

Application Note

TI 嵌入式处理器作为 CODESYS EtherCAT 控制器的性能指标



Daolin Qiu, Pekka Varis, Jared McArthur

摘要

几乎所有的实时工业控制都依赖于周期性计算和通信的架构。设计解决方案的性能估算通常基于中断延迟等微基准测试（例如 **Cyclictest**）和一些计算基准测试（如备受推崇的 **Dhrystone**）。EtherCAT® 是用于机器人、工厂自动化和精确电机驱动控制领域中的控制系统的主要工业以太网协议之一。主要性能指标是在给定设备数量和交换数据结构大小的情况下，可实现的最短周期时间。无论是在控制器内还是在网络内，时序和同步稳定性都是另一个重要考虑因素。EtherCAT 控制器（以前称为 EtherCAT 主站）解决方案可用于高级操作系统（例如适用于 Windows® 的 TwinCAT® 和适用于 Linux 的 CODESYS®），以及高度优化的商业解决方案（例如 icECAT 和开源 IgH 协议栈）。一些优化的设计能够在微控制器核心上运行，并且彼此之间相比以及与全功能高端解决方案相比涉及不同的权衡。本应用手册探讨了使用 CODESYS EtherCAT 协议栈运行实时 Linux 的德州仪器 (TI) 嵌入式处理器上可实现的周期时间。

内容

1 引言.....	2
1.1 什么是 EtherCAT？.....	2
1.2 什么是 PLC？.....	3
1.3 什么是 CODESYS？.....	3
2 评估平台和方法.....	4
2.1 硬件.....	4
2.2 软件.....	4
2.3 测试拓扑.....	5
3 性能指标.....	6
3.1 Cyclictest 性能指标.....	6
3.2 EtherCAT 性能指标.....	8
4 优化.....	9
4.1 已实现的优化.....	9
4.2 未来注意事项.....	13
5 总结.....	15
6 参考资料.....	15
7 附录 A：如何使用 CODESYS 协议栈将 TI 嵌入式处理器设置为 EtherCAT 控制器.....	17
7.1 硬件要求.....	17
7.2 软件要求.....	17
7.3 硬件设置.....	18
7.4 软件设置.....	18
7.5 如何查看性能测量结果.....	22
8 附录 B：如何在 CODESYS 协议栈上实现无限运行时间.....	25
8.1 CODESYS 许可背景.....	25
8.2 获取 CODESYS 许可证.....	25
8.3 激活 CODESYS 许可证.....	25
8.4 验证已应用 CODESYS 许可证.....	26

商标

EtherCAT® and TwinCAT® are registered trademarks of Beckhoff Automation GmbH.

Windows® is a registered trademark of Microsoft Corporation.

CODESYS® is a registered trademark of CODESYS Group.

ARM® is a registered trademark of ARM Limited.

所有商标均为其各自所有者的财产。

1 引言

工业控制需要具有确定性延迟的实时通信。所使用的技术已从串行现场总线发展到 EtherCAT 等 IEC 标准中定义的工业以太网协议。这些标准使用 IEEE 以太网的一部分来利用以太网提供的一些规模经济，但这些标准带来了一些小的变化（例如直通交换），这些变化超越并部分限制了典型 IEEE 桥接器和端点的使用。与典型的消费类或企业系统（其中平均响应能力或吞吐量是关键性能指标）不同，在这些应用程序中，性能受网络中与输入和输出交互的最坏情况延迟的限制。

典型的工业控制网络拓扑如图 1-1 所示。以太网指定局域网 (LAN) 的第 1 层和第 2 层。对于标准以太网，这使得可变大小帧可以从一个端点无状态且不可靠地传输到另一个端点，以及在两个端点之间切换。此域由图 1-1 中左侧虚线矩形区域表示。顶部的协议（例如 EtherCAT）是高度非对称的；有一个控制器管理少则几个多则数百个设备。不同的协议对这种非对称关系使用的术语略微不同，并且这种不对称的程度也各不相同。本文档使用“控制器”一词表示实施控制的实体，使用“设备”一词表示被控制的设备。

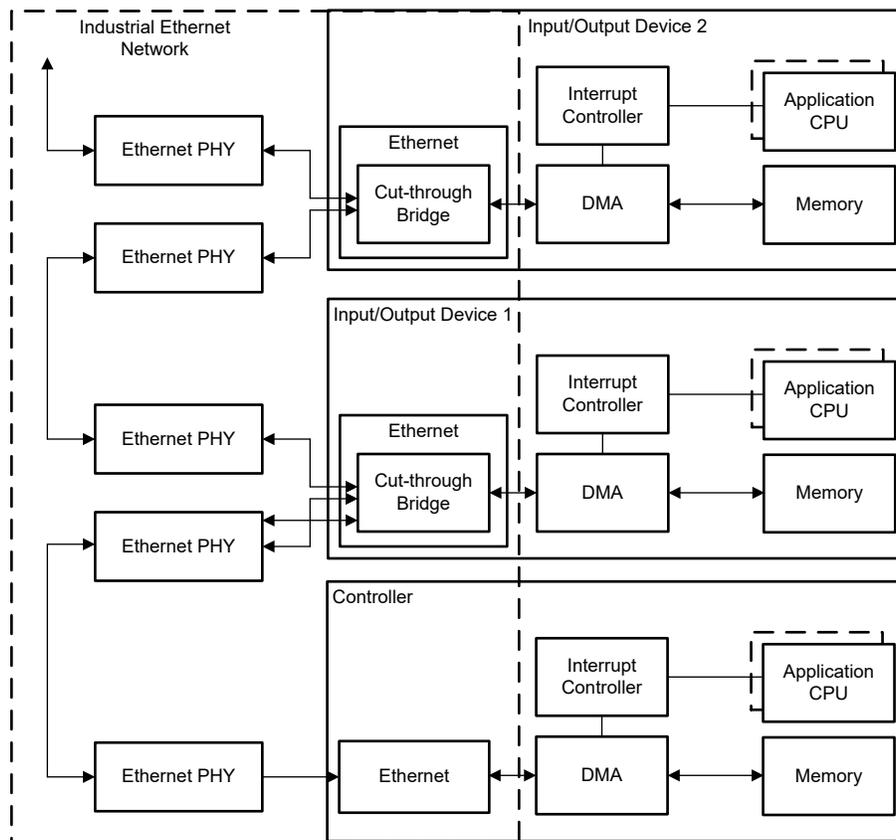


图 1-1. 典型的工业以太网

1.1 什么是 EtherCAT？

EtherCAT 是一种基于 IEEE 802.3 以太网的现场总线系统，由国际电工委员会 (IEC) 61158 进行了标准化。该技术由 EtherCAT 技术协会（一个由用户和供应商组成的国际社区）提供支持。该协议在运动和电机控制中尤为常用。EtherCAT 的主要优势可在要求数据更新时间短和通信抖动低的自动化应用中充分发挥。在 EtherCAT 协议中，EtherCAT 控制器发送一个通过每个设备节点的帧。每个 EtherCAT 设备在检测到寻址到设备的数据后立即读取该数据。然后，当动态桥接帧时，设备会将数据插入到帧中。段（或分支）中的最后一个设备检测到开放端

口，便将报文发送回控制器。EtherCAT 控制器是段中唯一主动发送 EtherCAT 帧的节点。该功能允许网络实现 90% 以上的可用网络带宽，同时防止不可预测的延迟，从而保证实时系统响应。EtherCAT 是使用 Ethertype 标识符 (0x88A4) 传输的。LAN 上发送的唯一帧来自 EtherCAT 控制器，分支上的最后一个从属控制器将帧返回给前一个控制器。控制器中的典型优化是使协议栈直接访问以太网 MAC 控制器，不仅绕过网络协议栈（就像基于原始以太网的 OPC UA Pub-Sub 一样），还绕过以太网驱动程序来直接或本地拥有整个以太网外设。这种情况的一个示例如图 1-2 所示。Acontis 和 IBV 是提供此优化的协议栈提供商。

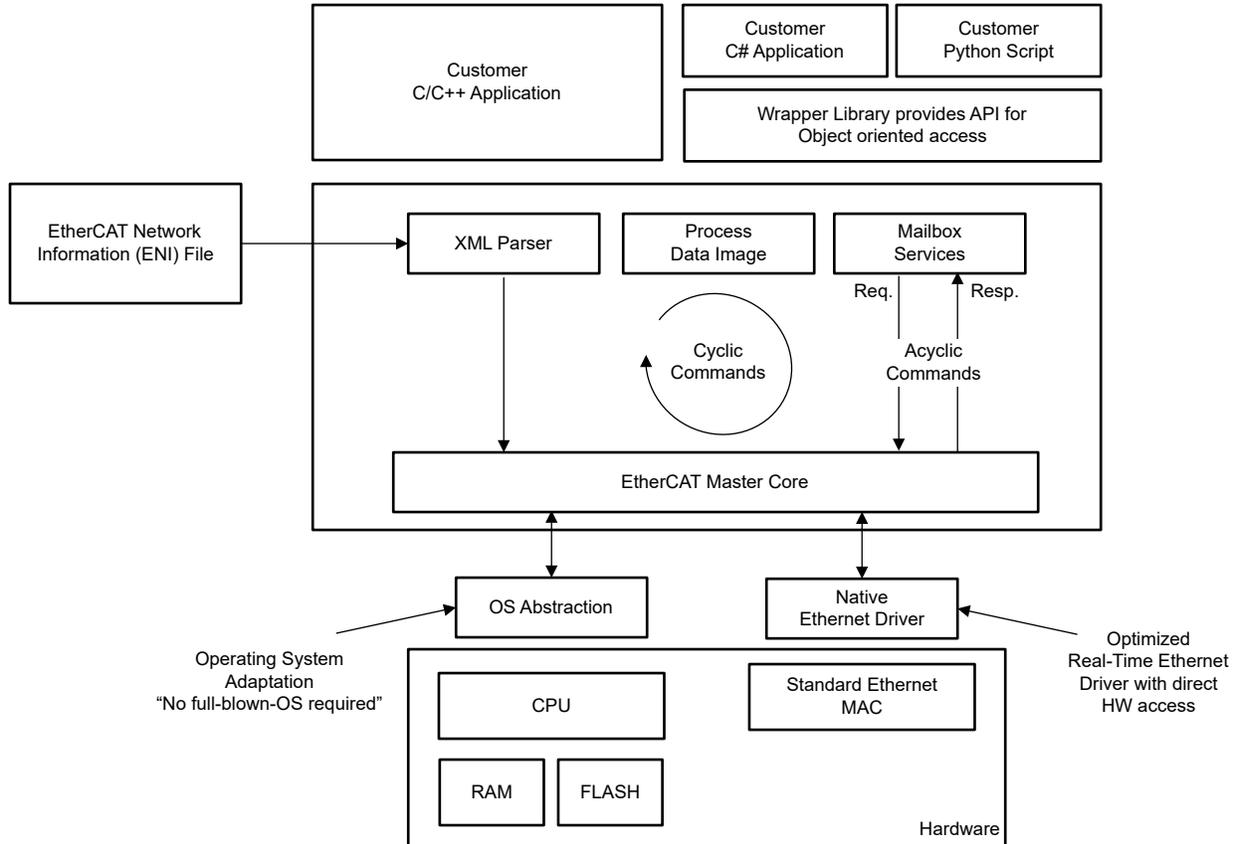


图 1-2. EtherCAT 控制器软件架构

1.2 什么是 PLC ？

可编程逻辑控制器 (Programmable Logic Controller, PLC) 和移动控制器是用于控制输入、输出和电机的工业级计算机，主要用于需要高可靠性、易于编程和过程故障诊断的制造过程。通常，PLC 可以使用工业以太网协议栈（例如 EtherCAT 控制器通信协议栈）将数据从控制器传输到用于控制分布在 LAN 上的传感器和执行器的设备。PLC 通常用于实时和周期性的应用中。当每个周期性输入到达时，就会进行计算以生成新的输出。计算和通信延迟变化（通常称为抖动）会对可实现的 EtherCAT 周期时间产生不利影响。

1.3 什么是 CODESYS ？

CODESYS 是 *COntrols DEvelopment SYStem* 的缩写，它是用于根据 IEC 61131-3 对 PLC 应用程序进行编程的集成开发环境。CODESYS 是独立于制造商的领先 IEC 61131-3 自动化软件，适用于工程控制系统。使用 CODESYS 时，软件有两个关键部分。第一部分是称为 CODESYS Development System 的集成开发环境，它充当程序员和控制工程师的 GUI，用于使用 IEC 61131-3 编程语言编写控制应用程序并为这些控制应用程序创建可视化效果。第二个是通常安装在 PLC 上的运行时软件，但 CODESYS 可以安装在满足运行时最低要求的任何智能设备上。运行时系统的一个示例是 CODESYS Control for Linux ARM SL，适用于基于 ARM®、用于控制应用程序的电路板或设备，预装了 Linux，有或没有硬实时要求。

