如何将 MIPI CSI-2 图像传感器连接至 EZ-USB™ CX3

关于本文档

范围和目的

本应用手册详细介绍了如何将 MPI CSI-2 图像传感器与 EZ-USB™CX3 连接。

目标读者

本文档适用于使用 USB 3.2 Gen 1 应用程序的任何用户： 将摄像机未压缩的数据串流 (MIPI CSI-2 图像传感器与 EZ-USB™CX3 连接)傳輸到 PC 中。

目录

目录 1
1 简介 3
2 将图像传感器连接至 CX3 5
2.1 MIPI CSI-2 接口 5
2.2 GPIF II 模块 6
2.2.1 DMA 概述 6
2.2.2 为何使用两个套接字 7
2.2.3 GPIF II 状态机 8
3 设置 DMA 系统 10
3.1 DMA 缓冲区 15
4 USB 视频类型 (UVC) 17
4.1 枚举数据 17
4.2 操作码 17
4.3 UVC 的必要條件 18
4.3.1 UVC 的 USB 描述符 18
4.3.2 UVC 特定的请求 20
5 CX3 固件 27
5.1 应用线程 28
5.1.1 USB 暂停事件 28
5.2 DMA 通道复位事件 29
5.3 处理 UVC 请求 29
5.4 初始化 29
5.5 枚举 29
5.6 配置 MIPI CSI-2 控制器 29
5.6.1 CX3 MIPI 接收器配置工具 32
5.7 配置图像传感器 33
5.8 启动视频流 33

目录

5.9 选择并切换帧设置 34
5.10 设置 DMA 缓冲区 34
5.11 在视频流期间中处理 DMA 缓冲区 34
5.12 在帧结束时恢复串流 35
5.13 终止视频流 35
6 硬件设置 36
6.1 使用 CX3 RDK 进行测试 36
6.1.1 套件购买 36
6.2 设计自己的电路板 36
7 基于 UVC 的主机应用 37
8 故障排除 38
9 总结 40
10 相關資源列表 41
文档修订记录 42

1 简介

USB 3.2 Gen 1 提供的高带宽对用于将外设连接至 USB 的各个 IC 提出很高的要求。一个流行的示例就是摄像机将非压缩数据输送到 PC 中。由于在 3.2 Gen 1 版本中，已经增大了 USB 带宽，因此连接在电脑上的摄像机的分辨率也会增加。为了满足高带宽摄像机接口的要求，一个被称为移动工业处理器接口 (MIPI) 的联盟制定了摄像机串行接口 2 (CSI-2) 规范。

本应用笔记根据赛普拉斯 EZ-USB CX3 为 MIPI CSI-2 到 USB 3.2 Gen 1 转换器提供了实现细节 (EZ-USB™ FX3 是 EZ-USB™ FX3 的一种变形，专用于该目的)。

如果您以前從未使用過 EZ-USB™產品, 請參考 Application Note “**AN75705** - Getting Started with EZ-USB™ FX3”你可以更了解這份 Application Note.

CX3 提供了 SuperSpeed USB 連接的能力, 透過 USB Video Class (UVC), 支持 MIPI CSI-2 介面的圖像傳感器。符合此类别的摄像设备能够使用 OS 内置的驱动程序进行操作，使摄像机与主机应用 (如 e-CAMView、Webcamoid、经典媒体播放器和 VLC 媒体播放器) 相互兼容。

作为 FX3 的一个派生产品，CX3 与 FX3 不一样，具体情况如下：

- 它添加了一个具有 MIPI CSI-2 接收器接口的 MIPI CSI-2 控制器。
- 清除了 USB 2.0 OTG 和充电器检测功能。
- 需要两个参考时钟：内核的 CLKIN 和 MIPI CSI-2 控制器的 REFCLK。
- 但只有频率为 19.2 MHz 的振荡器受 CLKIN 的支持。

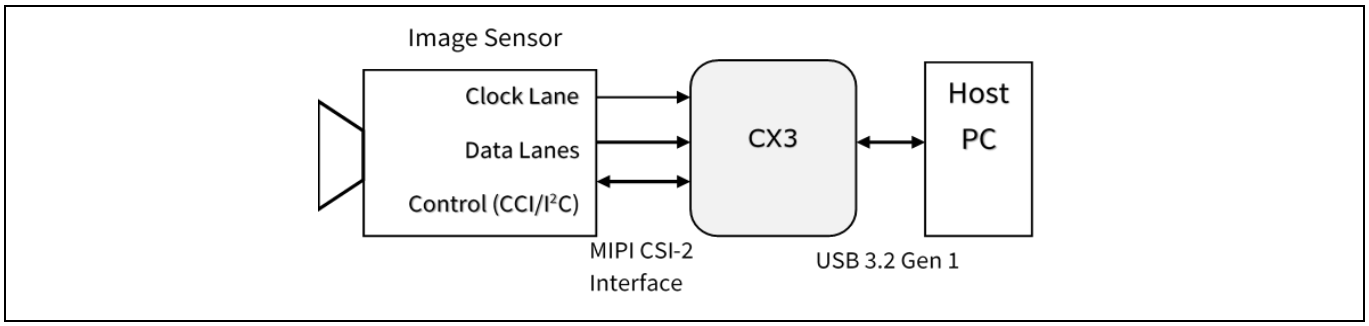


Figure 1 摄像机应用

由于 CX3 是 FX3 的专用版本，因此所有 FX3 开发工具和大部分 FX3 文档都适用于 CX3。**AN75779** 的实例体现了这种兼容性，它说明的是另外一个基于 FX3 的摄像机设计的实例 (拥有一个并行接口)。由于两个芯片中的数据传输结构是相同的，因此很多 CX3 内部工作原理材料都来自该笔记。

通过本应用笔记所提供的详细信息可帮助您理解可用的英飞凌固件项目。如果您的设计使用的图像传感器与本笔记中所述的相同，那么几乎不需要修改任何代码。如果使用了其他传感器，则可以根据该笔记进行稍微修改代码模块。

Figure 1 描述的是摄像机应用。右侧是一个装备超速 USB 3.2 Gen 1 端口的 PC。左侧显示的是一个包含了 MIPI CSI-2 接口的图像传感器，它支持下列各项内容：

- 1 到 4 个 MIPI CSI-2 数据通道
- RAW8/10/12/14、YUV422 (CCIR/ITU 8/10 位)、RGB888/666/565、压缩格式如 M-JPEG 和用户定义的 8 位、16 位和 24 位图像格式

例如，OV5640 传感器支持 VGA 60 fps、HD — 高清 (720p) 60 fps、Full HD — 全高清 (1080p) 30 fps 以及 5 M-Pixel 15 fps。以下的章節詳細介紹了 *cycx3_uvc_ov5640* 範例中 (安裝 FX3/CX3 SDK 時提供) 如何將 YUV 格式的串流, 轉換成 VGA、HD 和 Full HD 的解析度。

