEPDORAMADDRESS:0x28000~0x28fff;

## 3、MCU 控制函数说明

用于拿到我们提供 EtherCAT 主站代码后，在代码量里面相关业务开发说明如下所示。

### 3.1 头文件部分宏定义

ethercatuser.h

```
#define ETHERCAT_AUTO_CONFIG 1
```

实例使用了 2 种方法进行配置初始化模式：

0：使用从站 xml 描述文件通过转换成 xti 描述文件进行初始化配置从站；

1：主站通过读取从站 eeprom 参数进行自动初始化配置；

```
#define ETHERCAT_CYC_PARAM 1000000
```

EtherCAT 的通信周期设置，单位为 ns，故上面设置为 1ms 的通信周期；

```
#define ETHERCAT_DC_SHIFT 50000
```

SYNC 同步输出时间偏移设置，相对于周期通信的时间节点进行偏移；

#### **ethercatxti.h**

```
#define XTI_FLASH_ADDR 0x08040000
```

存放 XTI 二进制文件的开始地址，用于保存解析文件的基地址；

### **ethercatconfig.h**

```
#define EtherCatBaseAddress    0x6C040000
```

设置 etherCAT 基地址。使用了 stm32f407 的片选 4 地址为 0x6c000000，fpga 应用使用了 assignnCsEtherCat=(!nMcuCS&&McuAddr[17])?1'b0:1'b1；  
所以得出 etherCAT 基地址 0x6C040000。

## **3.2 业务函数**

```
ec_set_cyc_param(&Context, ETHERCAT_CYC_PARAM);
```

### **函数说明：**

初始化EtherCAT缓存堆Context，并设置通信周期为ETHERCAT\_CYC\_PARAM；

```
ecx_communication_status_switch(&Context, EC_STATE_OPERATIONAL);
```

### **函数说明：**

管理EtherCAT 通信阶段的切换和初始化从站，配置从站；在内部使用了 Servo\_Control\_State(EtherCat\_Context\*Context)管理从站控制字及状态切换；

```
ec_SD0read(unsigned short slave, unsigned short index, unsigned  
charsubindex, unsignedcharCA, int*psize, void*p, inttimeout)
```

### **函数说明：**

根据索引、子索引等进行对支持COE从站进行参数读取操作。

```
ec_SD0write(unsigned short Slave, unsigned short Index, unsigned  
charSubIndex, unsignedcharCA, intpsize, void*p, intTimeout)
```

### **函数说明：**

根据索引、子索引等进行对支持COE从站进行参数写入操作。

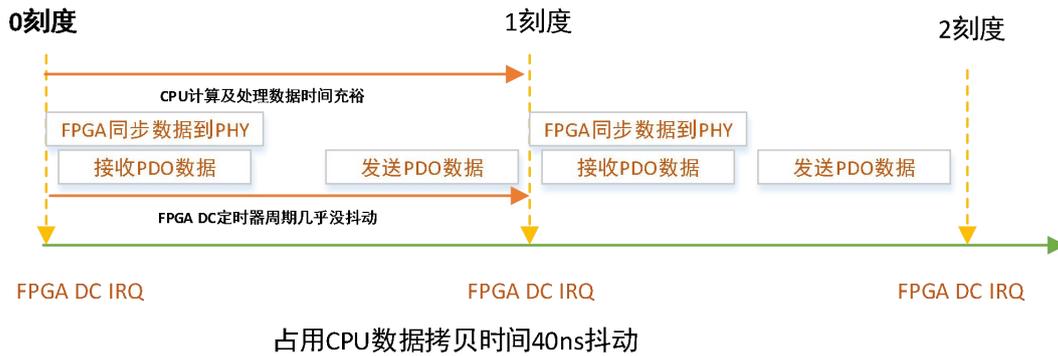
### 3.3 中断处理流程

#### 1) 中断处理框图

根据用户设置，fpga 周期触发 MUC 中断，MCU 进入中断流程如图，该流程是主站处理硬实时流程。用户需要在周期时间内把数据写入 fpga 中，如果错过周期，fpga 将会继续发送上次数据给 PHY。



在中断过程中，fpga 和 CPU 相互交接流程如下所示，CPU 只管进行业务处理，将业务处理后结果写入/读取 FPGA，FPGA 会将数据打包发送给 PHY 中。