2 评估平台和方法

2.1 硬件

通过使用 CODESYS EtherCAT 协议栈，SK-AM62B 作为 EtherCAT 控制器运行时可捕获性能指标。由十个 Beckhoff 数字输出器件 (EL2889) 组成的网络使用 Beckhoff EtherCAT 耦合器 (EK1100) 连接到 EtherCAT 控制器。这些指标在同一网络中的 TMD564EVM、SK-AM69 和 SK-TDA4VM 上会进一步捕获。

除了 AM62A 平台之外，表 2-1 中的所有平台都作为 EtherCAT 控制器进行基准测试。如表中所示，AM62A 在更高的 DDR 速度和更大的 DDR 总线宽度方面具有一些潜在的优势。

表 2-1. 德州仪器 (TI) 基于 ARM® 的处理器

器件型号	ARM CPU	ARM (最大) MHz	操作系统	高速缓存	EVM 上的外部内存
AM62x	4 个 Cortex-A53 SoC	1400 (64 位)	Linux	32KB L1 DCache 32KB L1 ICache 512KB L2 高速缓存	DDR4 1600MT/s 16 位宽度
AM64x	2 个 Cortex-A53 SoC，带 Cortex-R5F 协 CPU	800、1000 (64 位)	Linux、RTOS	32KB L1 DCache 32KB L1 ICache 256KB L2 高速缓存	DDR4 1600MT/s 16 位宽度
AM69	8 个 Cortex-A72 SoC，带 Cortex-R5F 协 CPU	2000 (64 位)	Linux、RTOS	32KB L1 DCache 48KB L1 ICache 2MB L2 共享缓存	LPDDR4 2133MT/s 32 位宽度
TDA4VM	2 个 Cortex-A72 SoC，带 Cortex-R5F 协 CPU	2000 (64 位)	Linux、RTOS	32KB L1 DCache 48KB L1 ICache 1MB 共享 L2 高速缓存	LPDDR4 2133MT/s 32 位宽度
AM62A	4 个 Cortex-A53 SoC	1400 (64 位)	Linux	32KB L1 DCache 32KB L1 ICache 512KB 共享 L2 高速缓存	LPDDR4 3733MT/s 32 位宽度

为了更好地复现真实的工厂自动化环境 (其中 EtherCAT 通常无限期使用)，需要获取 CODESYS 许可证并将其应用于专用的 CODESYS USB 加密狗。如果没有 CODESYS 许可证，每次运行都会在 30 分钟后超时，此时需要手动重新启动运行。要打破运行 30 分钟的时间限制，每个目标 EtherCAT 控制器都需要一个 USB 端口来读取在 CODESYS USB 加密狗上应用的许可证。

2.2 软件

SK-AM62B 和 TMD564EVM 在实时 Linux 操作系统上运行，使用 SD 卡启动，SD 卡上分别刷入了 PROCESSOR-SDK-LINUX-RT-AM62x 和 PROCESSOR-SDK-LINUX-RT-AM64x 默认 wic 映像，这些映像来自软件开发套件 (SDK) 版本 09.01.00.08。

TDA4VM 和 AM69 在来自各自 SDK 09.01.00.06 版本的实时内部版本上运行。

通过以下操作在 Linux 根目录中手动安装 CODESYS Control for Linux ARM64 4.9.0.0 版：从 CODESYS 网站下载软件包，提取 ipk 软件包，将 ipk 复制到目标 EtherCAT 控制器的 Linux 根目录中，并使用 opkg 命令提取软件包。

除了 CODESYS Control for Linux ARM64 版本 ipk 软件包之外，还必须安装 CodeMeter 应用程序，以读取 CODESYS USB 加密狗上的 CODESYS 许可证。必须从 CODESYS Control for Linux ARM64 已下载的软件包中提取该应用程序 (它采用 deb 软件包的形式)，并使用 opkg 命令手动安装在目标 EtherCAT 控制器上。

有关如何使用 CODESYS 协议栈将 TI 嵌入式处理器设置为 EtherCAT 控制器的更具体说明，请参阅节 7。

2.3 测试拓扑

如节 2.1 中所述，用作 EtherCAT 控制器的每个 TI 嵌入式处理器都使用 Beckhoff EtherCAT 耦合器连接到 10 个 Beckhoff 数字输出设备。除此之外，通过另一个以太网端口，每个测试的 EtherCAT 控制器都连接到以太网交换机，该交换机还连接运行 CODESYS Development System 的 Windows 计算机，以实时监控性能指标。（可选）将 DHCP 服务器连接到以太网交换机，为 Windows 计算机和目标 EtherCAT 控制器动态分配 IP 地址。此处报告的基准测试使用静态分配的 IP 地址。如果 Windows 计算机之前已连接到互联网，则在连接到 EtherCAT 网络后，可能无法再访问互联网。

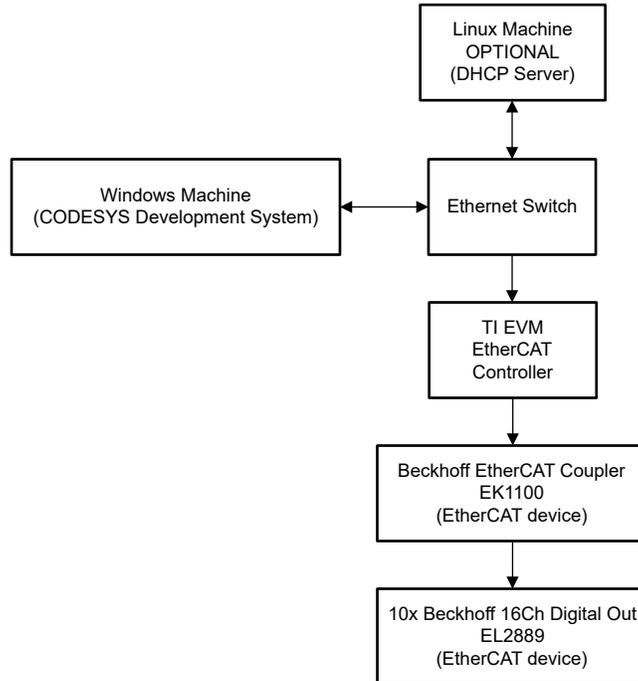


图 2-1. 使用 CODESYS Control for Linux ARM SL 的 EtherCAT 网络测试拓扑

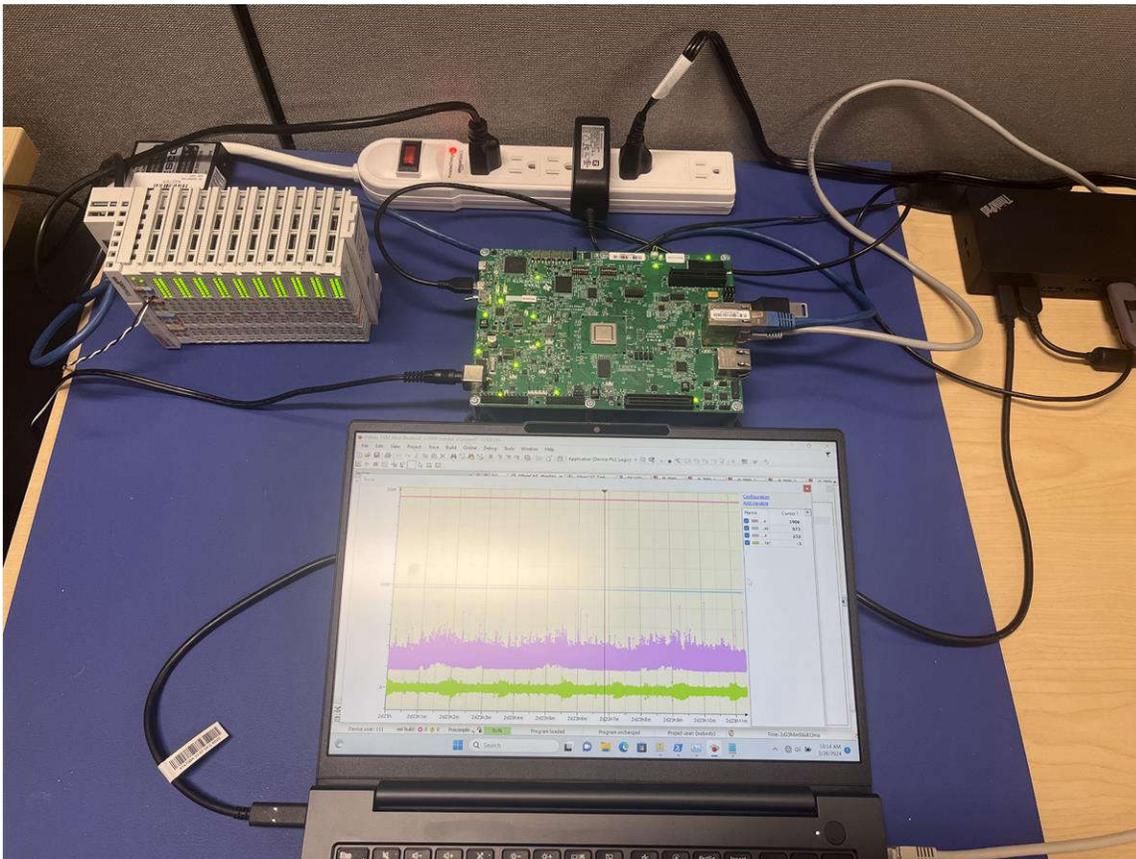


图 2-2. AM64x 作为控制器且 Beckhoff 设备作为 EtherCAT 设备的 EtherCAT 网络

3 性能指标

3.1 Cyclictest 性能指标

在运行 EtherCAT 应用程序时，在评估每个硬件平台的性能之前，通常建议先通过运行 *cyclictest* 收集基准性能统计信息。这些统计信息是运行 CODESYS 协议栈时 EtherCAT 控制器性能的良好指标。

Cyclictest 是 Linux 中的一种实用工具（另请参阅 [Linux Foundation](#)），它可以重复地准确测量线程的预期唤醒时间与线程实际唤醒时间之间的差异，以提供有关系统延迟的统计信息。实时系统中由硬件、固件和操作系统引起的延迟可以通过 *Cyclictest* 来测量。

Cyclictest 的结果将与开源自动化开发实验室 (Open Source Automation Development Lab, OSADL) 在 [QA Farm on Real-time of Mainline Linux](#) 中发布的数据采用相同的直方图格式绘制。OSADL 是一个组织，它的宗旨是通过引入更广泛的社区来帮助开发，以促进个人、团体和公司开发开源软件。QA Farm 是 OSADL 提供的一项服务，它是嵌入式系统的质量保证和评估测试中心。

Cyclictest 的结果是在 AM62x 评估板 (SK-AM62B)、AM64x 评估板 (TMD64EVM)、TDA4VM 入门套件 (SK-TDA4VM) 和 AM69 入门套件 (SK-AM69) 上捕获的，以显示在这些平台上运行的基本延迟。SK-AM62B 和 TMD64EVM 分别在 PROCESSOR-SDK-LINUX-RT-AM62x 和 PROCESSOR-SDK-LINUX-RT-AM64x 默认 wic 映像上运行，这些映像均来自软件开发工具包 (SDK) 版本 09.01.00.08。TDA4VM 和 AM69 在来自各自 SDK 09.01.00.06 版本的实时内部版本上运行。在负荷试验下，每个 *cyclictest* 均使用 *stress-ng* 工具运行 6 小时。如需了解此设置，请参阅下面的代码片段。

```
stress-ng -c <number of cpu cores> --cpu-method all &
cyclictest -m -Sp98 -d6h -h400 -i200 -q > <histogram name>.hist
```