简介

Figure 2 显示的是带有编号的框图子模块。另外还说明了每个子模块所执行的任务。

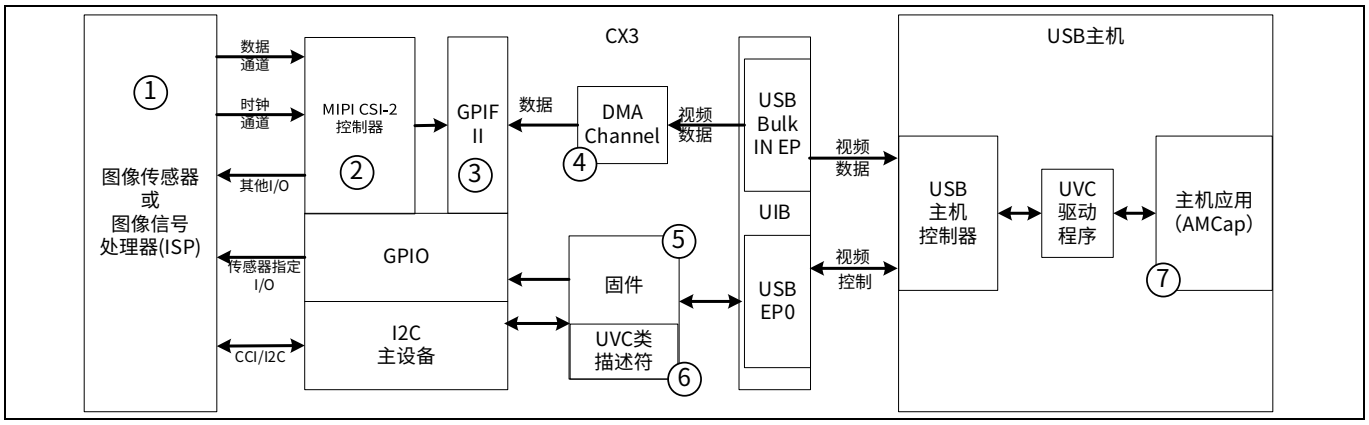


Figure 2　　系统框图

1. 基于 MIPI CSI-2 的图像传感器与 CX3 相连接，并且通过使用摄像机控制接口 (CCI) 总线配置该传感器。
2. 配置 CX3 上的 MIPI CSI-2 控制器，以读取该传感器中的图像数据，分解该数据并将它发送到 GPIF II 模块。请参考**第 2.1 节**。
3. 根据图像数据格式配置 GPIF II 模块。请参考**第 2.2 节**。
4. 构建一个 DMA，它会将图像数据从 GPIF II 模块转移到 USB 接口模块 (UIB)。在本应用中，必须将头数据添加到图像传感器内的视频数据中，以符合 UVC 规格的要求。因此应配置 DMA，以允许 CPU 将所需的头数据添加到 DMA 缓冲区内。必须设计该通道以让最大带宽满足将视频从图像传感器输送到 PC 的要求。请参考**第 3 节**。
5. CX3 固件初始化 CX3 的硬件模块 (**第 5.3 节**)，配置图像传感器(**第 5.7 节**)和 MIPI CSI-2 控制器 (**第 5.5 节**)，枚举器件成为一个 UVC 摄像机 (**第 4.3.1 节**)，处理 UVC 特定请求 (**第 4.3.2 节**)，通过 CCI (I^2C) 接口将视频控制设置 (如亮度) 传输给图像传感器，将 UVC 头数据添加到视频数据流 (**第 4.3.4 节**)，以及将带有头数据的视频数据提交给 USB (**第 5.10 节**)。
6. 提供正确的 USB 描述符，保证主机能够识别符合 UVC 的外围设备。请参考**第 4 节**。
7. 主机应用 (如 e-CAMView、Webcamoid、经典媒体播放器或 VLC 媒体播放器) 将存取 UVC 驱动程序，以通过 UVC 控制接口配置图像传感器，并通过 UVC 流接口接收视频数据。请参考**第 7 节**。

如果將 Camera 插入 USB2.0 接口, CX3 firmware 會利用 CCI bus 來選擇, 以降低的幀速率和幀大小 (如果可用) 用來適應較低的 USB 帶寬。PC 主机可以选用视频控制接口以将亮度、对比度、色调、饱和度、曝光、自动对焦和 PTZ (平移、倾斜和缩放) 等调整内容传输给摄像机。

2 将图像传感器连接至 CX3

要想将 MIPI CSI-2 图像传感器连接至 CX3，并从该器件读取数据，您必须了解 CSI-2 接口和 CX3 的 DMA 功能。

2.1 MIPI CSI-2 接口

由于摄像机的应用比较复杂，因此更加需要具有更高分辨率的图像传感器。该要求会加大并行图像传感器接口的极限值，而这些接口可能难以扩展，并需要多个互联。因此，移动工业处理器接口 (MIPI) 联盟制订了摄像机串行接口 2 (CSI-2) 标准，以提供功能强大、低功耗且高速的标准串行接口，用于支持各种图像解决方案。

MIPI CSI-2 接口是一个带有数据及时钟信号的单向差分串行接口。每次可以传输四个数据通道，数据速率可达 1 Gbps。

用于配置图像传感器的控制接口与 I^2C 标准相兼容，并将该接口称为摄像机控制接口 (CCI)。

在 CX3 中的 MIPI CSI-2 控制器提供了这两种接口以及图像传感器通常需要的各附加信号。

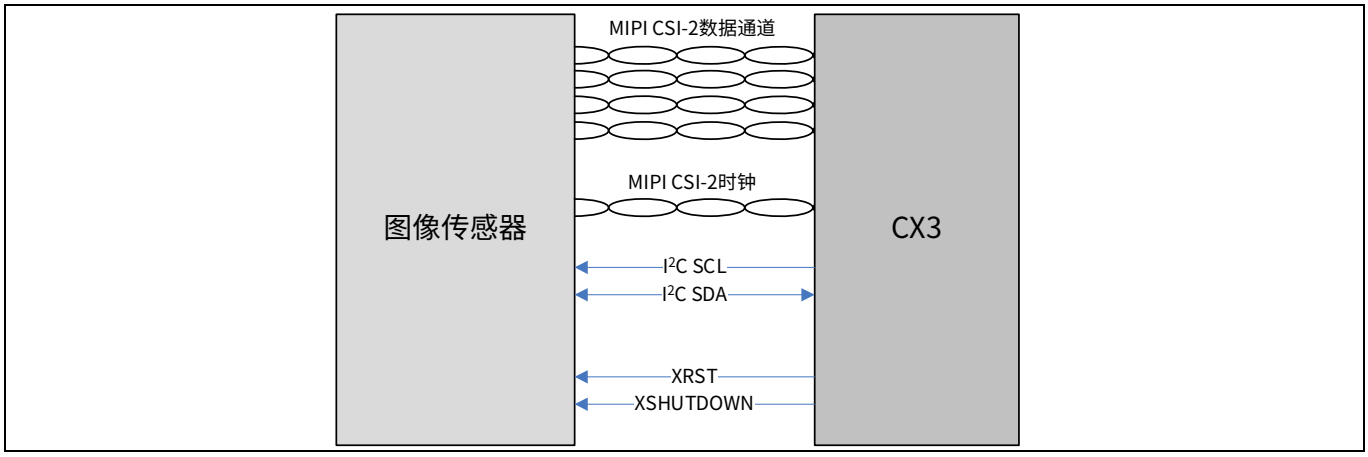


Figure 3 带有图像传感器的 CX3 接口

该接口中所显示的的三个附加信号包括：

XRST：配置前，图像传感器通常需要 CX3 中的复位信号。通过使用 CX3 的 XRST 引脚可以实现此复位。

XSHUTDOWN：当主机不需要图像传感器中的视频流时，可关闭传感器或使其进入待机模式以降低功耗。通过使用传感器中的关闭引脚完成此操作，使用 CX3 XSHUTDOWN 引脚可以控制该关闭引脚。

Note: FX3 SDK 中的 ***CyU3PMipicsiSetSensorControl*** *API 可用於將 XRES 和 XSHUTDOWN 信號傳輸到傳感器。*

MCLK：Master (or reference) Clock 圖像傳感器所需的主 (或參考) 時鐘，由外部時鍾振盪器提供。CX3 提供的 MCLK 僅用於測試圖像傳感器。此信號不適合使用在最終產品。做為最終的產品, 必須使用外部時鐘產生器作為傳感器的時鐘輸入.

關於 CSI-2、CCI 和三個附加信號的引腳映射, 請參閱數據表, **EZ-USB™ CX3 MIPI CSI-2 to SuperSpeed USB Bridge Controller**。此外，请参见 **CX3 RDK 原理图**了解一个完整的示例。

将图像传感器连接至 CX3

可配置该 MIPI CSI-2 控制器以支持一至四个数据通道，不同的数据格式 (如 RAW、YUV 或 MJPEG) 以及不同的摄像机分辨率。有关详细信息，请参考**第 5.5 节**中的内容。此应用通过使用一个配置传感器为四个数据通道提供 YUV 图像数据。

配置后，MIPI CSI-2 控制器接受来自图像传感器的串行图像数据，分解该数据，然后将其转换成并行数据以通过并行接口发送。此接口使用以下各信号：

- FV：帧有效 (表示帧的开始和结束)
- LV：行有效 (表示行的开始和结束)
- PCLK：像素时钟
- 数据：图像数据的 8 位、16 位和 24 位数据总线

这些信号的时序同 **AN75779** 的第 3.1 节中所描述的并行接口相类似。

CX3 支持的最大流通量为 2.4 Gbps，因為 CX3 中可使用的最大 GPIF II 數據總線寬度為 24-bit，支持的最大 PCLK 為 100 MHz。

CX3 MIPI configuration utility 中的相機串行接口 (CSI) 時鐘設置, 被設置為 DDR 模式 (數據在兩個時鐘邊沿進行採樣)。下表列出了每個 MIPI 通道設置支持的最大 CSI 時鐘以及每個配置支持的最大流通量。

MIPI data 通道數量	最大允許設置的 CSI 時鐘頻率	MIPI 通道的最大資料比率	最大支持的 bit 率
1	500 MHz	1 Gbps	1 Gbps
2	500 MHz	1 Gbps	2 Gbps
3	400 MHz	800 Mbps	2.4 Gbps
4	300 MHz	600 Mbps	2.4 Gbps

Note: *使用 CSI 時鐘計算出來的 CX3 支持的最大 bit 率 (CSI clock x 2 x number of MIPI lanes) 也包含了 blanking time。*