## 2) 中断处理主要函数

`Void ec_send_process_data(void);`

函数说明:

把周期数据 PDO 写入 EtherCAT 协议栈, 由 fpga 实时发送;

`Void ec_receive_process_data(void);`

函数说明:

把接收的周期数据 PDO 从 EtherCAT 协议栈读到缓存堆中;

`Void ec_receive_aperiodic_data(void);`

函数说明:

查询是否接收到 SDO 非周期数据包并读入到缓存堆栈里面;

`Void ex_int_init();`

函数说明:

业务处理中断初始化函数, 该函数初始化中断, 并将业务处理函数注册进系统中。

## 3.4 伺服电机初始化部分

`Void ecx_config_universal_coe_drive(EtherCat_Context*Context)`

•

**函数说明：**对伺服/步进电机进行封装分类，应用于 ETHERCAT\_AUTO\_CONFIG==0 自动初始化；在此添加通用伺服/步进电机的 eep\_man 和 eep\_id 由程序自动识别为驱动电机类，进而自动初始化及分配空间；

```
intinit_txpdo_rxpdo_map(EtherCat_Context*Context);
```

**函数说明：**自动分配缓存堆，也就是从这里确定一个从站伺服/步进电机的数据所在缓存位置，也是在这里添加通用伺服/步进电机的 eep\_man 和 eep\_id；

```
intEC_COE_UNI_Drive_Setup(unsignedshortslave);
```

**函数说明：**在此函数确定通用伺服/步进电机的通信内容及数据量，可根据需求进行配置。

### 三、生成 XTI 操作步骤

TwinCAT 安装包如下：

链接：[https://pan.baidu.com/s/1leT\\_Rr4SBUqOmFr.jVIOYmW?pwd=1234](https://pan.baidu.com/s/1leT_Rr4SBUqOmFr.jVIOYmW?pwd=1234)