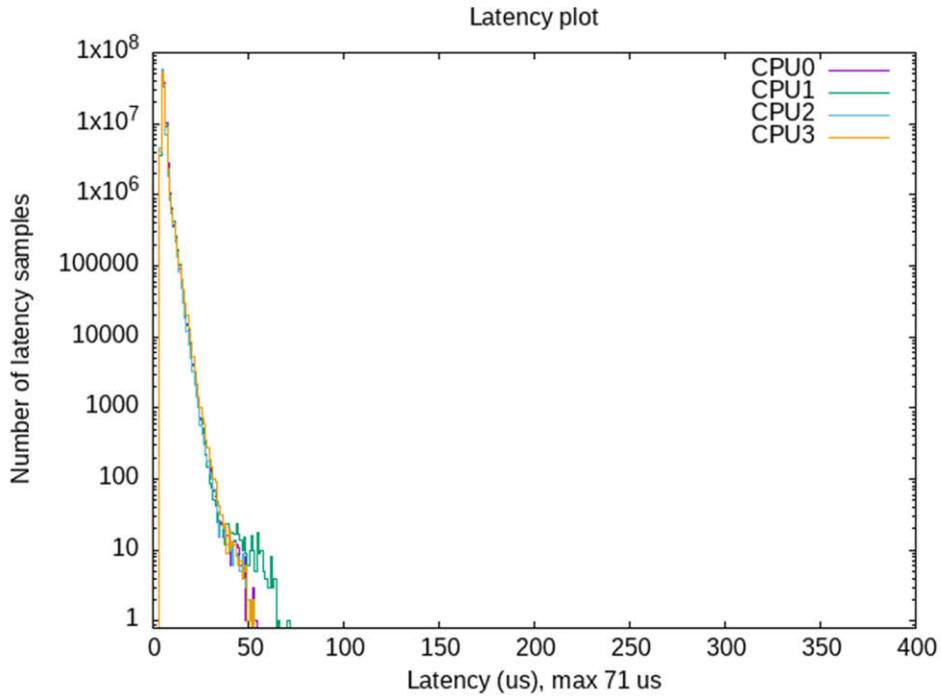


图 3-1. 在使用 stress-ng 的情况下 AM62x 运行 6 小时的延迟图

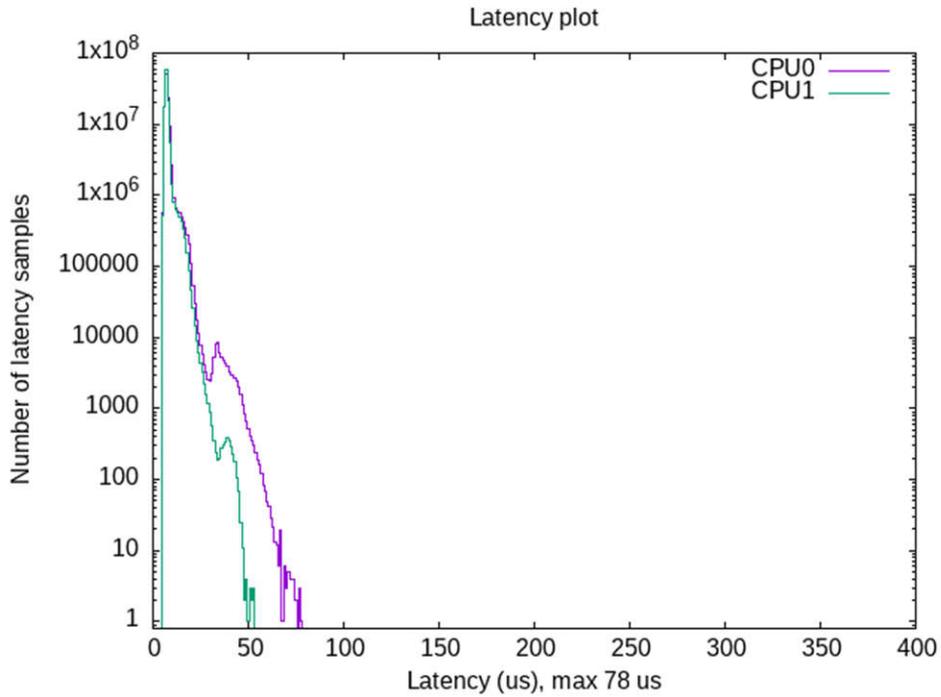


图 3-2. 在使用 stress-ng 的情况下 AM64x 运行 6 小时的延迟图

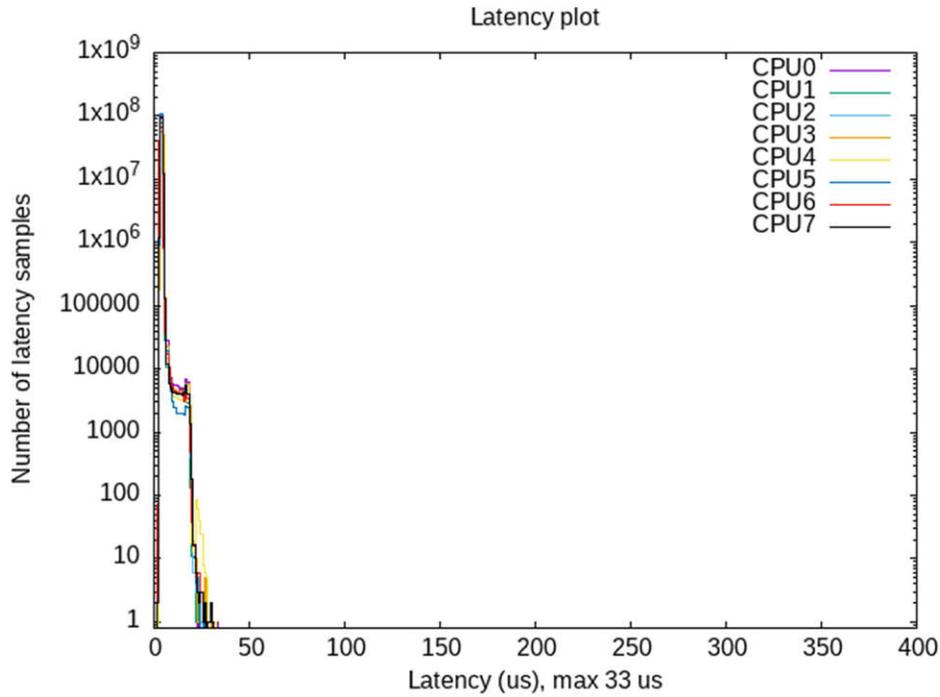


图 3-3. 在使用 stress-ng 的情况下 AM69 运行 6 小时的延迟图

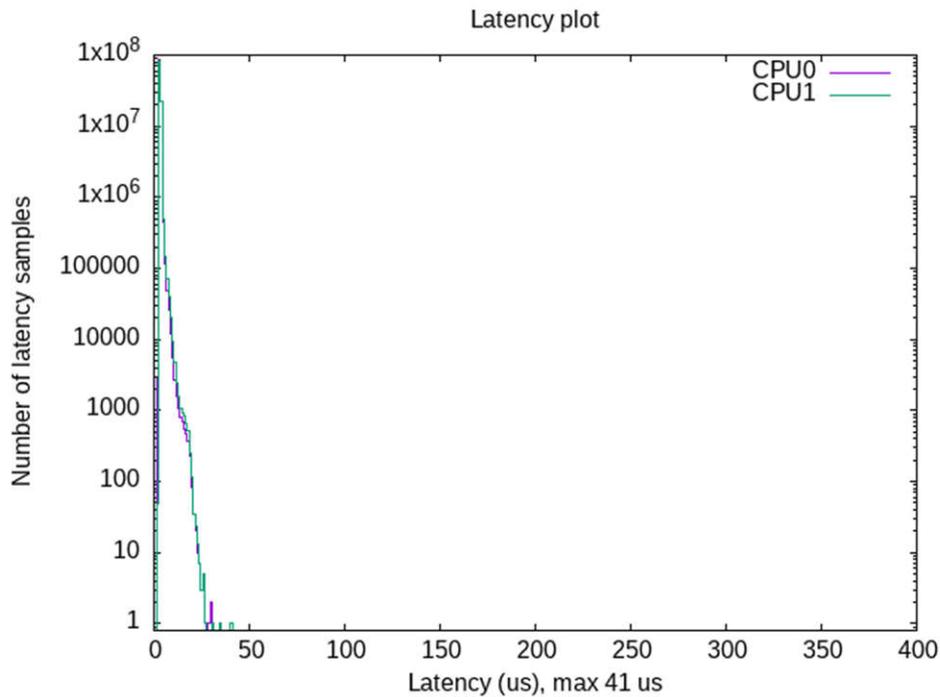


图 3-4. 在使用 stress-ng 的情况下 TDA4VM 运行 6 小时的延迟图

3.2 EtherCAT 性能指标

下表 (表 3-1) 总结了使用 CODESYS EtherCAT 协议栈产生的最大测量周期时间和最大抖动的关键性能指标 (KPI)。

表 3-1. CODESYS® EtherCAT 性能摘要 (1ms 周期时间)

HW	运行时	ECAT 网络	最大周期时间	过滤后的最大周期时间 ⁽¹⁾	最大抖动	过滤后的最大抖动 ⁽¹⁾
AM62x	63 小时	11 个 Beckhoff EtherCAT 设备	700µs	500µs	116µs	80µs
AM64x	63 小时	11 个 Beckhoff EtherCAT 设备	1906µs	865µs	973µs	112µs
AM69	63 小时	11 个 Beckhoff EtherCAT 设备	384µs	250µs	53µs	45µs
TDA4VM	63 小时	11 个 Beckhoff EtherCAT 设备	371µs	330µs	65µs	48µs

表 3-1 中所有用作 EtherCAT 控制器的 TI 硬件平台均通过由十个 Beckhoff 16 通道数字输出器件 (EL2889) 组成的 EtherCAT 设备网络进行了基准测试，该网络通过 Beckhoff EtherCAT 耦合器 (EK1100) 进行连接。所有 160 个通道每 1 秒在高电平和低电平之间切换一次。周期时间段配置为 1000µs。

表 3-1 中 AM62x 和 AM64x 的结果来自性能改进调整。

4 优化

4.1 已实现的优化

通过禁用多个后台进程并调整在 ARM64 A53 核心上 Linux 中运行的 CODESYS 特定进程，调整了 AM62x 和 AM64x EtherCAT 控制器，实现了更好的性能。从 AM62x 的实时 “tisdk-default-image” 软件开发套件 (SDK) 版本 09.01.00.08 映像中禁用的主要后台进程有 *ti-apps-launcher*、*pulseaudio*、*systemd-timesyncd*、*telnetd*、*weston* 和 *containerd*。最初激活这些后台进程是为了展示与 EtherCAT 应用程序无关的功能。禁用这些进程的目的在于减少四个 A53 核心上的部分 CPU 负载，并将映像配置为接近 AM62x 的实时 “tisdk-thinlinux-image”。这样做是因为当将默认映像与精简映像进行比较时，周期时间性能略有改善，从大约 350µs 的平均值更改为大约 250µs。

对周期时间的一个主要影响是名为 Codemeter 的 CODESYS 许可应用程序。大多数计划使用 CODESYS 的 EtherCAT 应用程序都需要使用正在运行的 Codemeter 来实现无限的运行时间。在没有许可应用程序的情况下，CODESYS 只能运行约 30 分钟，然后系统会终止，必须重新启动 (另请参阅 [CODESYS Control Standard S](#))。Codemeter 是在接近真实 EtherCAT 用例的环境中进行基准测试的必备应用程序；不过，大约每隔 5-6 分钟，应用程序的 CPU 负载就会激增到接近 100%，这随后会导致周期时间 (从大约 350µs 到大约 1000µs) 和抖动激增。随着时间的推移，可通过在 Linux 中监控 *htop* 以及在 CODESYS Development System GUI 中监控周期时间和抖动，可观察到这种激增。

```

0[##### 5.3% Tasks: 43, 66 thr, 129 kthr; 4 running
1[##### 3.2% Load average: 0.52 0.53 0.43
2[##### 2.7% Uptime: 00:23:46
3[#####100.0%
Mem[##### 243M/1.89G
Swp[##### 0K/0K

CPU PID USER PRI NI VIRT RES SHR S CPU%-MEM% TIME+ Command
0 3356 daemon 20 0 768M 53612 25668 S 108. 2.7 0:24.32 /usr/sbin/CodemeterLn -f
3 3371 daemon 20 0 768M 53612 25668 R 106. 2.7 0:16.19 /usr/sbin/CodemeterLn -f
  
```

图 4-1. htop 中的 Codemeter CPU 激增

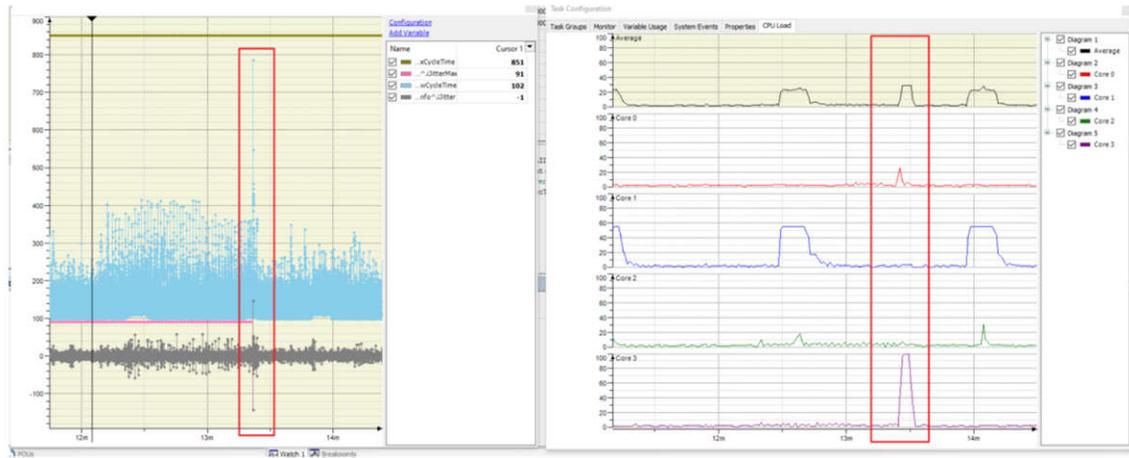


图 4-2. Codemeter CPU 负载激增导致的最大周期时间激增

经过进一步调查，Codemeter 的 CPU 亲和性似乎默认在 CPU 0、2 和 3 之间变化，并且从不使用 CPU 1。CPU 1 是默认用于运行 CODESYS 控制应用程序的核心。通过将 CPU 亲和性更改为 CPU 1，巨大的周期时间峰值（约 1000 μ s）减小到约 500 μ s。图 4-3 中高于 400 μ s 的一般峰值是由于在这些时间段内运行 *htop* 所致。关于 CPU 亲和性变化为何会降低峰值的详细信息仍不清楚。