2.2 GPIF II 模块

将 MIPI CSI-2 控制器中的并行输出接口连接到 GPIF II 模块，该模块是处理器接口模块 (PIB) 的一部分。

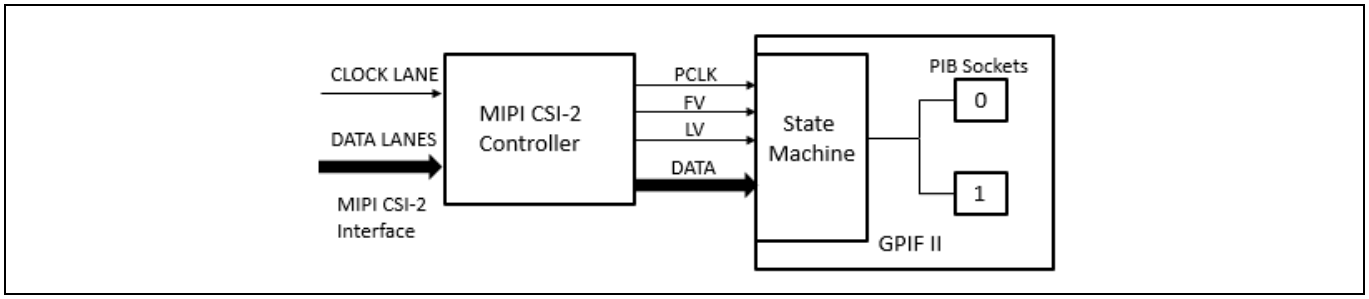


Figure 4 GPIF II 模块

GPIF II 模块使用了一个状态机将该接口中的数据连接到两个 DMA 套接字。要想知道数据传输如何发生，需要了解 CX3 的 DMA 功能。

2.2.1 DMA 概述

通过 CX3 的 DMA (直接存储器访问) 可以在两个模块之间快速进行不间断的数据传输。想要了解数据传输如何发生，需要明白下面的术语：

- 套接字 (Socket)
- DMA 描述符
- DMA 缓冲区

套接字是外设硬件模块和 CX3 RAM 之间的连接点。CX3 上的每个外设硬件模块 (如 USB、GPIF、UART 和 SPI) 具有与本身相关的固定套接字数量。输出给外设的独立数据数量等于该外设上的套接字的数量。套接字的实现包括一组寄存器，用于指向有效的 DMA 描述符，并使能或置位与该套接字相关的中断。

DMA 描述符是一组位于 FX3 RAM 中的寄存器。它保存了 DMA 缓冲区的地址和大小，以及指向下一个 DMA 描述符的指针。这些指针构建成了 DMA 描述符链。

DMA 缓冲区是 RAM 的一部分，用于存储通过 CX3 器件传输的中间数据。通过 CX3 固件，可将部分 RAM 空间作为 DMA 缓冲区使用。这些缓冲区的地址被存储为 DMA 描述符的一部分。

2.2.2 为何使用两个套接字

在了解为何使用两个套接字前，必需更加详细地了解套接字。

套接字可通过各个事件来互相发出信号，或者它们可通过中断向 CX3 CPU 发出信号。这些操作是由固件配置的。例如，考虑将数据流从 GPIF II 模块传输给 USB 模块。GPIF 套接字可以通知 USB 套接字它已经向 DMA 缓冲区填充了数据，或者 USB 套接字可以通知 GPIF 套接字它已经清空了该 DMA 缓冲区。该操作被称为*自动 DMA 通道*。当 CX3 CPU 无需修改数据流中的任何数据时，通常会使用自动 DMA 通道实现。

或者，GPIF 套接字可以向 CX3 CPU 发送一个中断，以通知 GPIF 套接字已经填充了 DMA 缓冲区。CX3 CPU 可将该信息反馈给 USB 套接字。USB 套接字会向 CX3 CPU 发送一个中断，通知 USB 套接字已经读空了 DMA 缓冲区。此时，CX3 CPU 可以将此信息反馈给 GPIF 套接字。该操作被称为*手动 DMA 通道*。

如果 CX3 CPU 需要添加、删除或修改数据流中的数据，通常需要执行该手动 DMA 通道操作。在本应用笔记中描述的固件示例使用了手动 DMA 通道操作，因为该固件需要添加 UVC 视频数据标头。

将数据写入 DMA 缓冲区内的套接字被称作*发送套接字*。从 DMA 缓冲区内读取数据的套接字被称作*接收套接字*。套接字使用存储在 DMA 描述符上的 DMA 缓冲区地址、DMA 缓冲区大小和 DMA 描述符链的值来管理数据。

套接字写满或读空 DMA 缓冲区后，需要经过一段时间 (几微秒) 从一个 DMA 描述符转移到另一个描述符。转换过程中，套接字不能传输数据。这个延迟在 GPIF II 块中使用两个 GPIF 线程来克服。

GPIF 线程是 GPIF II 模块内的专用数据路径，用来将数据引脚同套接字连接起来。但每次只有一个 GPIF 线程能够传输数据。

GPIF 线程选择机制同 MUX 一样。切换有效的 GPIF 线程时会切换用于数据传输的有效套接字，从而改变用于数据传输的 DMA 缓冲区。该切换的延迟等于一个时钟周期，使其基本上是瞬间发生的。在 DMA 缓冲区边界上实现对 GPIF II 状态机的切换，这样可以屏蔽在切换到新的 DMA 描述符时 GPIF 套接字所产生的延迟。这样，当 DMA 缓冲区为满时，GPIF II 模块可以使用传感器中的数据。

Figure 5 显示的是本应用程序中使用的各个套接字、DMA 描述符和 DMA 缓冲区连接以及数据流。使用两个 GPIF 线程填充备用的 DMA 缓冲区。这些 GPIF 线程使用了单独的 GPIF 套接字 (作为生产套接字使用) 和 DMA 描述符链 (描述符链 1 和描述符链 2)。USB 套接字 (作为消耗套接字使用) 使用了第 3 个 DMA 描述符链 (描述符链 3) 按正确顺序读取数据。有关套接字、套接字切换和相关延迟的详细信息，请参考**第 3 章**。

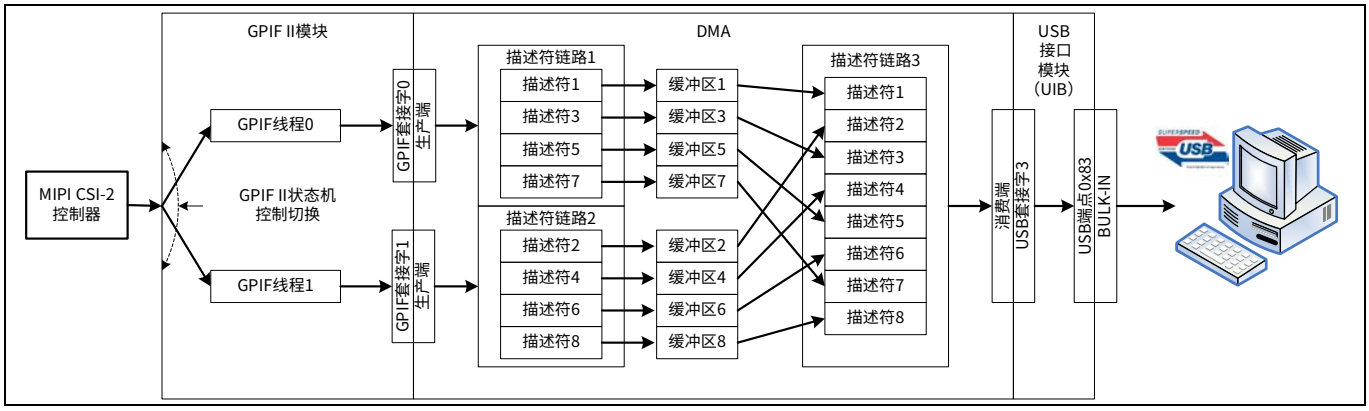


Figure 5　　CX3 数据传输架构

2.2.3　GPIF II 状态机

GPIF II 模块内部使用了一个状态机从 MIPI CSI-2 控制器的并行输出接口读取视频数据。

正如上一节 (**第 2.2.2 节**) 所描述的情况，该状态机将视频数据加载到两个套接字上，以便在各个 DMA 描述符间切换时防止丢失数据。

这样就可以使用计数器来跟踪读入套接字的数据量，然后当该计数器达到其极限值时，它将切换到其它套接字。使用计算机的限定值来设置 DMA 缓冲区的大小。

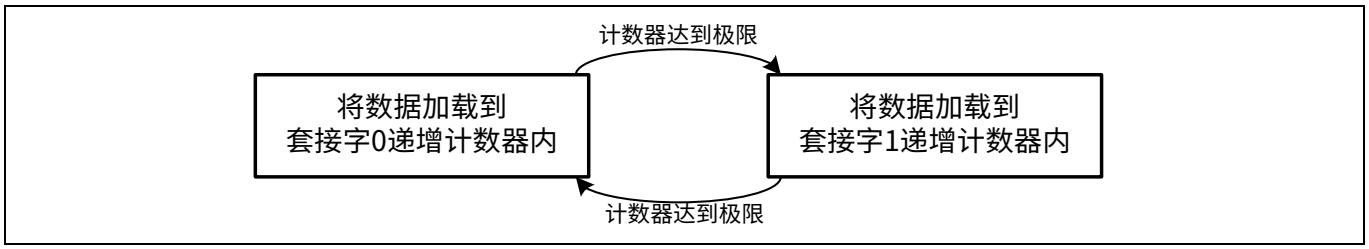


Figure 6　　将数据分配给两个套接字进行传输

每经过一个时钟周期，计数器的值都会加 1。因此，根据接口的数据总线宽度，计数器的限定值会不一样。例如，如果数据总线的宽度为 16 位、DMA 缓冲区大小为 16 KB (即 16,384)，那么每个周期会读取两个字节的数据，因此编程限定值应该为(16384/2) – 1 = 8191。