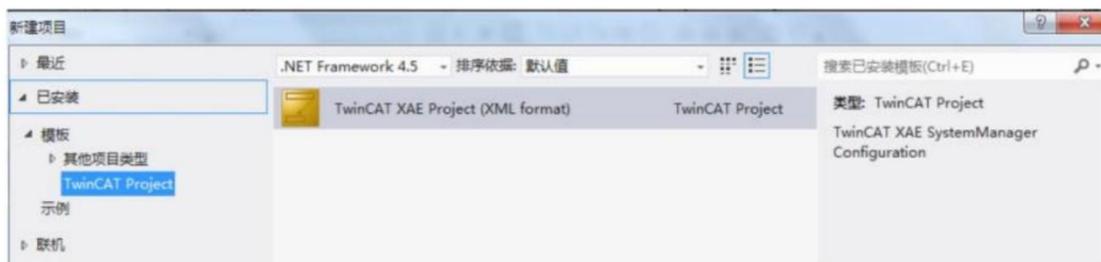
提取码：1234

1. 将所需要连接的从站的 xml 文件复制到 Twincat 软件的 \3.1\Config\Io\EtherCAT 目录下。

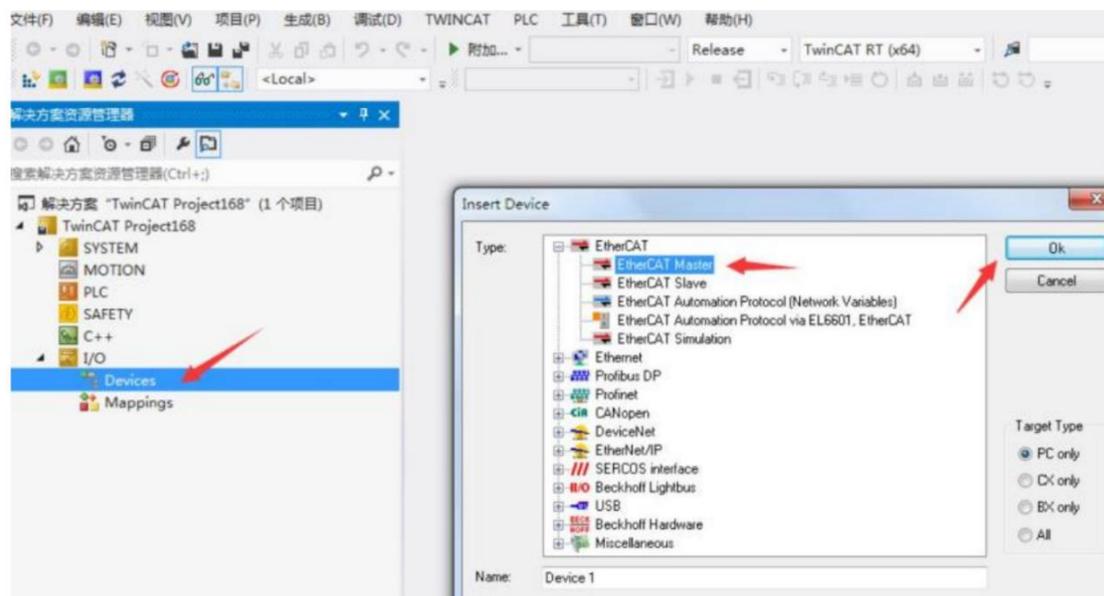
2. 打开 Twincat，电机文件->新建->项目



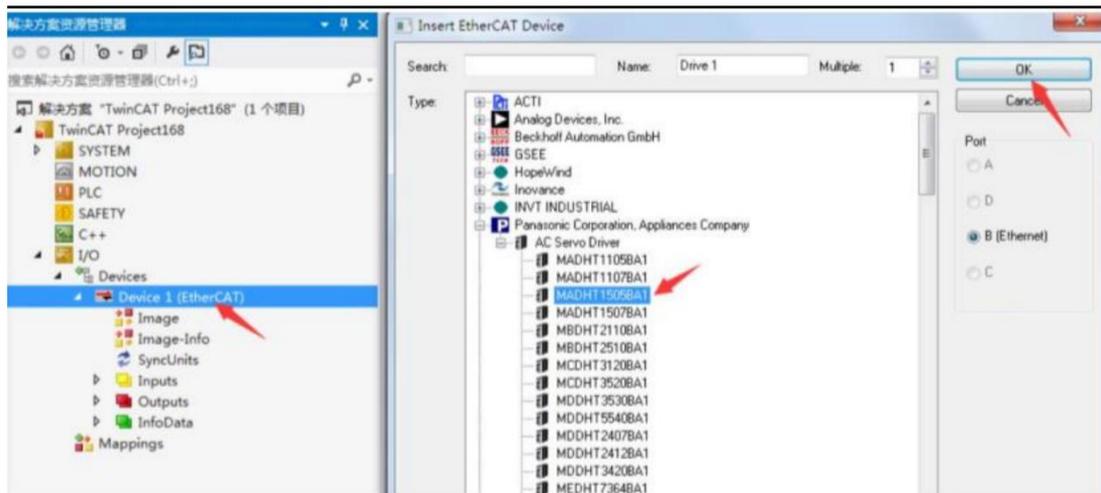
3. 选择 TwincatXAEPProject (XMLformat) , 并点击确认



4. 右键点击界面右侧的 Device, 在弹出的菜单中点击添加新项, 然后选择 EtherCATMaster 并点击 OK。在之后弹出的 Devicefoundat 菜单中, 选择 none 即可