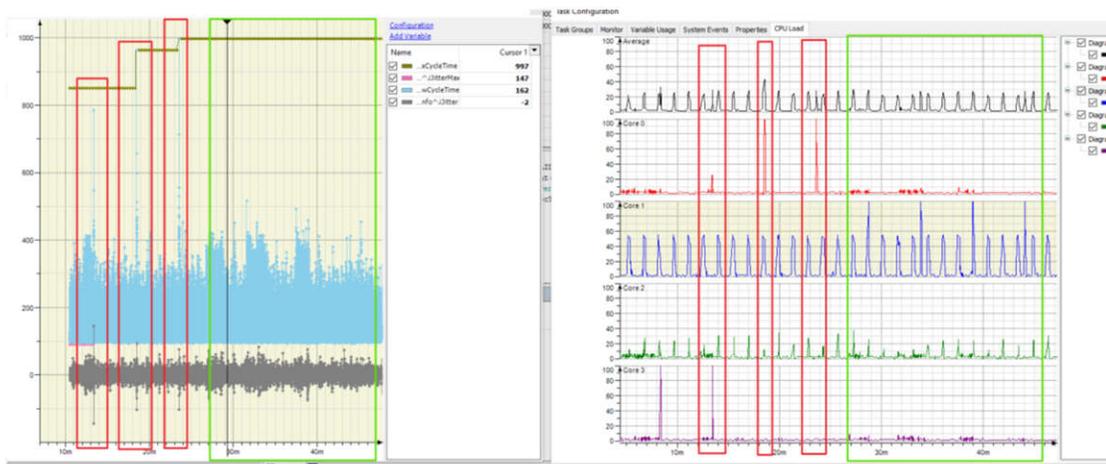


图 4-3. 绿色框中为在 CPU 1 上运行的 Codemeter 应用程序及其产生的 KPI，红色框中为在 CPU 0 或 3 上运行的 Codemeter 应用程序

另外需要注意的是，表 3-1 中报告的所有硬件平台的最大周期时间是与 EtherCAT 协议启动的大约 3 秒周期时间相关的异常值。此周期时间异常值包括设备从“Init”状态循环到“Operational”状态的几个状态的时间（基于 EtherCAT 状态机 (ESM)），启动配置通过非循环邮箱协议从 EtherCAT 控制器发送到 EtherCAT 设备（另请参阅 [EtherCAT | 系统描述](#)）。因此，性能的焦点是设备处于稳定状态时。通过 CODESYS Development System 界面注销并重新登录时，也会出现类似的循环时间峰值。在分析中排除此用例的前提是，在实际工厂自动化环境中，不太可能发生持续的注销和登录。

表 3-1 中的“过滤后的最大周期时间”和“过滤后的最大抖动”列指的是过滤掉启动异常值后的近似最大周期时间。这些过滤后的值基于最后 2 小时 46 分钟 39 秒的运行时间，因为 CODESYS Development System 的缓冲区大小仅足以存储这段时间的数据点。

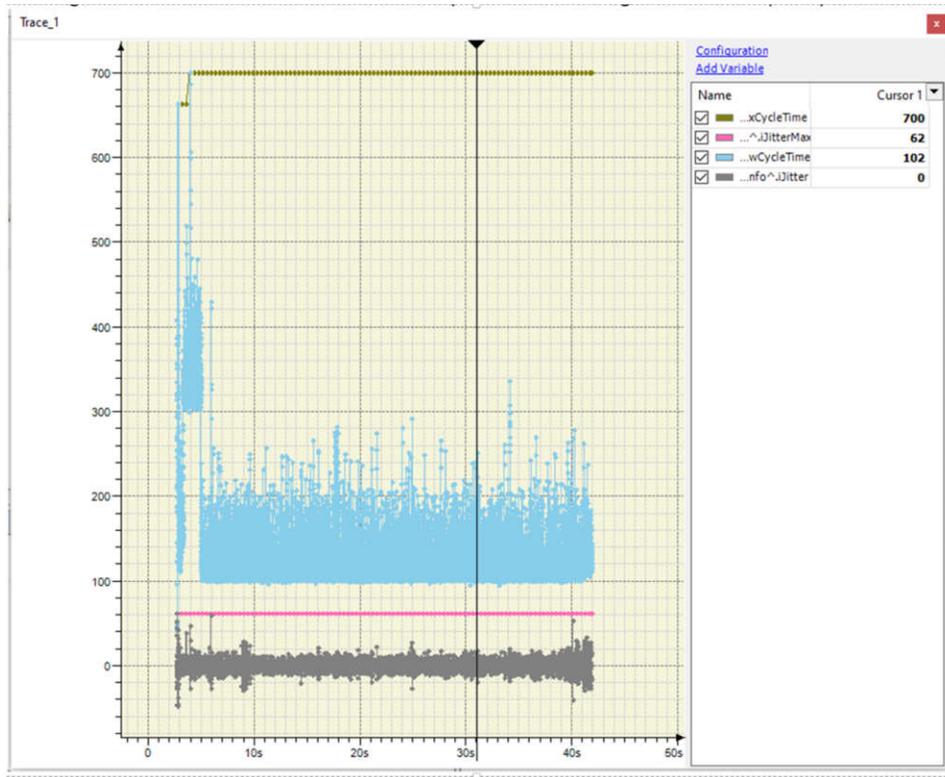
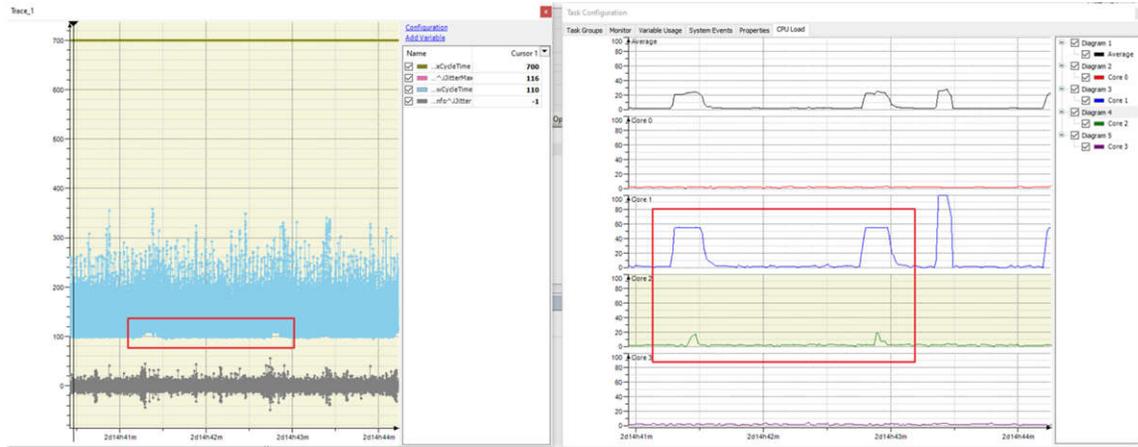


图 4-4. 作为 EtherCAT 控制器的 AM62x 上启动期间的 KPI



图 4-5. 作为 EtherCAT 控制器的 AM69 上启动期间的 KPI

AM62x 的这些优化的最终结果快照可以在图 4-6 中看到，其中最大周期时间是 $700\mu\text{s}$ 的异常值，过滤后的最大周期时间可以估计为大约 $500\mu\text{s}$ 。仍观察到一些周期性周期时间增加，有待调查。



备注

包括每 1 到 1.5 分钟增加一次周期性周期时间，如红色框所示。

图 4-6. 作为 EtherCAT 控制器的 AM62x 上 63 小时运行时间后的 KPI

总体而言，AM64x 的性能低于 AM62x，部分原因是从 4 个 A53 核心减少到 2 个 A53 核心，导致用于运行 EtherCAT 所需的各种任务和其他后台任务的核心更少。这种核心之间的差异会影响 CPU 负载，导致 AM64x 所有核心的总体 CPU 负载高于 AM62x。高速缓存大小、DDR 速度和总线宽度的差异也是造成 AM64x 和 AM62x 之间性能差异的一个因素。此外，在 AM64x 平台上观察到，在 63 小时的运行时间内，偶尔会出现 542μs 到 1032μs 之间的周期时间峰值。导致 AM64x 上出现此行为的具体原因仍在调查中。

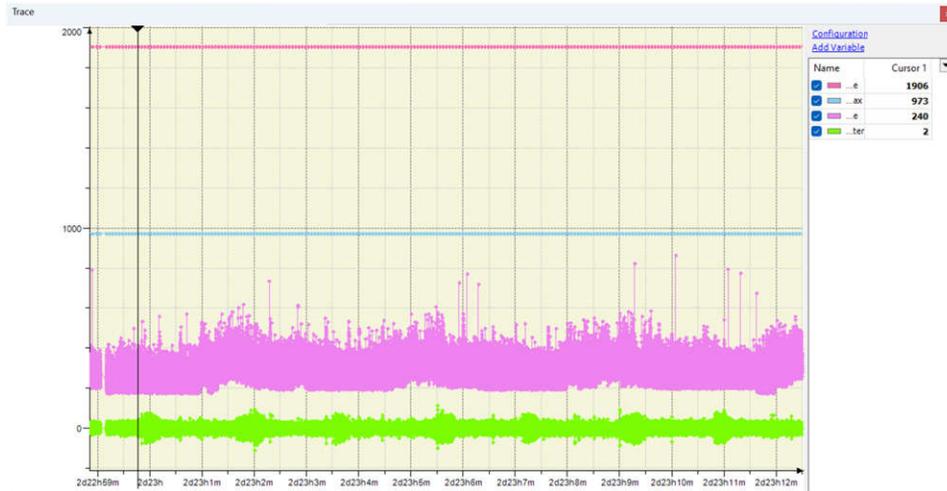


图 4-7. 作为 EtherCAT 控制器的 AM64x 上的 KPI 如 CODESYS Development System 中所示

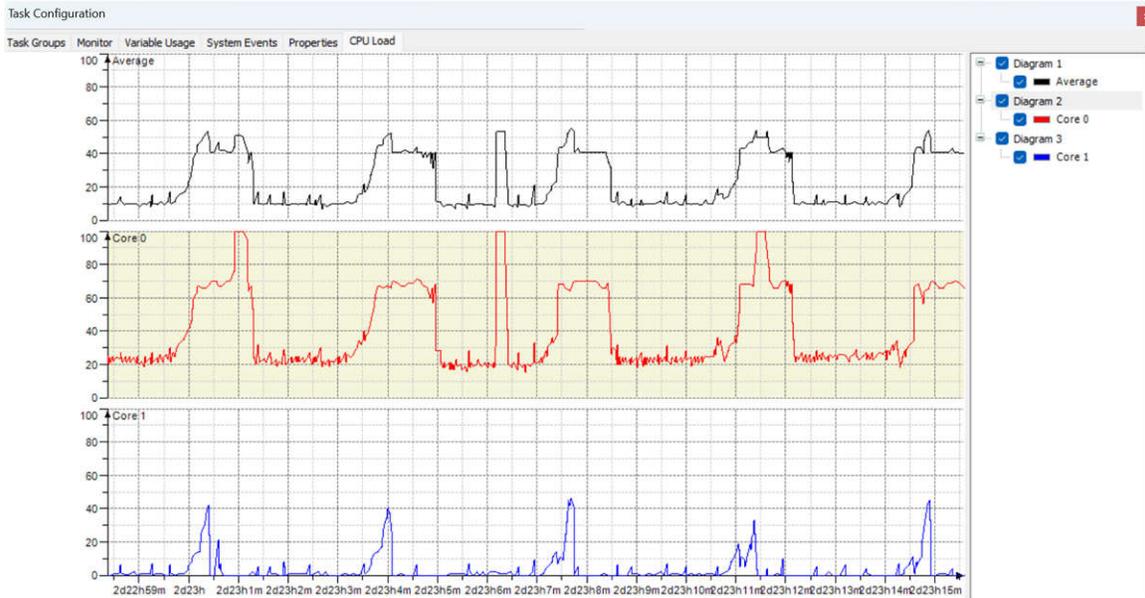


图 4-8. 作为 EtherCAT 控制器的 AM64x 的 CPU 负载如 CODESYS Development System 中所示

备注

这些是将所有 CODESYS 相关任务设置为 CPU 0 的 CPU 负载结果。

4.2 未来注意事项

如上一节所述，通过另外几个步骤可以更好地了解 CODESYS EtherCAT 协议栈影响 CPU 负载和 KPI 的原因。这些步骤尚未得到广泛调查，但在此描述为未来可能采取的步骤。其中许多步骤基于 CODESYS 针对 Linux 系统优化资源的建议。