一般情况下，DMA 缓冲区的计数限制值为：

$$count = \left(\frac{producer_buffer_size(L)}{data_bus_width\ (in\ bytes)} \right) - 1$$

总线宽度的注释：

必须根据图像传感器中的数据格式选择 GPIF II 数据总线宽度。Resolution 的寬度 (或行大小) 單位為 Byte, 必需能被 GPIF II 總線寬度整除。在本应用笔记中，传感器被配置为输出 16 位 YUV422 数据，因此，数据总线宽度被配置为 16 位。有关每个图像格式的总线宽度的详细信息，请参考 **FX3 SDK API 指南**中 CyU3PMipicsiDataFormat_t 节的内容。

发送一帧数据后，状态机将中断 CPU 以指出已成功完成发送该帧，从而允许它添加标题并执行发送其它帧。

将图像传感器连接至 CX3

下一节将介绍用于数据流和支持 UVC 的固件的 DMA 通道。

關於 image line 大小的說明：

Resolution 的寬度或行大小, 單位為 Byte, 必須是 4 的倍數。如果不滿足此條件，CX3 的 CSI-2 MIPI 控制器模塊將在行數據添加額外的填充字節。

3 设置 DMA 系统

GPIF II 模块的工作频率可达 100 MHz，并且数据总线宽度为 24 位 (300 MBps)。为了将数据传输到内部 DMA 缓冲区内，GPIF II 使用两个 GPIF 线程连接到 DMA 生产套接字 (如**第 2.2.2 节**中所述)。该应用使用了套接字和 GPIF 线程的默认映射 (**Figure 7**) — 套接字 0 与 GPIF 线程 0 相连，套接字 1 与 GPIF 线程 1 相连。请注意，四个可用线程中只有两个可用于这种情况。在上一章中介绍的 GPIF II 状态机内实现了 GPIF 线程切换。

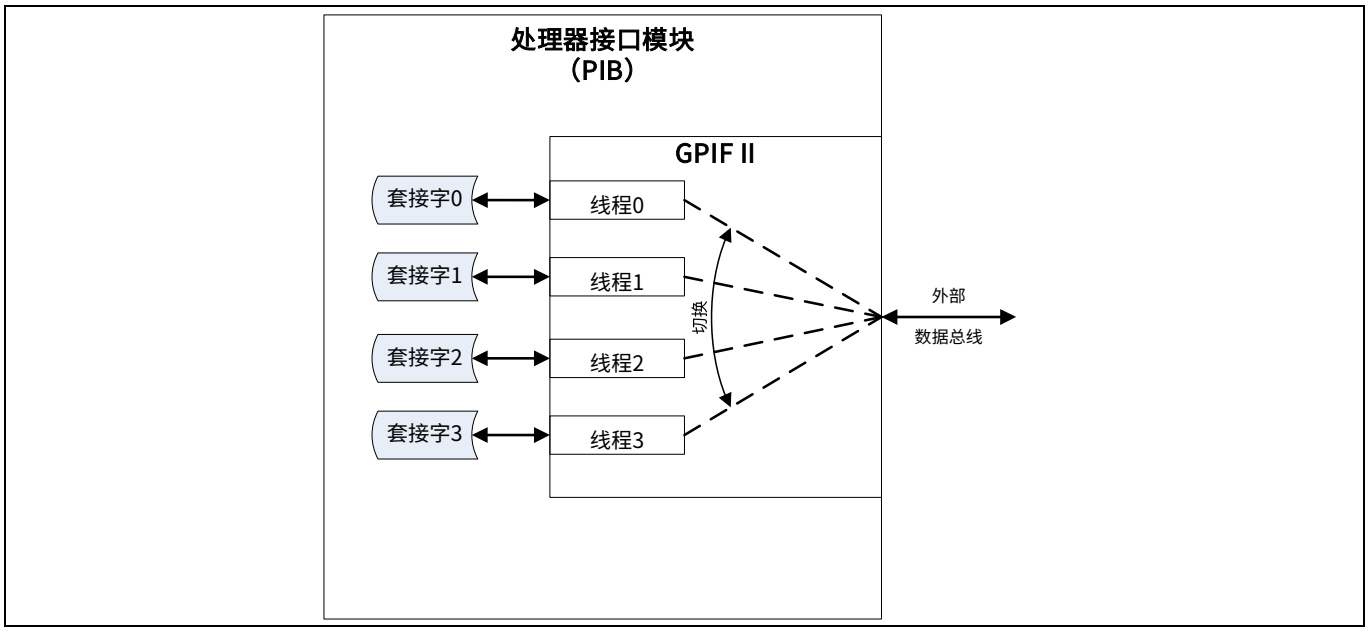


Figure 7 GPIF II 套接字/线程的默认映像

为了理解 DMA 传输，下面四个图中继续使用了在**第 2.2.1 节**中初次介绍的套接字概念。**Figure 8** 显示的是两个主套接字属性、一个链接列表及一个数据路由器。

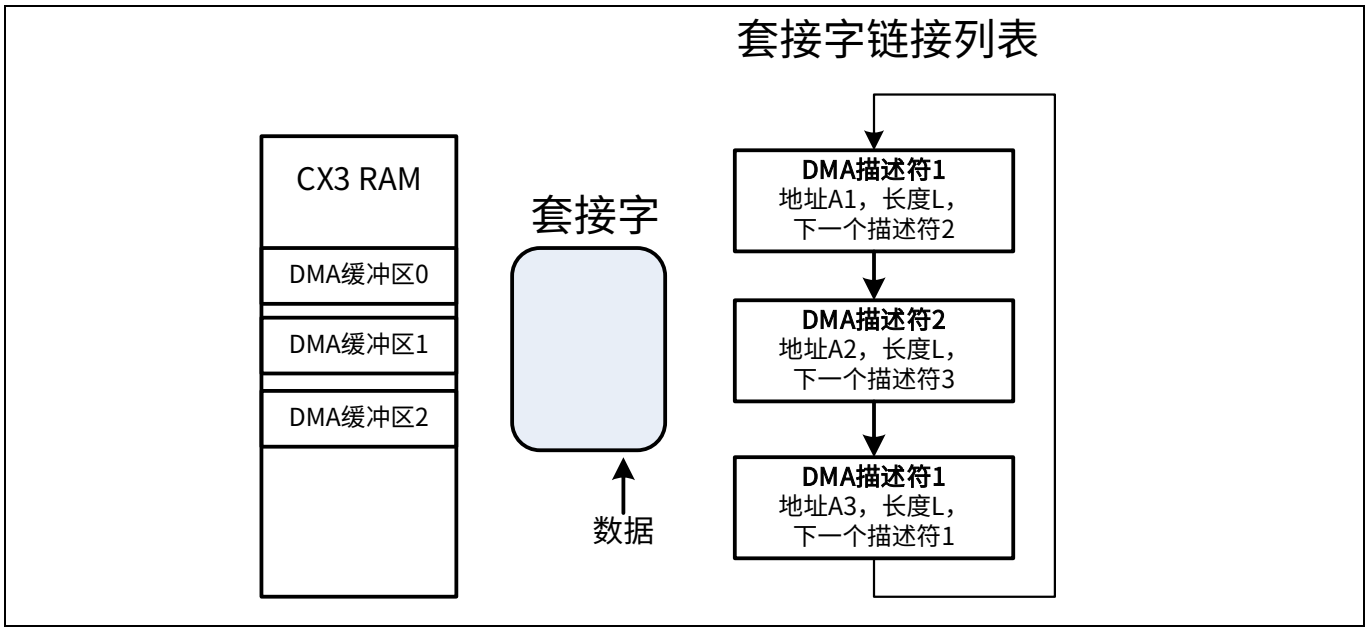


Figure 8 套接字根据 DMA 描述符列表路由数据

设置 DMA 系统

套接字链接列表是主存储器中的一组数据结构，这组数据结构又被称为 DMA 描述符。每个描述符指定了 DMA 缓冲区的地址 (An) 和长度 (*L*)，以及指向下个 DMA 描述符的指针。套接字运行时，每次仅检索各 DMA 描述符中的一个描述符，这样可以将数据路由到描述符地址和长度所指定的 DMA 缓冲区。传输 *L* 个字节后，该套接字会检索下一个描述符，并继续将各字节传输到另一个 DMA 缓冲区内。