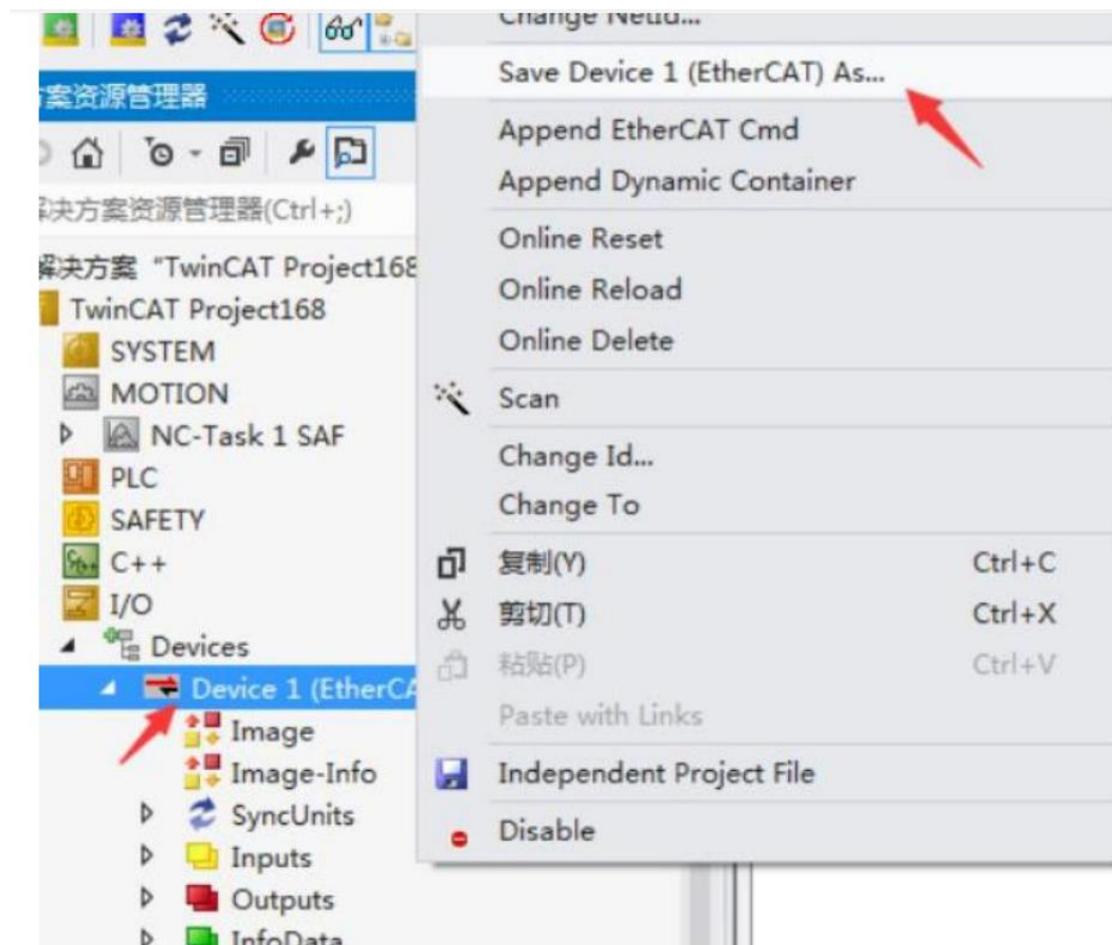


5. 右键点击 Device1, 并点击添加新项, 在弹出的 InsertEtherCATDevice 选择主站连接的第一个从站, 之后点击 OK 添加。(本例中第一个从站为松下 A5B 驱动器)



6. 使用相同的方法依次添加第二个从站、第三个从站...最后一个从站。

7. 右键点击 Device1 (EtherCAT), 在弹出菜单栏中选择 SaveDevice1 (EtherCAT) As..., 将配置保存成 xti 文件。



•

8. 打开 XmlParser 并将保存的 xti 文件生成二进制文件。（生成的文件为 hexConf, 手动将后缀修改为. bin）

9. 将. bin 二进制文件烧录到#defineXTI\_FLASH\_ADDR 0x08040000 定义的地址中。

## 四、常见问题解答

1.主站占用 fpga 资源多少？

答：大概 4k LUT.

2.主站支持连接从站数量是多少？

答：目前数据最大支持 4k 字节长度，按每个站点 8 个字节，大概可以支持 500 个从站左右

3.确认 EtherCAT 主站在 256 轴电机需要资源

答：如果控制 256 轴以上需要增加 FPGA RAM，需要换高级 FPGA。控制多少轴跟资源没关系（资源大概 4K 左右）

4.样卡 ethercat 主站最多能控制多少轴电机

当前主站 FPGA 是 PGL25G，可控制 EtherCAT 主站 128 轴

5.EtherCAT 主站最低在紫光哪款芯片使用

在紫光 PGL12G 可以放进 EtherCAT 主站，具体能控制多少轴还不确定（跟 FPGA 内部 RAM 有关系）。

6.紫光 EtherCAT 主站和传统主站上有区别吗？

答：传统主站是由电脑/MCU/CPU 充当，价格贵且相对没 FPGA 稳定，在紫光 PGL12G 可以使用主站。价格优势显著

7.FSMC 速率是多少？

答：当前设置时钟是 100MHz，数据线 16 根，总共 16\*100MHz 得访问速率

8. 在中断临界过程中，MCU 将数据写入 FPGA，FPGA 此时在向 PHY 发送数据，数据是否有冲突？

答：不会，FPGA 有预留这种保护机制。