上一节中描述的优化主要是通过实验发现的。目前已知的是，当在每个硬件平台上启动 CODESYS 应用程序时，会出现与该应用程序相关的多个线程的列表。图 4-9 显示了使用 htop 捕获的此观察示例。此 htop 是在 Codemeter 应用程序导致 CPU 负载峰值时获取的。CODESYS 文档将这些线程称为“IEC 任务”。

优先级为 -56 的线程是使用 FIFO 优先级方案调度的 EtherCAT 任务。这一点可通过以下方法验证：在 CODESYS Development System 上进行检查，并在 CODESYS Linux 系统上的任务优先级映射资源中找到的关联优先级。理解其他 CODESYS 线程（“IEC 任务”）的功能要困难一些。

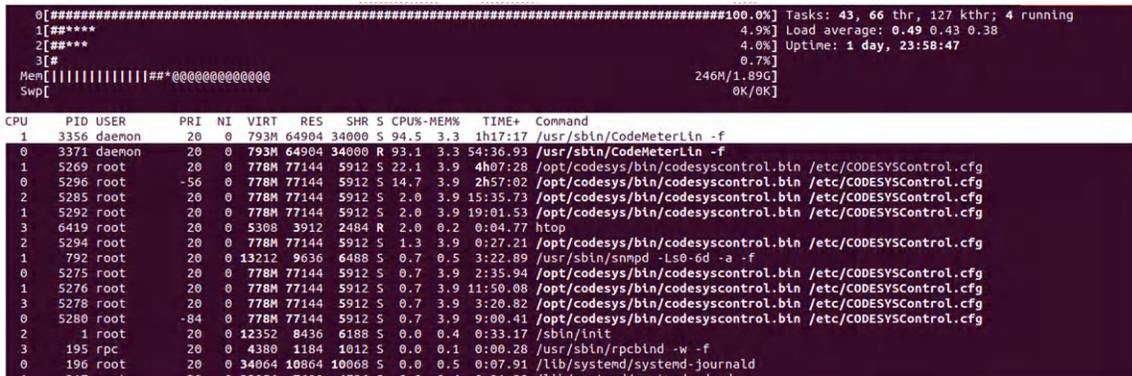


图 4-9. AM62x 上的 CODESYS 相关线程

为了获得更多可见性，可以执行以下步骤：

- 启用内核跟踪 (ftrace) 以检测中断或其他服务是否会干扰 EtherCAT 任务的性能
- 在 CODESYS Development System 中实现 CPU 负载跟踪对象，以便更好地了解 CPU 负载随时间的变化

- 在 CODESYS EtherCAT 应用程序在后台运行的同时运行 `cyclictest`，以查看各项指标是否与使用 `stress-ng` 运行的 `cyclictest` 显著不同。执行此任务的动机是查看各项指标是否与 EtherCAT 周期时间匹配，以及查看最大延迟是否以表明以太网流量很大的方式增加。

其他要尝试的实验包括以下各项。请记住，对于每个实验，首先使用 `cyclictest` 测试结果比从 CODESYS 捕获数据的更广泛过程更有益。

4.2.1 设置最大 CPU 频率

默认情况下，AM62x 的所有 4 个 A53 核心的 CPU 频率在 PROCESSOR-SDK-LINUX-RT-AM62x 的默认 wic 映像上以 1.25GHz 的频率运行；不过，AM62x 能够以最大 1.4GHz 的频率运行。使用 “`k3conf dump clock | grep A53`” 检查与每个核心关联的时钟数，将 CPU 频率更改为最高额定值；使用 “`k3conf set clock <clock number> 3 1400000000`” 检查与每个核心关联的时钟数，将 CPU 频率设置为 1.4GHz。执行此过程可提供最佳平台，以尽快处理线程并改进 KPI 指标。

4.2.2 隔离核心

由于 IEC 任务与 Linux 内核调度的其他线程竞争（请参阅 [CODESYS 任务配置](#)），这会影响到用于运行 IEC 任务的核心上的负载，因此最好将这些其他线程与用于运行 IEC 任务的核心隔离。可以使用命令行参数 `isolcpus` 将核心与内核调度隔离开来。更具体地说，可以使用 U-boot 命令 `optargs="isolcpus=<core number>"`。

4.2.3 设置 CPU 亲和性

当 IEC 任务分布在多个 CPU 核心上时，不再保证按优先级处理 IEC 任务（请参阅 [多核](#)）。为了防止这种情况，请将所有 CODESYS 线程的 CPU 亲和性设置为单个核心。可以在 Linux shell 中使用 `taskset -p <cpu core number (1 indexed)> <Process ID>` 命令来设置 CPU 亲和性。

在针对 AM64x 执行基准测试期间，通过将 `htop` 中所有可见 IEC 任务的 CPU 亲和性设置为一个核心来测试此过程。不过，每 4-5 分钟，另一个核心上的 CPU 负载仍然会增加大约 40%（请参阅 [图 4-8](#)）。在基准测试期间，包含 IEC 任务的核心上的 CPU 负载从 $\approx 20\%$ 增加到 $\approx 65\%$ 。根据增加的 CPU 负载，可能会有与在另一个核心上运行的 IEC 任务相关的额外线程。周期时间的改善很难跟踪。也就是说，没有观察到周期时间的明显改善。

4.2.4 隔离核心并设置 CPU 亲和性

此过程结合了 [节 4.2.3](#) 和 [节 4.2.2](#) 中的测试。为了更加简单，请在具有 2 个 A53 核心的 AM64x 上运行此测试。必须使用内核命令行参数 `isolcpus` 将一个核心与内核调度隔离。此外，将所有 CODESYS 线程的 CPU 亲和性设置为隔离的核心。内核调度的其他线程只应在另一个核心上运行。

4.2.5 Ksoftirqs 到 FIFO

默认情况下，与每个核心关联的 `ksoftirqs` 调度为 “TS”、“time-sliced”、“time-sharing” 或 “SCHED_OTHER” 调度策略（同一调度策略有四个不同的名称）。这是一个循环的周期时间共享策略，适用于所有不需要特殊实时机制的进程。由于 `ksoftirqs` 由与以太网相关的中断使用（这些中断由 EtherCAT 数据包使用），因此验证 `ksoftirqs` 在实时方案中以尽可能高的优先级进行调度非常重要。因此，尽可能将 “TS” 更改为 “FIFO”。换句话说，使用优先级高于 -56（EtherCAT 任务设置为此优先级）的 “SCHED_FIFO”。此过程可使用 `chrt -f -p <priority> <process id>` 命令执行。

4.2.6 增加实时调度时间

Linux 内核对具有实时调度策略的线程可以使用的总时间施加了全局限制。剩余的时间用于调度 SCHED_OTHER 进程。增加分配的实时调度时间可能有助于防止实时线程错过截止日期，包括 EtherCAT 任务线程和 `ksoftirq` 线程。在 RT-Linux 上，默认情况下，可以从 `/proc/sys/kernel/sched_rt_runtime_us` 读取实时分配的运行时间，结果为 950000（即 0.95 秒）。整个测量周期可以从 `/proc/sys/kernel/sched_rt_period_us` 读取，结果为 1000000（即 1 秒）。根据这些运行时间，默认情况下，为 SCHED_OTHER 线程保留 0.05 秒。不建议将实时运行时间完全设置为 1 秒；不过，将实时运行时间增加到 0.98 或 0.99 秒可能会有所帮助。

4.2.7 禁用 irqbalance

在 AM64x 上进行基准测试时，在 CPU 负载最重的 10 个线程列表中观察 `irqbalance`。此外，此功能会自动在 CPU 之间分配由不同中断源生成的负载，以实现更好的系统性能。不过，这可能会将负载分配给正用于运行

EtherCAT 任务和其他 IEC 任务的相同核心上，这反而会 **对 EtherCAT 任务的观察到的周期时间和抖动产生负面影响**。

4.2.8 使用独立的网络接口卡 (NIC)

在对 AM62x 进行基准测试期间，一个通用平台交换机 (CPSW) 以太网接口连接到 EtherCAT 网络，另一个 CPSW 以太网接口连接到运行 CODESYS Development System 的 PC，以查看 EtherCAT 统计信息。由于 CPSW 的设计方式是将 2 个外部端口连接到单个内部端口，因此通过 2 个外部端口的所有帧都会通过单个内部端口。这种设计阻止了在两个外部端口之间明显隔离任何以太网相关中断的可能性。因此，无法隔离数据包通过哪个 CPU 核心。控制哪个 CPU 核心处理数据包的唯一潜在方法是通过应用程序级别的线程。如果一个应用程序仅使用一个外部端口，而另一个应用程序使用另一个外部端口，则将每个应用程序的 CPU 亲和性设置为所需的 CPU 核心。不过，这种设置仍然不意味着以太网中断使用所需的 CPU 核心。从中断和 CPU 核心的角度来看，两个外部端口无法区分的另一个原因是两个 CPSW 端口都使用相同的 CPSW 以太网驱动程序。由于此策略可以使用两个单独的以太网驱动程序，因此可以通过将 CPSW 用于一个端口并将工业控制通信子系统 - 千兆位 (ICSSG) 以太网用于另一个端口来隔离两个外部端口。

4.2.9 禁用不必要的驱动程序

实时内核调优技巧指南指出，用于不必要的外围设备（如远程核心、I2C 控制器、MMC 和 USB 设备）的驱动程序会占用运行目标应用程序的 CPU 的时间。禁用设备树中的设备可防止内核加载相应的驱动程序，从而限制 CPU 核心必须处理的中断数量。

5 总结

AM62x、AM64x、AM69 和 TDA4VM 的基准测试结果表明，在过滤掉 CODESYS 启动或登录产生的结果的情况下，配置 1ms 周期时间是可以实现的。经过一些优化后，AM69 和 TDA4VM 等较高端的处理器可以实现低至 500µs 的周期时间，而 AM62x 和 AM64x 等较低端的处理器可以实现 1ms 的周期时间。此外，在最大周期时间方面，AM62x 的表现优于 AM64x，前者为 500µs，比后者约 900µs 低得多。AM62x 具有四个 A53 核心，运行时钟速度较高，为 1.4GHz，而 AM64x 只有两个 A53 核心，运行时钟速度为 1GHz。CODESYS EtherCAT 控制器应用程序使用的用于重新分配“IEC 任务”的核心数量更多，时钟速度更高，可以使周期时间缩短约 400µs。

预期是将从 **cyclicttest** 结果中观察到的最坏中断延迟与运行 CODESYS EtherCAT 协议栈中观察到的周期时间对齐。实际结果表明，CODESYS 增加了大量额外的抖动。进一步的调整（例如更改时钟频率）可以改善周期时间性能；不过，更好地了解 CODESYS 相关线程如何影响调度和 CPU 负载有助于确定如何进一步优化周期时间性能。

请注意，本应用手册提供的分析来自使用通用以太网驱动程序在 Linux 上运行的 CODESYS EtherCAT 控制器协议栈。使用本机驱动程序运行 EtherCAT 控制器协议栈可以表现出更好的性能。基于 RTOS 的 EtherCAT 控制器协议栈已经展示了这一点，该协议栈具有由 IBV 开发的优化型本机驱动程序（协议栈名称为 **icECAT**），在基于高端 MCU 的平台上运行时周期时间达到 100µs。使用更有针对性的商业协议栈（例如 IBV **icECAT**），即使使用标准以太网驱动程序，在运行 Linux 的系统上也可以实现更好的性能（另请参阅 [icECAT EtherCAT 主站协议栈基准测试](#)）。

6 参考资料

1. Acontis, [EtherCAT 主站协议栈](#), 网页
2. IBV icECAT, [适用于嵌入式系统的 EtherCAT® 主站协议栈](#), 网页
3. RealPars, [CODESYS 基础知识 | 什么是 CODESYS? 为什么它很重要?](#), 网页
4. Linux Foundation, [Cyclicttest](#), 网页
5. 开源自动化开发实验室 (OSADL) eG, [OSADL QA Farm on Real-time of Mainline Linux](#), 网页
6. 开源自动化开发实验室 (OSADL) eG, [主页](#), 网页
7. CODESYS, [CODESYS Control Standard S](#), 网页
8. Beckhoff, [EtherCAT | 系统描述 | EtherCAT 状态机](#), 网页
9. CODESYS, [优化 Linux 系统](#), 网页
10. CODESYS, [Linux 系统上的任务优先级映射](#), 网页
11. CODESYS, [CODESYS 任务配置](#), 网页
12. CODESYS, [多核](#), 网页

13. 德州仪器 (TI), [如何调整实时 Linux - 禁用不必要的驱动程序](#), 网页
14. IBV - Echtzeit, [icECAT EtherCAT 主站协议栈基准测试](#), 文档