该结构使套接字变得非常灵活，因为可以在存储器中的任何位置上创建任何 DMA 缓冲区数量，而且这些缓冲区可以自动被链接在一起。例如，**Figure 9** 中的套接字以重复循环检索 DMA 描述符。

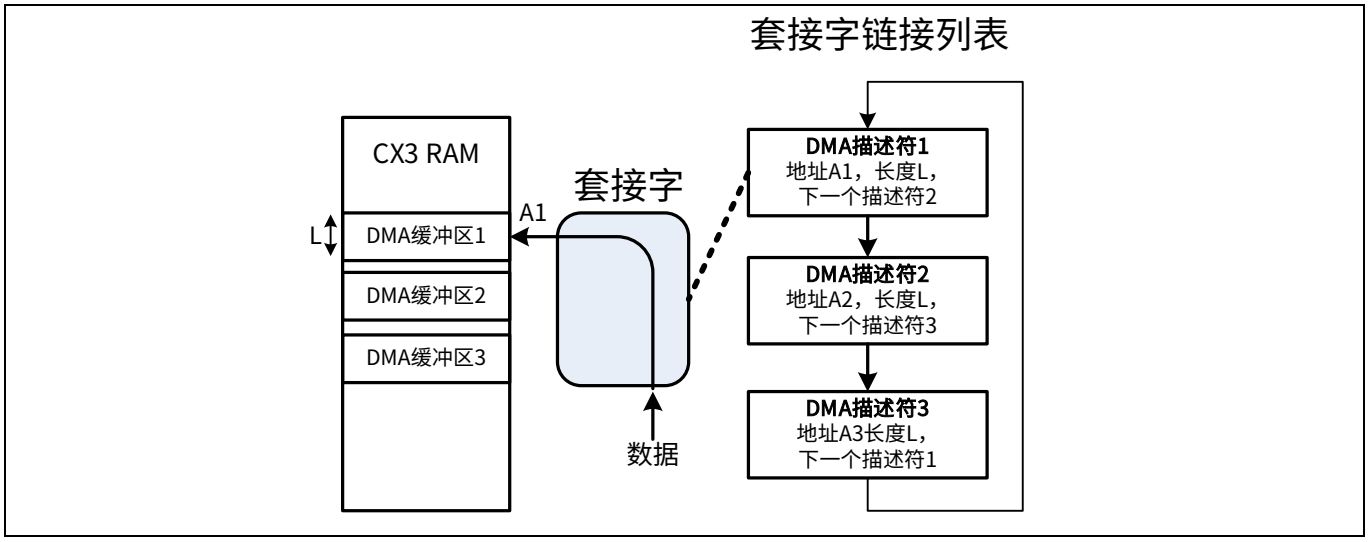


Figure 9　　使用 DMA 描述符 1 运行的套接字

在 **Figure 9** 中，该套接字加载了 DMA 描述符 1，该描述符会通知套接字将字节传输到起始地址为 A1 的 RAM 内，直到传输完 *L* 字节为止。这时，它将检索 DMA 描述符 2 并对其地址和长度设置 (分别为 A2 和 *L*) 进行相同的操作 (**Figure 10**)。

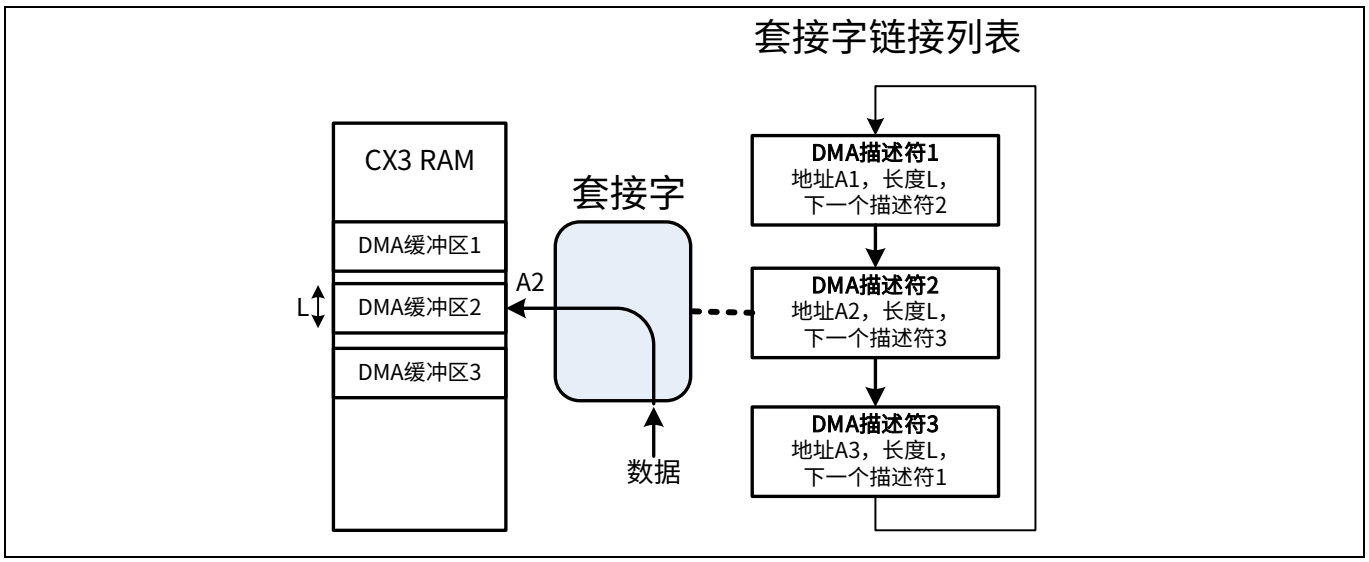


Figure 10　　使用 DMA 描述符 2 运行的套接字

设置 DMA 系统

在 **Figure 11** 中，该套接字检索第三个 DMA 描述符，并传输从 A3 地址开始的数据。传输 *L* 字节后，将使用 DMA 描述符 1 重复该序列。

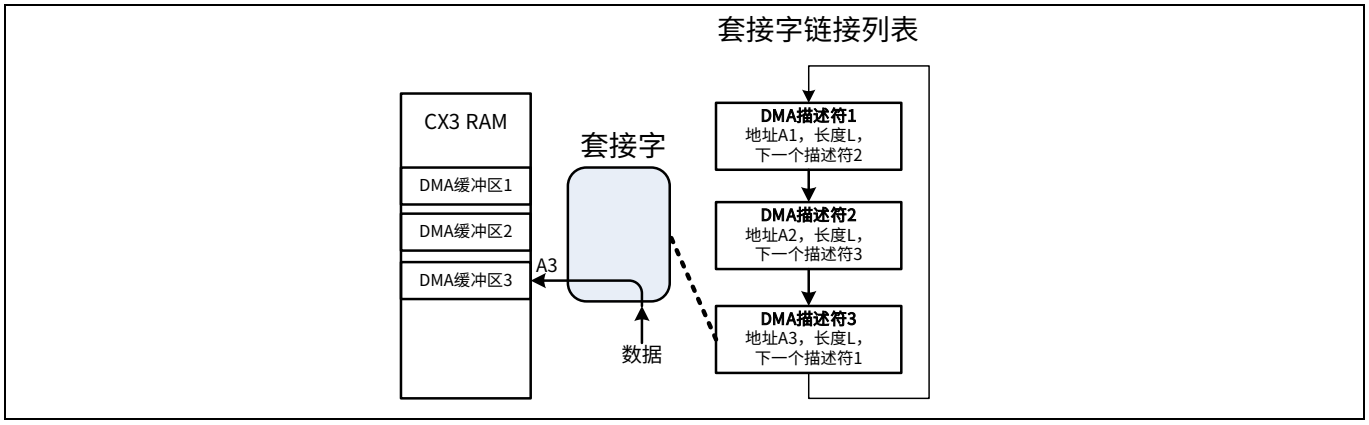


Figure 11　　使用 DMA 描述符 3 运行的套接字

Figure 12 显示的是 DMA 数据传输的详细内容。本示例使用了三个长度为 L 的 DMA 缓冲区，这些缓冲区通过链接方式形成一个圆形循环。CX3 存储器地址位于左边。蓝色箭头表示该套接字从存储器加载套接字链接列表的描述符。红色箭头表示数据传输路径。下面各步骤描述了数据被发送到内部 DMA 缓冲区时所进行的套接字操作序列。

步骤 1：将存储器中的 DMA 描述符 1 加载到套接字内。获取有关 DMA 缓冲区的位置 (A1)、DMA 缓冲区的大小 (L) 以及下一个描述符 (DMA 描述符 2) 的信息。转到步骤 2。

步骤 2：将数据传输到起始地址为 A1 的 DMA 缓冲区。传输完 DMA 缓冲区大小 (L) 的数据数量后，转到步骤 3。

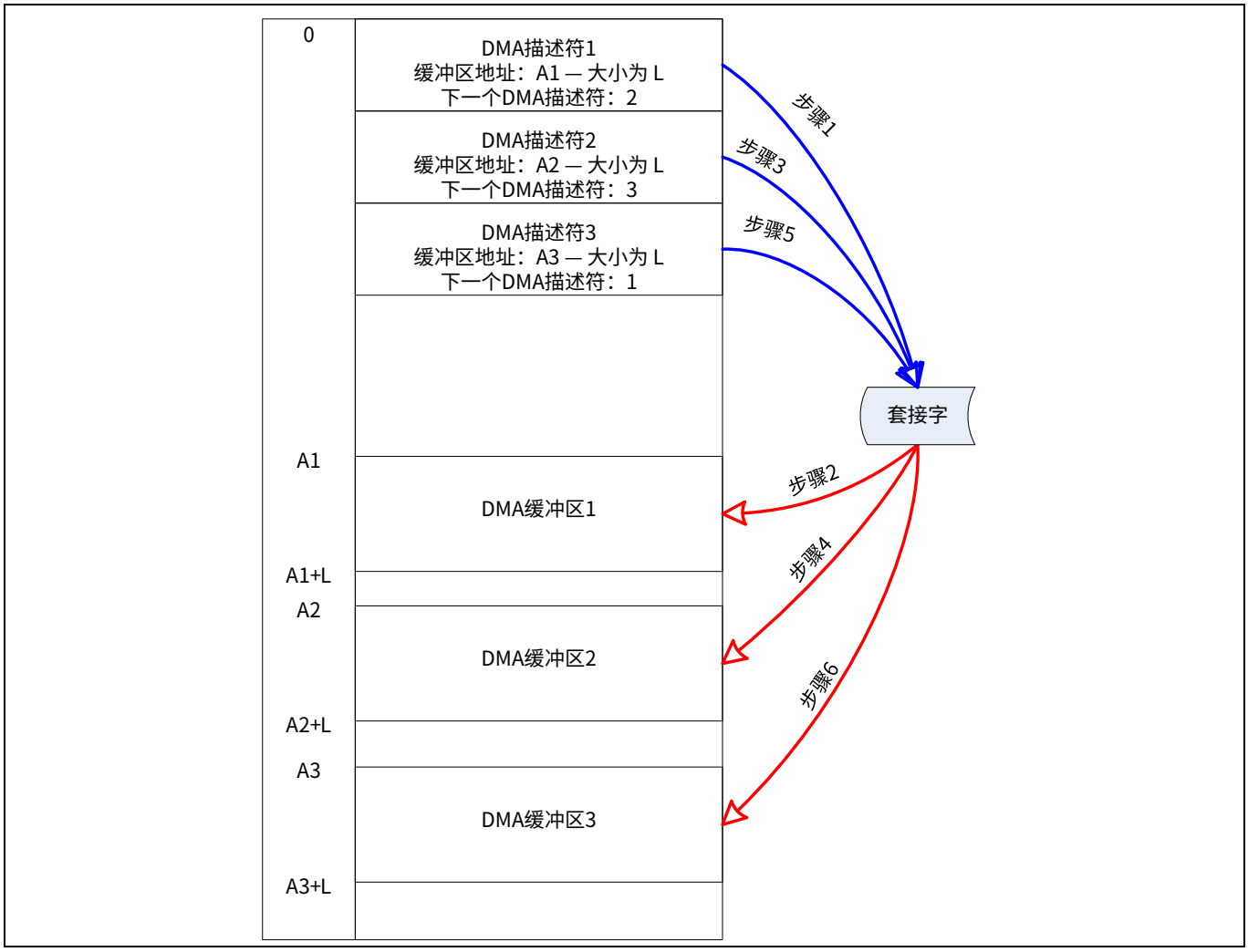


Figure 12　　DMA 传输示例

步骤 3：加载当前 DMA 描述符 1 所指向的 DMA 描述符 2。获取有关 DMA 缓冲区的位置 (A2)、DMA 缓冲区的大小 (L) 以及下一个描述符 (DMA 描述符 3) 的信息。转到步骤 4。

步骤 4：将数据传输到起始地址为 A2 的 DMA 缓冲区。传输完 DMA 缓冲区大小 (L) 的数据数量后，转到步骤 5。

步骤 5：加载当前 DMA 描述符 2 所指向的 DMA 描述符 3。获取有关 DMA 缓冲区的位置 (A3)、DMA 缓冲区的大小 (L) 以及下一个描述符 (DMA 描述符 1) 的信息。转到步骤 6。

步骤 6：将数据传输到起始地址为 A3 的 DMA 缓冲区。传输完 DMA 缓冲区大小 (*L*) 的数据数量后，转到步骤 1。

该简单方案在摄像机应用中存在问题。套接字检索存储器中下一个 DMA 描述符所需要的时间通常为 1 µs。如果在数据传输中间暂停了该传输，将丢失视频数据。为了避免发生这种情况，可以将 DMA 缓冲区大小设置为的视频线长度的倍数。这样，DMA 缓冲区切换造成的暂停会与视频线无效事件 (即傳感器的水平行 Blanking 間隔) 同步发生。然而，在某些情况 (如视频分辨率发生改变时)，该方法缺少灵活性。

设置 DMA 缓冲区大小等于线大小也不是一个好方案，因为这样 BULK 传输将不会利用 USB 3.2 Gen 1 的最大突发速率。USB 3.2 Gen 1 允许各 BULK 端点最大可达 16 个突发 x 1024 个字节。这就是将 DMA 缓冲区大小设置为 16 KB 的原因。

设置 DMA 系统

利用套接字在一个时钟周期内进行切换而不需要延迟的特性，该方案更具有优势。因此，建议使用*两*个套接字将数据存储在四个交错的 DMA 缓冲区内。

Note: *每个套接字至少需要两个缓冲区用于防止丢失数据。例如，当主机读取某个缓冲区内的数据时，可将套接字写入另一个缓冲区内。*

Note: ***Figure 5** 中的每个套接字的缓冲区数量与后面其他图形中所显示的不同，具体情况为：每个套接字四个缓冲区；每个套接字两个缓冲区。这样可以简化说明。不过，实际上，固件会为每个套接字设置四个缓冲区。*

Figure 13 显示的是使用了双套接字的数据传输以及执行的各步骤。套接字 0 和套接字 1 对 DMA 缓冲区的访问分别使用了红色和绿色箭头 (套接字有各自的独立数据路径) 区分。每一步骤中的“a”和“b”操作同时发生。通过硬件的此并行操作可以消除 DMA 描述符的返回的死区时间并允许 GPIF II 连续将数据流动到内部存储器内。这些步骤同 **Figure 12** 中的“Step”线相对应。