7 附录 A : 如何使用 CODESYS 协议栈将 TI 嵌入式处理器设置为 EtherCAT 控制器

7.1 硬件要求

表 7-1 介绍了设置本应用手册中所述类似 EtherCAT 网络所需的硬件。本应用手册中所述的其他一些硬件平台可能需要类似的硬件。

表 7-1. TI 处理器作为 EtherCAT 控制器时的硬件要求

项目	数量	说明	链接
SK-AM62B-P1	0-1	AM62x 器件用作 EtherCAT 控制器	https://www.ti.com.cn/tool/cn/SK-AM62B-P1
TMDS64EVM/SK-AM64B	0-1	AM64x 器件用作 EtherCAT 控制器	https://www.ti.com.cn/tool/cn/TMDS64EVM https://www.ti.com.cn/tool/cn/SK-AM64B
SK-AM69	0-1	AM69 入门套件用作 EtherCAT 主站	https://www.ti.com.cn/tool/cn/SK-AM69
TDA4VM	0-1	TDA4VM 用作 EtherCAT 主站	https://www.ti.com.cn/tool/cn/J721EXCPXEVM https://www.ti.com.cn/tool/cn/J721EXSOMXEVM
Beckhoff EK1100	1	EtherCAT 耦合器	https://www.beckhoff.com/en-us/products/i-o/ethercat-terminals/ek1xxx-bk1xx0-ethercat-coupler/ek1100.html
Beckhoff EL2889	10	EtherCAT 终端, 16 通道数字输出	https://www.beckhoff.com/en-us/products/i-o/ethercat-terminals/el2xxx-digital-output/el2889.html
Linux PC	0-1	可选的 Linux PC 用作 DHCP 服务器	不适用
Windows PC 的易于使用的评估软件	1	运行 CODESYS Development System 的 64 位 Windows PC	不适用
以太网交换机	0-1	可选的以太网交换机, 用于将 CODESYS Development System 连接到 EtherCAT 网络	不适用
Micro USB-B 转 USB-A 适配器	0-1	在 TMDS64EVM 上将 CODESYS 许可证 USB 加密狗用作 EtherCAT 控制器时需要	不适用
CODESYS 密钥	1	用于基于 CodeMeter 技术安全存储 CODESYS 许可证的 USB 加密狗	https://us.store.codesys.com/codesys-key.html
以太网电缆	2+	用于 EtherCAT 网络的至少 2 根 CAT5 或 CAT6 以太网电缆。如果添加了以太网交换机和其他 EtherCAT 设备, 则需要更多	不适用
具有 5.5mm x 2.5mm x 9.6mm 桶形插孔的 12V (最高 8A) 电源	0-1	用于 TMDS64EVM 或 TDA4VM 的 12V 电源	不适用
5V-15V (最高 3A) 电源	0-1	用于 SK-AM64B、SK-AM62B-P1、SK-AM69 的 5V 电源	不适用

7.2 软件要求

表 7-2 描述了用于设置与本应用手册中所述相同的 EtherCAT 网络的软件。本应用手册中所述的其他一些硬件平台需要类似的软件。请注意, 表中描述的特定软件版本是经过测试和验证可以正常工作的版本。较新的版本也可以工作, 但尚未经过专门测试。

表 7-2. TI 处理器作为 EtherCAT 控制器的软件要求

项目	版本	说明	链接
AM64x/AM62x TI 处理器 SDK RT-Linux 映像	09.01.00.008	刻录到 SD 卡中的默认 RT-Linux WIC 映像	https://www.ti.com.cn/tool/cn/download/PROCESSOR-SDK-LINUX-RT-AM64X/09.01.00.08 https://www.ti.com.cn/tool/cn/download/PROCESSOR-SDK-LINUX-RT-AM62X/09.01.00.08
AM69x/TDA4VM TI 处理器 SDK RT-Linux 映像	09.01.00.07	刻录到 SD 卡的 SDK 09.01.00.07 映像的实时内部版本	请参阅脚注 (1) (2) (3)
CODESYS Development System V3	3.5.19.10 64 位	适用于工业控制和自动化技术 GUI 的编程工具	https://us.store.codesys.com/codesys.html
CODESYS Control for Linux ARM SL	4.9.0.0 64 位	基于 SoftPLC for Linux/ARM 的工 业控制器	https://us.store.codesys.com/codesys-control-for-linux-arm-sl-1.html

- (1) ti.com 上没有 AM69 或 TDA4VM RT Linux 映像版本链接。生成 RT Linux 映像需要用户执行“处理器 SDK Linux”文档的“使用 Yocto 构建 SDK”一节中概述的步骤（尤其记得看“RT 支持”小节）。
- (2) 另请参阅 [TDA4VM SDK 09.01](#) 文档
- (3) 另请参阅 [AM69 SDK 09.01](#) 文档

7.3 硬件设置

EtherCAT 网络必须设置为树形、线形或星形拓扑。也可以使用环形拓扑以实现冗余。有关本应用手册中所述的同一网络，请参阅图 2-1。

7.4 软件设置

下面提供有关如何设置 CODESYS EtherCAT 项目并使用 TI 嵌入式处理器作为 EtherCAT 控制器的说明。请注意，CODESYS 是一款商业软件。如果遇到与 CODESYS 特定设置相关的任何问题，请直接联系 CODESYS 技术支持。

7.4.1 Windows PC 设置

1. 使用 CODESYS 创建帐户
 - a. “企业客户”帐户需要获得批准后才能下载和安装任何软件
 - b. “个人客户”帐户允许下载和安装所有必需的软件
2. 下载 CODESYS Development System V3

备注

推荐：下载 .zip 版本，因为 .exe 版本可能会导致 Windows PC 在尝试运行时崩溃

3. 以管理员身份运行 CODESYS Development System V3 安装程序
 - a. 管理员权限允许安装程序安装用户尚未在其系统上安装的依赖项
 - b. 请记住，病毒防护软件可能会干扰 CODESYS 的安装。如果 CODESYS 安装失败，请尝试禁用任何病毒防护软件
 - c. 这些依赖项可能包括在图 7-1 中

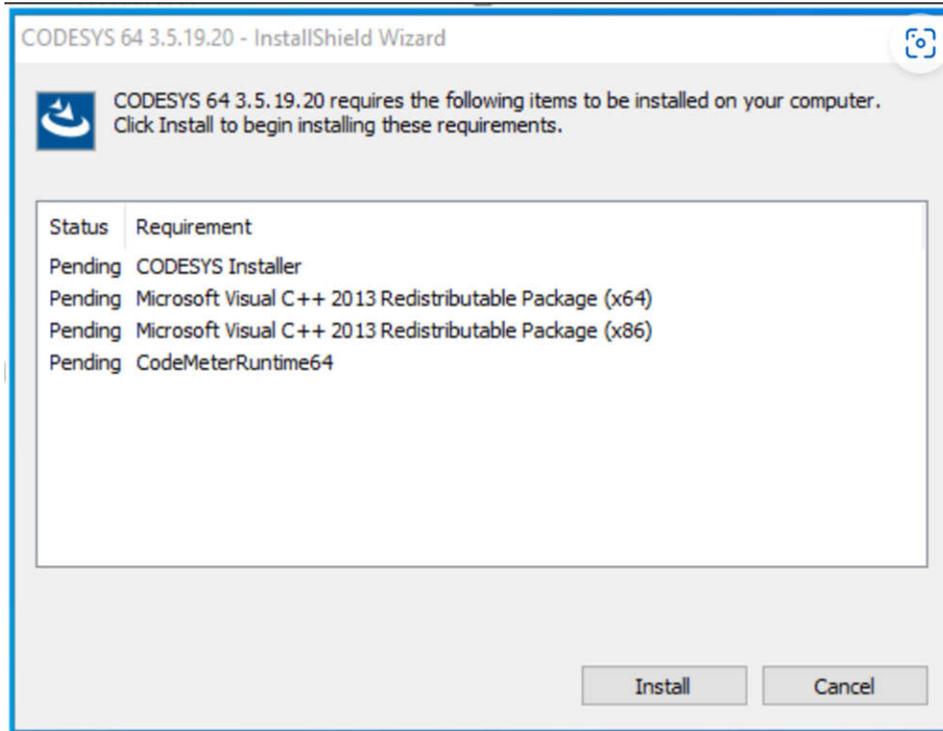


图 7-1. CODESYS Development System 安装依赖项

- d. 在未获得完全管理员权限的情况下尝试以管理员身份运行会导致 Windows PC 冻结并崩溃
4. 下载 CODESYS Control for Linux ARM SL
5. 执行以下步骤，将 CODESYS Control for Linux ARM SL 安装到之前安装的 CODESYS Development System 中：
 - a. 双击下载的 **CODESYS Control for Linux ARM64 SL <version>.package**，然后选择 **CODESYS 64 <version>**。
 - b. 点击 *Continue* 以安装软件包

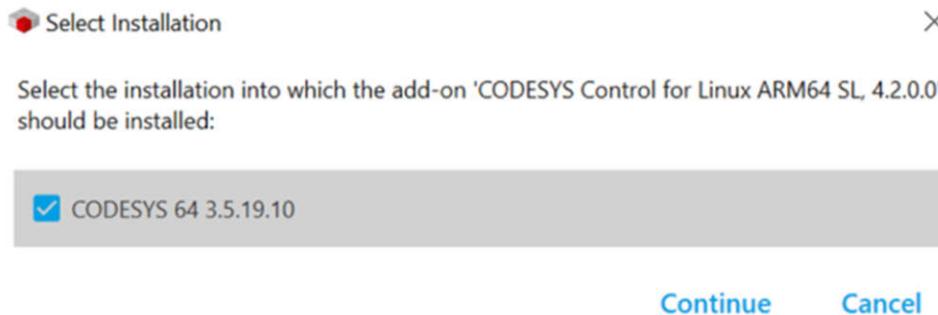


图 7-2. CODESYS 安装版本

6. 执行以下步骤，将所需的文件从 CODESYS Control for Linux ARM SL 软件包复制到要设置为 EtherCAT 控制器的设备上：
 - a. 使用 7-Zip 等实用程序解压缩 **CODESYS Control for Linux ARM64 SL <version>.package** 的内容
 - b. 导航至 **CODESYS <version>** 程序文件中的 **CODESYS Control for Linux ARM64 SL** 目录
 - c. 将下面两个文件放置在 EtherCAT 控制器根目录中的任意位置：
 - i. **codemeter-lite_<version>_arm64.deb**：位于 '**CODESYS Control for Linux ARM64 SL [<version>]/Dependency**' 目录中
 - ii. **codesyscontrol_linuxarm64_<version>_arm64.ipk**：位于 '**CODESYS Control for Linux ARM64 SL [<version>]/Delivery/[linuxarm64]**' 目录中

- d. 可以使用文件传输协议 (scp、sftp 等) 将以前的文件复制到 EtherCAT 控制器，或者可以直接将这些文件放在包含用于运行 EtherCAT 控制的操作系统 SD 卡上

7.4.2 EtherCAT 控制器设置

1. 使用 BalenaEtcher 等工具将 EtherCAT 控制器的默认 RT Linux SDK 映像刷写到 SD 卡中
 - a. 对于 AM62x/AM64x 器件
 - i. 如果器件是 GP 器件，则必须将 GP 版本的 `tiboot3.bin` 重命名为 `tiboot3.bin`，并替换原来的 `tiboot3.bin`
 - ii. 如果器件是 HS-FS 器件，则可以将 `tiboot3.bin` 保留为默认版本
 - b. 对于 AM69x 器件
 - i. 必须将默认的 `tiboot3.bin` 替换为 HS-FS 版本的 `tiboot3.bin`
 - ii. 例如：
 1. `$ cd <path-to-boot-partition>`
 2. `$ mv tiboot3-j784s4-hsfs-evm.bin tiboot3.bin`
2. 将 CODESYS Control for Linux ARM SL 软件包中的必需文件复制到要设置为 EtherCAT 控制器的设备的根目录中，并且设备已使用 SD 卡启动后，使用所复制的文件安装将设备设置为 EtherCAT 控制器所需的软件包。在目标设备上运行以下命令
 - a. `$ opkg -v2 install <path-to-file>/codemeter-lite_<version>_arm64.deb`
 - i. 对于 AM62x 器件、如果出现 “does not have a compatible architecture” 错误，请改为运行以下命令
 1. `opkg -v2 install --nodeps --offline-root / --add-arch arm64:13 <path-to-file>/codemeter-lite_<version>_arm64.deb`
 - b. `$ opkg -v2 install <path-to-file>/codesyscontrol_linuxarm64_<version>_arm64.ipk`
3. 使用以下命令之一在 EtherCAT 控制器上启动 CODESYS 应用程序
 - a. `$ /opt/codesys/bin/codesyscontrol.bin /etc/CODESYSControl.cfg`
 - b. `$ systemctl start codesyscontrol.service`
 - c. `$ /etc/init.d/codesyscontrol start`
4. 在尝试将 EtherCAT 控制器连接到 Windows PC 上的 CODESYS Development System 之前，请验证 CODESYS 应用程序是否正在运行
5. 验证 CodeMeter 应用程序 (在 EtherCAT 控制器设备上) 是否正在运行，并且 EtherCAT 控制器上已激活 CODESYS 许可证。请参阅节 8，了解如何激活 CODESYS 许可证。激活 CODESYS 许可证后，使用以下命令启动许可应用程序：
 - a. `$ systemctl start codemeter.service`