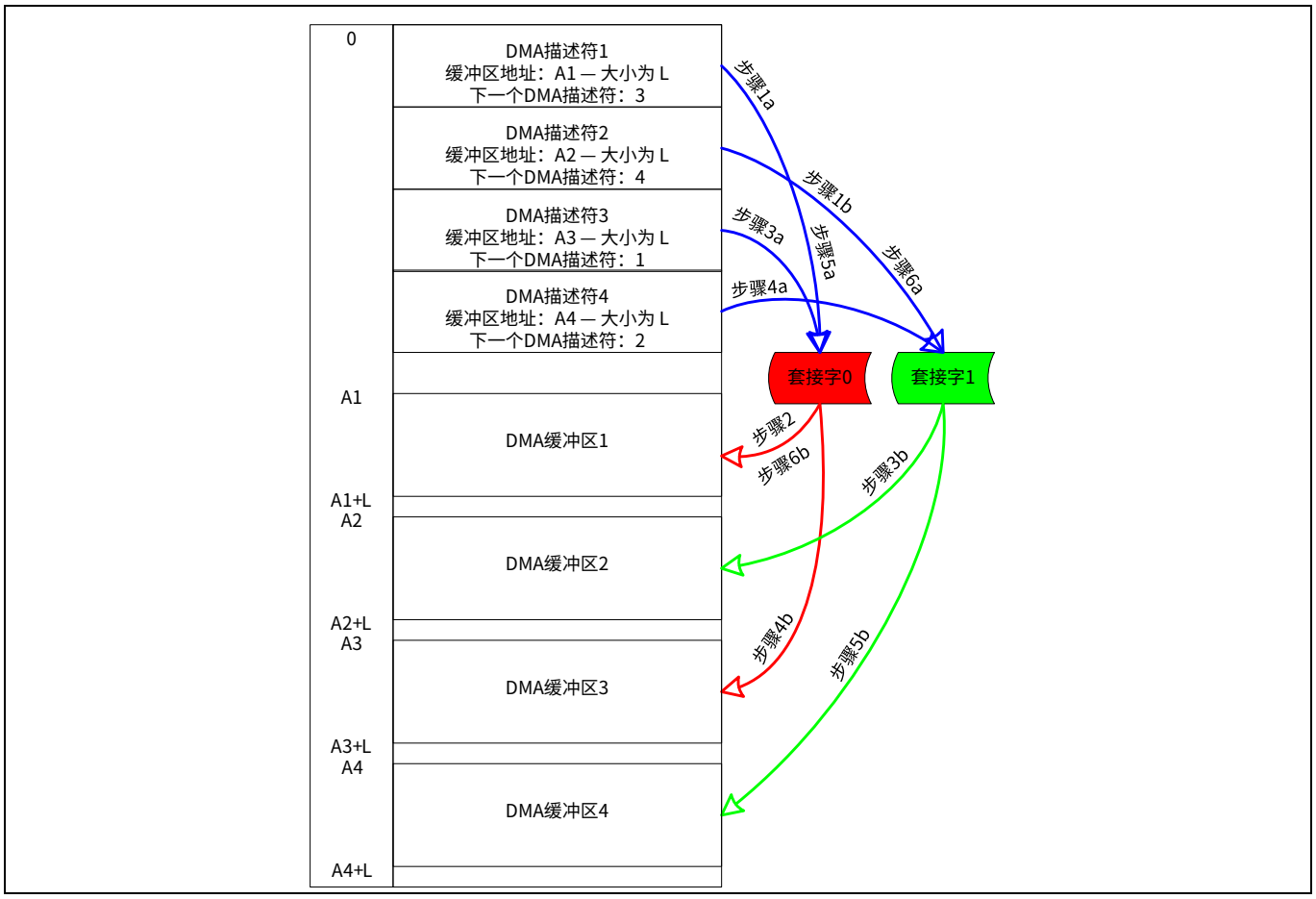


Figure 13　　使用双套接字的无缝传输

步骤 1：启动各套接字时，套接字 0 和套接字 1 分别加载 DMA 描述符 1 和 DMA 描述符 2。

步骤 2：当数据可用时，套接字 0 将数据传输到 DMA 缓冲区 1 内，该传输的长度为 *L*。传输结束后，请转到步骤 3。

步骤 3：GPIF II 切换 GPIF 线程，从而切换进行数据传输的套接字。套接字 1 开始将数据传输到 DMA 缓冲区 2 内，同时套接字 0 将加载 DMA 描述符 3。在套接字 1 完成传输 L 个字节数据时，套接字 0 已准备好将数据传输到 DMA 缓冲区 3 内。

设置 DMA 系统

步骤 4：GPIF II 此时切换回到原始的 GPIF 线程。这时，套接字 0 将长度为 L 的数据传输到 DMA 缓冲区 3 内。同时，套接字 1 将加载 DMA 描述符 4，以便准备好将数据传输到 DMA 缓冲区 4 内。套接字 0 完成传输 L 个字节的数据后，会转到步骤 5。

步骤 5：GPIF II 将套接字 1 的数据传输到 DMA 缓冲区 4 内。同时，套接字 0 加载 DMA 描述符 1 以准备好将数据传输到 DMA 缓冲区 1 内。请注意，步骤 5a 与步骤 1a 相同，但它的套接字 1 不进行初始化，而是同时进行数据传输的。

步骤 6：GPIF II 再次切换套接字后，套接字 0 会将 L 个字节数据传输到 DMA 缓冲区 1 内。假设 UIB 接收套接字此时已经读空了 DMA 缓冲区中的所有数据。套接字 1 同时会加载 DMA 描述符 2 并准备将数据传输到 DMA 缓冲区 2 内。在此，继续转到步骤 3。

只有消费套接字 (USB) 为空并及时释放了 DMA 缓冲区 (这样可以接受 GPIF II 中的下个视频数据块) 时，GPIF II 套接字才能传输视频数据。如果接收端的速度不够块，对 DMA 缓冲区的写操作将被忽略，使套接字丢失数据。这样，字节计数器将不再与实际传输同步，而且此现象将传播到下一帧。因此，如果经过很长时间仍未传输该帧，则需要一个清理机制。该机制在**第 5.1** 节中进行了介绍。

在以下四种情况下，帧传输会被终止：

- 套接字 0 已经传输已满的 DMA 缓冲区
- 套接字 1 已经传输已满的 DMA 缓冲区
- 套接字 0 已经传输未满的 DMA 缓冲区
- 套接字 1 已经传输未满的 DMA 缓冲区

在最后 2 种情况下，CPU 需要将该未满的 DMA 缓冲区传送到 USB 消耗端。

本应用笔记使用了 *SDK_INSTALL_PATH*\firmware\cx3_examples\cycx3_uvc_ov5640 文件夹中的项目，其中 *SDK_INSTALL_PATH* 是 **FX3 SDK** 安装的位置。32 位系统的默认安装路径是 *C:\Program Files\Cypress\EZ-USB FX3 SDK\1.3*。64 位系统的默认安装路径是 *C:\Program Files (x86)\Cypress\EZ-USB FX3 SDK\1.3*。

通过使用 *cycx3_uvc.c* 文件中名称为“CyCx3AppInit”的函数可以启动 DMA 通道。在 CyFxUVCApplnInit 函数中的“dmaMultiConfig”结构中对 DMA 通道的详细配置内容进行了自定义。它的类型被设置为 MANUAL_MANY_TO_ONE。

另外，将数据流动到 USB 3.2 Gen 1 主机的 USB 端点被配置为使能 16 突发 x 1024 字节的 BULK 端点。通过使用传送到“CyU3PSetEpConfig”函数中的“endPointConfig”结构 (端点常量设置为“CX3_EP_BULK_VIDEO”) 将可实现该操作。

3.1 DMA 缓冲区

本节总结了如何创建 CX3 DMA 缓冲区，并将其用于该应用。DMA 通道的组成部分在**第 2.2.1 节**中进行了介绍。

在这应用中，GPIF II 单元作为产生端，而 CX3 USB 单元作为消耗端。这应用使用了 GPIF II 模块中的 GPIF 线程切换性能以避免丢失数据。

当生产套接字加载一个 DMA 描述符时，它将检查相关的 DMA 缓冲区以确定该缓冲区是否能够进行写操作。生产套接字将其状态修改为*活动*，如果 DMA 缓冲区为空，它会把数据写入到 CX3 RAM 内。生产套接字锁定 DMA 缓冲区，以用于进行写操作。

当生产套接字完成写入 DMA 缓冲区内后，它会释放锁定，使消费套接字能够访问 DMA 缓冲区。该操作称为“缓冲包装”或“包装”。然后，DMA 单元会将 DMA 缓冲区传送给 CX3 RAM。生产套接字已经提供了一个 DMA 缓冲区。缓冲区只会在生产端不在填充时被包装。

如果 DMA 缓冲区被完全填充满，并且在帧期间重复填充操作，则包装操作会自动进行。生产套接字释放 DMA 缓冲区上的锁定，将其传送到 CX3 RAM 内，切换到空 DMA 缓冲区，然后继续写入视频数据流。

设置 DMA 系统

当消耗套接字加载了一个 DMA 描述符时，它将检查相关的 DMA 缓冲区以确定该缓冲区是否能够进行读操作。如果 DMA 缓冲区已被传送，消耗套接字会将它的状态修改为*活动*以读取 CX3 RAM 中的数据。消耗套接字锁定 DMA 缓冲区，以执行读操作。

- 当消耗套接字已经读取了 DMA 缓冲区中的所有数据后，它会释放锁定以便生产套接字访问 DMA 缓冲区。消耗套接字已经消耗了 DMA 缓冲区。
- 如果生产套接字和消耗套接字使用了相同的 DMA 描述符，那么 DMA 缓冲区的满/空状态将通过 DMA 描述符和套接字间的事件在生产套接字和消耗套接字之间自动通信。
- 在本应用中，由于 CPU 需要添加一个 12 字节的 UVC 标头，因此生产套接字和消耗套接字需要加载不同 DMA 描述符组。生产套接字加载的 DMA 描述符指向的是 DMA 缓冲区，这些缓冲区位于离消耗套接字所负载的 DMA 描述符指向的相应 DMA 缓冲区 12 字节偏移的位置上。
- 由于生产套接字和消耗套接字使用了不同的 DMA 描述符，因此 CPU 必须管理生产套接字和消耗套接字之间 DMA 缓冲区状态的通信。这就是 DMA 通道实现被称为“手动 DMA”通道的原因。
- GPIF II 模块生产 DMA 缓冲区后，将通过中断通知 CPU。然后，CPU 添加标头信息并将 DMA 缓冲区传送到消费端 (USB 接口模块)。
- 在 GPIF II 侧，DMA 缓冲区中的视频数据被自动包装，并被传送到 CX3 RAM (帧中的最后 DMA 缓冲区除外)。
- 在帧结尾处，最后的 DMA 缓冲区很大机会没有被完全充满。这时，CPU 会手动包装 DMA 缓冲区，并将该缓冲区传送到 CX3 RAM。这种操作称为“强制包装”。