7.4.3 CODESYS Development System 项目

1. 设置一个类似于节 7.3 中所示的 EtherCAT 网络
2. 使用以下命令之一在 EtherCAT 控制器上启动 CODESYS 控制服务

```
$ /opt/codesys/bin/codesyscontrol.bin /etc/CODESYSControl.cfg
$ systemctl start codesyscontrol.service
$ /etc/init.d/codesyscontrol start
```

3. 在 Windows PC 上打开 CODESYS Development System V3
4. 通过以下步骤创建一个新的标准工程
 - a. 点击 “File > New Project”
 - b. 选择 “Standard project”
 - c. 选择工程名称和工程存储位置
 - d. 为 “Device” 条目选择 “CODESYS Control for Linux ARM64 SL (CODESYS GmbH)”
 - e. 为 “PLC_PRG in” 条目选择 “Structured Text(ST)”
 - f. 点击 “OK”

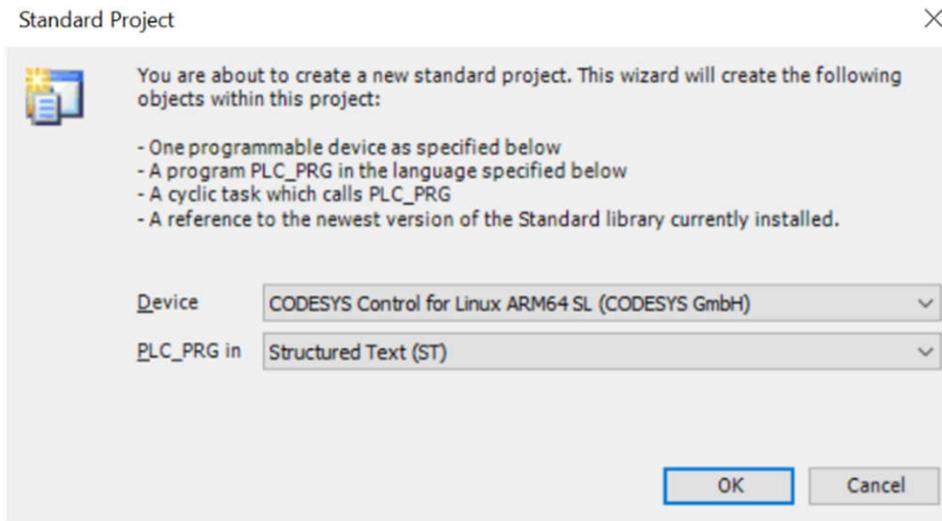


图 7-3. CODESYS 标准工程

5. 执行以下步骤将 EtherCAT 设备 ESI (EtherCAT 从站信息) 文件添加到工程
 - a. 将 AM243x/AM64x 用作 EtherCAT 设备的情况
 - b. 验证是否已安装 AM243x MCU+ SDK 或 AM64x MCU+ SDK
 - c. 点击 **“Tools > Device Repository > Install”**
 - d. 导航到并选择 (类似于 AM64x EtherCAT 设备的步骤)

```
<path-to-AM243-MCU+-SDK>/source/industrial_comms/ethercat_slave/stack/esi/'TI AM243X.R5F Simple.xml'
```

6. 点击 **“Open”**
7. 点击 **“Close”**
8. 对于其他 EtherCAT 设备 (例如 Beckhoff 设备) , 获取并下载 ESI 文件 , 然后以类似的方式安装到 Device Repository 中
9. 执行以下步骤扫描 EtherCAT 控制器设备 (“Origin” 设备)
 - a. 在 **“Devices”** 窗口中 , 双击 **“Device (CODESYS Control for Linux ARM64 SL)”**
 - b. 转至 **“Communication Settings”** , 然后点击 **“Scan Network”** 选项卡
 - c. 运行 CODESYS 控制服务的 EtherCAT 控制器设备的名称显示在 *Gateway-1* 下
 - i. 如果 EtherCAT 控制器未出现 , 请取消选中 **“Hide non-matching devices, filter by Target ID”** , 然后点击 **“Scan Network”**
 - d. 选择设备 , 然后点击 **“OK”**
 - e. 如果提示 **“user management”** :
 - i. 点击 **“Yes”**
 - ii. 创建用户名和密码
 1. 要更改用户名和密码 , 请右键点击 **“Device (CODESYS Control for Linux ARM64 SL)”** , 然后点击 **“Reset Origin Device [Device]”**
 - iii. 点击 **“OK”**
 - iv. 登录
10. 执行以下步骤添加 EtherCAT_Master 设备
 - a. 在 **“Devices”** 窗口中 , 右键点击 **“Device (CODESYS Control for Linux ARM64 SL)”**
 - b. 点击 **“Add Device”**
 - c. 展开 **“Fieldbuses > EtherCAT > Master > EtherCAT Master”**
 - d. 点击 **“Add Device”**
11. 通过匹配与其余 EtherCAT 设备连接的以太网端口的 MAC 地址 , 将您的 EtherCAT 控制器设置为 EtherCAT_Master。
 - a. 在 **“Devices”** 口中 , 双击 **“(CODESYS Control for Linux ARM64 SL)”**

- b. 点击 **“General > EtherCAT NIC Settings > Source address (MAC) > Select”** 按钮
- c. 选择连接到系统其余部分的 MAC 地址和以太网端口
12. 通过扫描设备来设置网络中的其余 EtherCAT 设备
 - a. 点击 **“Toolbar > Login (gear icon)”** 或按 **“Alt + F8”**
 - i. 如果出现显示以下内容的窗口，请点击 **“Yes”**：“No online change possible due to severe changes: Do you want to perform a download?”
 - b. 在 **“Devices”** 窗口中，右键点击 **“EtherCAT_Master (EtherCAT Master)”**
 - c. 点击 **“Scan for Devices...”**
 - d. 点击 **“Copy All Devices to Project”**
 - e. 如果 EtherCAT 设备未出现在已扫描设备列表中
 - i. 尝试对 EtherCAT 设备进行下电上电
 - ii. 尝试重新刷写电路板
13. 执行以下步骤从 EtherCAT 设备获取过程数据
 - a. 点击 **“Toolbar > Start”** 或按 **“F5”**
 - b. 在 **“Devices”** 窗口中，双击 EtherCAT 设备之一
 - c. 选择 **“General > Address > Additional > Expert settings”**
 - d. 点击 **“Expert Process Data > Load PDO from the Device”**
 - e. 为每个 EtherCAT 设备重复以下步骤以选择 **“Expert Process Data”**
 - f. 按 **“Ctrl + S”**
 - g. 点击 **“Toolbar > Stop”** 或按 **“Shift + F8”**
 - h. 点击 **“Toolbar > Logout”** 或按 **“Ctrl + F8”**
14. 可通过编写一个自定义的 PLC_PRG 程序，实现为每个设备映射 **“EtherCAT I/O Mapping”** 中的变量
 - a. 在 **“Devices”** 窗口中，双击 **“PLC_PRG”**
 - b. 请联系 CODESYS 以获得有关如何编写 PLC_PRG 程序的更多详细信息

7.4.4 执行

1. 使用以下命令之一在 EtherCAT 控制器上启动 CODESYS 控制服务

```
$ /opt/codesys/bin/codesyscontrol.bin /etc/CODESYScontrol.cfg
$ systemctl start codesyscontrol.service
$ /etc/init.d/codesyscontrol start
```

2. 打开 CODESYS 工程
3. 在 **“Devices”** 窗口中，双击 **“Device (CODESYS Control for Linux ARM64 SL)”**
4. 点击 **“Communication Settings > Scan Network”** 选项卡
5. 点击控制器器件 (AM64x 或 AM62x 器件)
6. 点击 **“OK”**
7. 登录 (如有必要)
8. 在 **“Devices”** 窗口中，双击 **“EtherCAT_Master (EtherCAT Master)”**
9. 点击 **“General > EtherCAT NIC Settings > Source address (MAC) > Select...”** 按钮
10. 选择连接到系统其余部分 (连接到 EtherCAT 网络) 的 MAC 地址和以太网端口
11. 在 **“Devices”** 窗口中，双击 **“MainTask (IEC-Tasks) > PLC_PRG”**
12. 点击 **“Toolbar > Login (gear icon)”** 或按 **“Alt + F8”**
13. 点击 **“Toolbar > Start”** 或按 **“F5”**

7.5 如何查看性能测量结果

以表格形式查看基本周期时间和抖动统计信息

1. 在 **“Devices”** 窗口中，双击 **“Task Configuration”**
2. 点击 **“Monitor”** 选项卡

查看特定于帧的统计信息

1. 在 **“Devices”** 窗口中，双击 **“EtherCAT_Master (EtherCAT Master)”**
2. 点击 **“Status”** 选项卡

以图形方式实时查看 CPU 负载测量值

1. 在 “**Devices**” 窗口中，双击 “**Task Configuration**”
2. 点击 “**CPU Load**” 选项卡。请注意，此选项卡仅在登录并启动 CODESYS 项目后出现

以图形方式实时查看周期时间和抖动统计信息

1. 添加 “**CmplecTask**” 库管理器
2. 将以下文本放在 PLC_PRG 的 “**VAR**” 部分中

```
PROGRAM PLC_PRG
VAR
  tTask : Task_Info2; (* Task Info *)
  aIecTasks : ARRAY[1..20] OF Task_Info2; (* All Task Info *)
  hCurrentTask : RTS_IEC_HANDLE := SysTypes.RTS_INVALID_HANDLE;
  Result : RTS_IEC_RESULT; (* Result Code *)

  pTaskInfo : POINTER TO Task_Info2;
  hIecTask : RTS_IEC_HANDLE;
  pResult : POINTER TO RTS_IEC_RESULT;

  ...
END_VAR
```

3. 将以下文本放在 PLC_PRG 正文中

```
...
(* Retrieve information about the current task *)
IF hCurrentTask = SysTypes.RTS_INVALID_HANDLE THEN
  hCurrentTask := IecTaskGetCurrent(pResult:=ADR(Result));
  pTaskInfo := IecTaskGetInfo3(hIecTask:=hCurrentTask, pResult:=ADR(Result));
END_IF
...
```

4. 将跟踪对象添加到 CODESYS 项目
 - a. 右键点击左侧面板上的 “**Application**”
 - b. 选择 “**Add Object > Trace**”

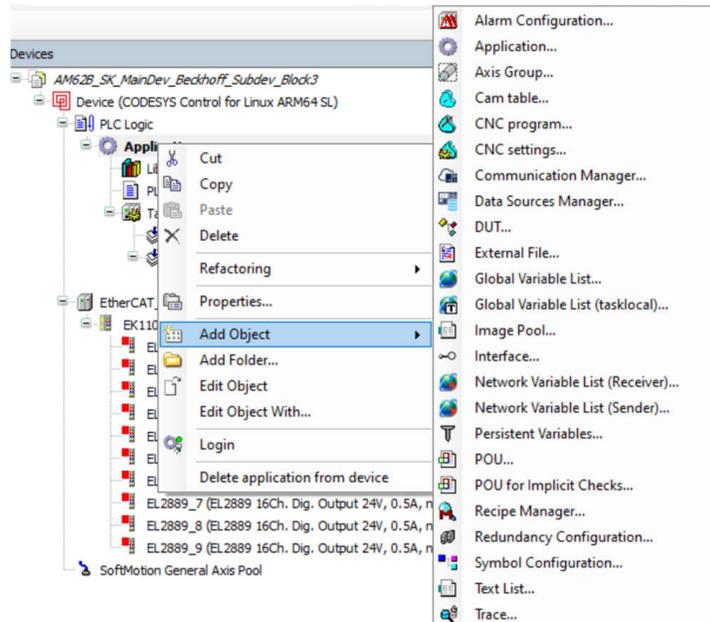


图 7-4. CODESYS 添加跟踪对象

- c. 在 “**Task for Trace Recording**” 下，选择包含 PLC_PRG 的任务
- d. 在跟踪对象中选择 “**pTaskInfo->cycle time**” 和 “**pTaskInfo->jitter**” 以进行跟踪

- i. 双击左侧面板中“**Application**”下的“**Trace**”对象
- ii. 右键点击跟踪对象图上的任意位置，然后选择“**Add Variable**”
- iii. 在“**Variable**”条目旁边，点击 3 个点
- iv. 选择与“**pTaskInfo->cycle time**”和“**pTaskInfo->jitter**”对应的变量

在跟踪对象中启动图形界面

- a. 右键点击跟踪对象图上的任意位置，然后选择“**Download Trace**”

将跟踪对象内容另存为 csv 或 txt 文件

- a. 右键点击跟踪对象图上的任意位置，然后选择“**Save Trace**”

7.5.1 附录 A 资源

TI 开发人员资源

- 德州仪器 (TI), [\[常见问题解答\] 如何使用 CODESYS Development System 建立一个带有 AM6xx 主器件和 AM64/AM243 子器件的 EtherCAT 系统](#), 博客
- 德州仪器 (TI), [CODESYS 工业通信控制器](#), 在线文档

在 CODESYS 资源中创建跟踪对象

- CODESYS, [示例：任务管理器](#), 在线文档
- CODESYS, [CODESYS 跟踪](#), 在线文档
- CODESYS, [跟踪示例](#), 在线文档
- CODESYS, [跟踪示例](#), 存储页面
- CODESYS, [将 DINT 数组转换为 STRING](#), 博客
- CODESYS, [数据类型：数组](#), 在线文档
- Tohid Alizadeh, [CODESYS：使用 OSCAT 库中的 Cycle_time 函数块的 PLC 周期时间](#), 视频
- Tohid Alizadeh, [CODESYS：使用跟踪保存和加载数据](#), 视频
- Zhou Gong, [CODESYS 教程：获取当前任务的实际周期时间](#), 博客

8 附录 B：如何在 CODESYS 协议栈上实现无限运行时间

8.1 CODESYS 许可背景

- 如果没有为 EtherCAT 控制器上的 CODESYS Runtime 系统购买 CODESYS 许可证，EtherCAT 网络将仅在演示模式下运行
- 在演示模式下运行意味着 EtherCAT 设备等现场总线设备 [会在 30 分钟后终止](#)
- 从 2023 年 12 月起，CODESYS 已迁移到跨不同硬件平台的新许可方式。新的许可证是[基于应用程序的许可证](#)
- 检查应用程序是否在演示模式下运行的快速方法是查看设备树中“EtherCAT_Master (EtherCAT_Master)”设备旁边是否有橙色圆环符号



图 8-1. CODESYS 演示模式符号

- 如果应用正在使用许可证运行，则会显示绿色圆环符号

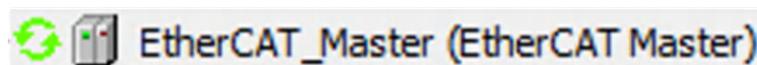


图 8-2. CODESYS 已许可模式符号

- 请注意，出于 GUI 目的在 Windows PC 上运行的 CODESYS Development System 不需要 CODESYS 许可证

8.2 获取 CODESYS 许可证

- 转到[说明基于应用程序的许可证 | CODESYS Store International](#) 并选择许可证包
- 为了进行测试和演示，建议使用的许可证如下：
 - [CODESYS Control Standard S](#) (对于单个 EtherCAT 控制器)
 - [CODESYS Control Standard L](#) (如果需要两个以太网现场总线网络)
- 此处提供了其他可用许可证的列表：[基于应用程序的许可证](#)
- 购买并下载许可证包以获取许可证代码

8.3 激活 CODESYS 许可证

8.3.1 背景

- 激活许可证有两种方法。有关这两种方法的更多详细信息，请参阅下面的链接：[产品许可 \(helpme-codesys.com\)](#)
 - 在线激活 - CODESYS Development System 需要互联网访问权限，而 EtherCAT 控制器 (目标系统) 不需要互联网访问权限
 - 离线激活 - 需要使用 CodeMeter Control Center，其随 CODESYS Development System 一起安装
- 如果 CODESYS Development System 通过直接以太网连接或通过交换机而不使用 DHCP 路由器服务来连接到 EtherCAT 网络，则无法通过 CODESYS Development System 上的 License Manager 对话框进行在线激活。
- 还必须在设备 (EtherCAT 控制器) 或 USB 加密狗上激活许可证。CODESYS 将在其上激活许可证的设备称为“容器”。设备称为“软容器”，USB 加密狗称为“加密狗”或“密匙”。
- 下面列出了在软容器上激活与在加密狗上激活之间的主要区别。可以在此 [CODESYS 密匙](#) 站点中找到相同的列表

表 8-1. 不同激活方法的优缺点

激活方法	优点	缺点
软件许可证容器 (SoftContainer)	<ul style="list-style-type: none"> • 无需额外硬件 • 无额外费用 • 无需实物交付 • 无可存储的许可证数量限制 • 许可证立即可用 • 如果加密狗被盗，许可证不会丢失 	<ul style="list-style-type: none"> • 许可证持有者独立于目标设备 • 可轻松地将许可证转移到另一个设备 • 即使设备出现故障，许可证仍然有效 • 可存储的许可证数量几乎不受限制（约 4000 个） • CODESYS 密钥还可用于安全任务（包括项目加密）
CODESYS 密钥 (加密狗)	<ul style="list-style-type: none"> • 无需将许可证传送到另一个设备 • 如果设备出现故障，许可证会丢失 	<ul style="list-style-type: none"> • 需要额外的硬件和匹配的插槽 (USB) • 密钥、运输和海关费用的额外费用（如果适用） • CODESYS 密钥送达之前等待许可证可供使用的时间 • 如果加密狗被盗，许可证将会丢失

8.3.2 建议的步骤

以下步骤详细介绍了一种已知的工作方法，它结合了在线和离线激活

- 在此处订购 CODESYS 密钥 (USB 加密狗)：[CODESYS 密钥](#)
- 获取许可证代码
- 将 CODESYS 密钥插入用于访问 CodeMeter Control Center 的 PC，并从以下位置进入 CodeMeter 服务器 [CodeMeter License Central WebDepot v23.03.554.500.ws4 \(codesys.com\)](#)：
- 打开 CodeMeter Control Center 并找到与 CODESYS 密钥关联的名称。在后续步骤中需要使用此名称。
- 转至 [CodeMeter License Central WebDepot v23.03.554.500.ws4 \(codesys.com\)](#) 并输入许可证代码
- 点击“Activate Licenses”
- 点击“CmDongle”选项
- 从“Select CmContainer”下拉菜单中，找到与 CODESYS 密钥关联的名称；此名称与 CodeMeter Control Center 中的名称相同
- 点击“ACTIVATE SELECTED LICENSES NOW”
- 现在应已在 CODESYS 密钥上激活许可证。验证许可证激活需要在 [CodeMeter License Central WebDepot](#) 中输入激活的许可证代码，以查看状态列是否为“Activated”，并且 CmContainer 列指示 CODESYS 密钥的名称。请参见下面的示例

Name	Ticket	Activated On	CmContainer	Status
CODESYS Control Standard S - (Full) Code Size 3072 I/O Channels 512 Data Sources CANopen CANopen Safety Master CANopen Safety Slave Modbus TCP Modbus Serial Profibus Instances of CAN/Modbus/Profibus (4) EtherCAT Profinet Controller/Device (C1FX) Ethernet/IP Scanner/Adapter (C1FX) Profinet Device Profinet Controller EtherNet/IP Scanner EtherCAT IQ-Link Terminal EL6224 EtherCAT ProfibusMaster Terminal EL6731 Instances of EtherCAT, EtherNet/IP, PROFINET (1) Dynamic C Code	GGPR4-BAY4G-UU9ED-C9DZZ-UL9MH	2023-11-07 13:57:17	• 3-558829	Activated
CODESYS OPC UA S - (Full) OPC UA Server (Information Models)	GGPR4-BAY4G-UU9ED-C9DZZ-UL9MH	2023-11-07 13:57:17	• 3-558829	Activated
CODESYS Visualization S - (Full)	GGPR4-BAY4G-UU9ED-C9DZZ-UL9MH	2023-11-07 13:57:17	• 3-558829	Activated

图 8-3. 已激活 CODESYS 许可证

8.4 验证已应用 CODESYS 许可证

要测试是否已将许可证应用于在 EtherCAT 控制器上运行的 CODESYS Runtime 系统，请执行以下步骤

1. 将 CODESYS 密钥 (USB 加密狗) 插入 EtherCAT 控制器中

2. 在 EtherCAT 控制器 Linux 终端上运行以下命令以启动 CodeMeter 服务的一个实例
 - `$ systemctl start codemeter.service`
3. 验证 codesyscontrol 实例是否正在 EtherCAT 控制器上运行，以将设备连接到 CODESYS Development System。使用以下命令之一启动 codesyscontrol 的一个实例
 - `$ /opt/codesys/bin/codesyscontrol1.bin /etc/CODESYSControl1.cfg`
 - `$ systemctl start codesyscontrol1.service`
 - `$ /etc/init.d/codesyscontrol1 start`
4. 将在 Windows PC 上运行的 CODESYS Development System 连接到 EtherCAT 网络
5. 打开 **“Tools→ License Manager”**
6. 选择 **“Device”**
7. 选择 **“Dongle”**
8. 选择用作 EtherCAT 控制器的设备名称，然后点击 **“OK”**
9. 许可证详细信息列表显示在左侧的 **“Licenses”** 框下面。此操作可能需要几秒钟到几分钟的时间

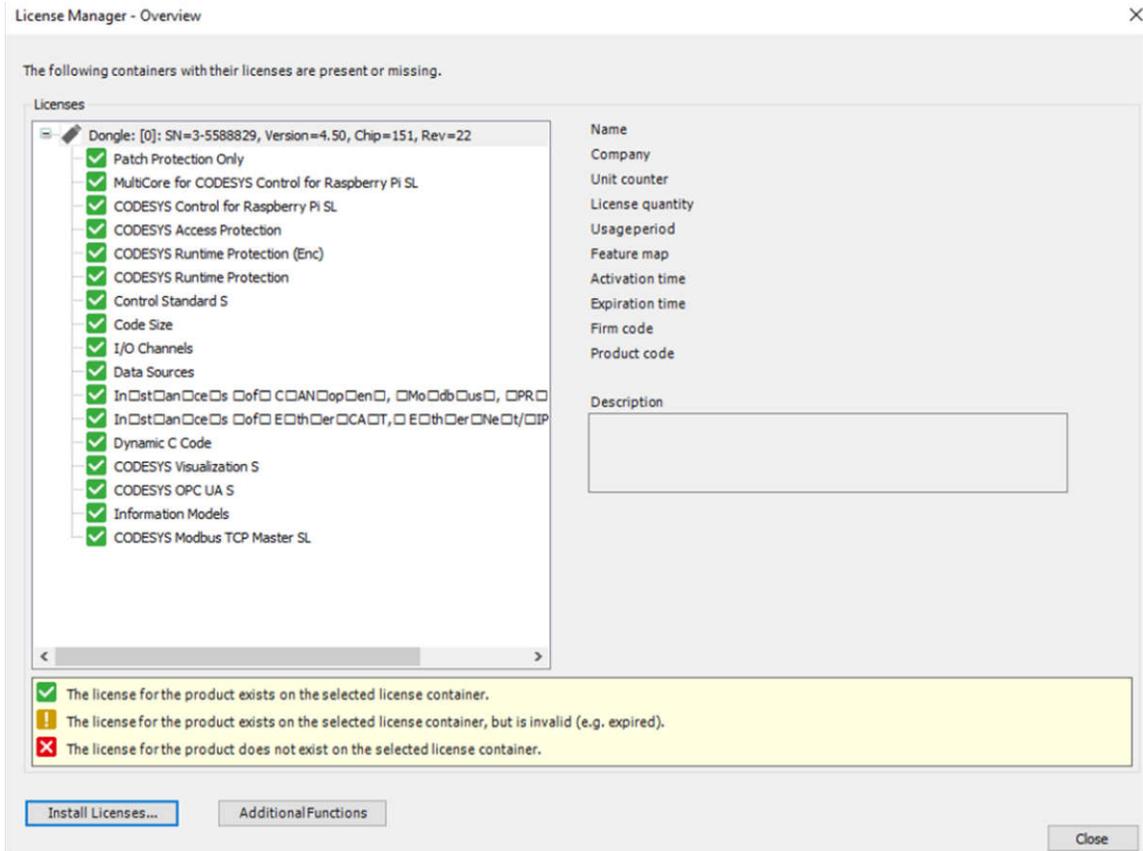


图 8-4. CODESYS Activated License 复选框

- a. 检查是否已将许可证“应用”到 EtherCAT 控制器的另一种快速方法是查看左侧面板中的 **“EtherCAT_Master (EtherCAT_Master)”** 设备旁边是否出现绿色环形符号

8.4.1 验证已应用 CODESYS 许可证的已知问题

如果验证 CODESYS 许可证失败，以下是一些修复或解决问题的技巧

- 对于在 Linux 上运行的某些 EtherCAT 控制器设备，请验证内核配置中是否启用了 HID 和 HIDRAW。这可能涉及修改现有内核配置和重建内核。
 - 有关此内核配置的更多详细信息，请参阅以下链接：[Wibu](#)、[CODESYS 密钥](#)、[加密狗：在 USB 加密狗上未找到许可证](#)

- 通过在命令行中输入“lsusb”，检查 EtherCAT 控制器设备是否注册了 USB 加密狗已连接。将出现类似于以下内容的消息

```
root@am62xx-evm:~# lsusb
Bus 001 Device 002: ID 064f:2af9 WIBU-Systems AG CmStick (HID, article no. 1001-xx-xxx)
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

- 在 EtherCAT 控制器设备 Linux 终端中运行以下命令以捕获 CodeMeter 日志。此日志显示有关 Codemeter 是否能够读取 USB 加密狗/CmContainers 的详细信息

```
cmu --cmdust --file CmDust.log
```

- 运行以下命令检查是否有与您的 USB 加密狗序列号 (3-XXXXXXX) 匹配的容器

```
cmu --list --list-content
```

在此 [CODESYS 系列文章](#)中可以找到其他一些可能有用的信息。请注意，需要登录 CODESYS 在线帐户才能查看该系列文章。

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2024，德州仪器 (TI) 公司