Note: *如果選擇的 DMA 緩衝區大小 (以 Byte 為單位) 不是 UVC Frame 大小 (以 Byte 為單位) 的倍數，則 Frame 的最後一個緩衝區將導致部分填充的 DMA 緩衝區。這個部分填充的 DMA 緩衝區可用於識別 Frame 結束條件，以便在傳輸圖像數據時, 添加相應的 UVC 標頭。*

Note: *生成的部分 DMA 緩衝區大小將會是 4 的倍數，因為 `CyU3PDmaSocketSetWrapUp` API 只能以 4 的倍數提交數據。*

4 USB 视频类型 (UVC)

本章节为您介绍的是 USB 视频类型，它是在通用串行总线规范顶层上创建的一个协议，用于将视频数据传输到 PC。

要符合 UVC 类型，则需要下面两个 CX3 代码模块：

- 枚举数据
- 操作码

4.1 枚举数据

安装 SDK 时所提供的 *cycx3_uvc_ov5640* 项目中具有包含了 UVC 枚举数据的 *cycx3_uvcdscr.c* 文件 (如**第 4.3.1 节**所述)。可从 **usb.org** 上获取用于定义 UVC 描述符的 USB 规范。本节介绍了描述符的概述。一个 UVC 器件具有四个逻辑元素，每个元素由一个描述符定义：

- 输入摄像机终端 (IT)
- 输出端 (OT)
- 处理单元 (PU)
- 扩展单元 (EU)

这些元素在描述符中互相连接，如 **Figure 14** 所示。通过将描述符中的各终端编号连接起来，可连接上述各元素。例如，输入 (摄像机) 终端描述符声明了其 ID 为 1；处理单元描述符的输入连接的 ID 为 1，所以逻辑上表示他连接至输入端。输出端描述符制定使用哪个 USB 端点 — 在本情况下是 BULK-IN 端点 3。

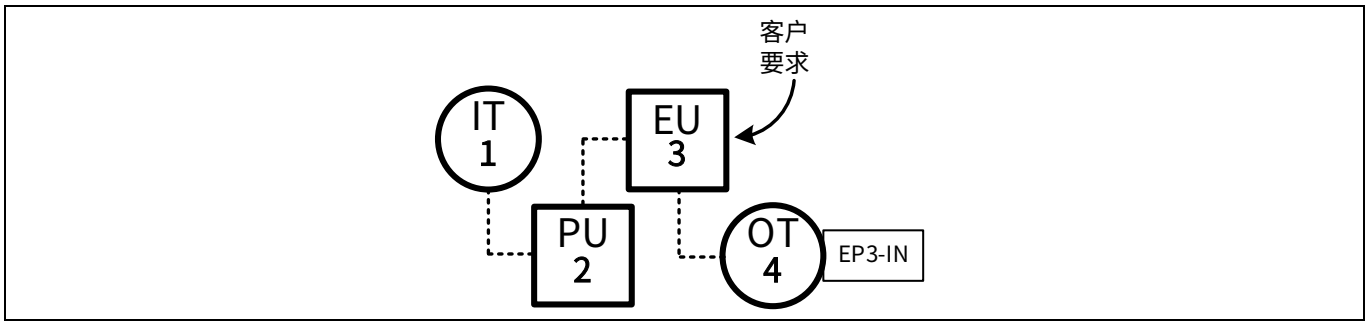


Figure 14 摄像机架构的 UVC 图

描述符还包括了视频属性 (如宽度、高度、帧频、帧大小和位深度) 以及控制属性 (如亮度、曝光、增益、对比度和 PTZ (平移、倾斜和缩放)) 等内容。

4.2 操作码

主机枚举摄像机后，UVC 驱动程序会向摄像机发出一系列请求，以确定工作特征。该操作叫做“能力请求阶段”。请求阶段先于串流阶段，其中主机应用会启动串流视频。

例如，假设 UVC 器件表明它支持某个 USB 描述符中的亮度控制。在能力请求阶段，UVC 驱动程序会查询器件以发现相关的亮度参数。

当主机发出一个请求要求更改亮度值时，UVC 驱动程序将发出‘SET’ (设置) 控制请求，用以更改亮度值 (SET_CUR)。该操作通过 USB 控制端点 (EP0) 完成。

同样，当主机应用需要选择输送一个受支持的视频格式、帧频或帧大小时，它会发出输送请求。共有两种类别的请求：PROBE 和 COMMIT。PROBE 请求用于确定 UVC 器件是否准备好切换到串流模式，COMMIT 则用于实现该切换。串流模式是图像格式、帧大小和帧频的组合。

4.3 UVC 的必要條件

本节介绍了示例项目如何满足 UVC 的要求。UVC 要求器件从事下面操作：

- 使用 UVC 特定的 USB 描述符进行枚举
- 处理 USB 描述符所记录的 UVC 控制和输送能力的 UVC 特定 SET/GET 请求
- 以符合 UVC 的颜色格式进行串流视频数据
- 添加一个符合 UVC 的标头到每个图像载荷内

可在 **UVC 规范**中了解这些要求的详情。

4.3.1 UVC 的 USB 描述符

cycx3_uvcdscr.c 文件含有 USB 描述符的表格。字节阵列“CyCx3USBHSConfigDscr” (高速) 和“CyCx3USBSSConfigDscr” (超速) 包含了 UVC 特定的描述符。这些描述符执行下面各子描述符树：

- 配置描述符
  - 接口关联描述符
  - 视频控制 (VC) 接口描述符
    - VC 接口标头描述符
      - 输入 (摄像机) 终端描述符
      - 处理单元描述符
      - 扩展单元描述符
      - 输出端描述符
    - VC 状态中断端点描述符
  - 视频流 (VS) 接口描述符
    - VS 接口输入标头描述符
      - VS 格式描述符
    - VS 帧描述符
  - BULK-IN 视频端点描述符

配置描述符是一个标准的 USB 描述符，它定义了 USB 设备在其子描述符中的功能。接口关联描述符用于向主机表明该设备是否属于标准 USB 类别。此处，该描述符使用了下面两个接口来上报符合 UVC 的设备：视频控制 (VC) 接口和视频流 (VS) 接口。因为 UVC 器件具有两个独立的接口，因此它是一个 USB 复合器件。

视频控制 (VC) 接口

VC 接口描述符及其子描述符将上报所有与控制接口有关的能力。示例包括亮度、对比度、色调、曝光和 PTZ 等控制。

VC 接口标头描述符是一个 UVC 特定的接口描述符，它表明了该 VC 接口属于哪一个 VS 接口。

输入 (摄像机) 终端描述符、处理单元描述符、扩展单元描述符以及输出端描述符均包含了各个位字段，这些位字段说明了相应终端或单元所支持的特性。

摄像机终端控制着机械 (或等效数字) 特性，比如传输视频流的设备的曝光和 PTZ。

USB 视频类型 (UVC)

处理单元控制着图像的各项属性，如正在流过它的视频的亮度、对比度以及色调。

与标准的 USB 供应商要求相似，扩展单元可以添加供应商指定的特性。在该设计中，扩展单元是空白的，但仍包含了该描述符，作为自定义功能的占位符。请注意，如果利用可改扩展单元，则标准主机应用将无法认出其特性，除非修改主机应用使其能识别这些特性。

输出端用于描述这些单元 (IT、PU、EU) 和主机之间的接口。VC 状态中断端点描述符是一个用于中断端点的标准 USB 描述符。该端点可用于传输 UVC 特定的状态信息。该端点的此功能属于本应用笔记范围外的内容。

UVC 规范已将这些功能分好了类，您可轻松安排实现该类别指定的控制请求。但实现上述功能是应用特定的。通过将相应的功能位设置为 ‘1’ ，可以在各自终端或单元描述符的位字段 “bmControls” (*cycx3_uvcdscr.c*) 中记录所支持的控制功能。在枚举时，UVC 器件的驱动程序会轮询控制的详细信息。通过 EP0 请求实现轮询的细节。所有相似的请求 (包括视频流请求) 均由 *cycx3_uvc.c* 文件中的 CyCx3AppUSBSetupCB 函数处理。

视频流 (VS) 接口

视频流接口描述符以及它的子描述符记录了各种帧格式 (如非压缩、MPEG、H.264 等)、帧的分辨率 (宽度、高度和位深度) 以及帧频。根据所记录的值，主机应用可以选择通过改变所支持的帧格式、帧分辨率和帧频的组合来切换流模式。

VS 接口的输入标头描述符指定了下面 VS 格式描述符的数量。

VS 格式描述符包含了图像的长宽比和颜色格式，如非压缩或压缩格式。

VS 帧描述符包含了图像的分辨率以及该分辨率的帧频。如果摄像机支持不同的分辨率，则多个 VS 帧描述符会遵循 VS 格式描述符。

BULK-IN 视频端点描述符是一个标准的 USB 端点描述符，它包含的有关批量端点可用于输送视频的信息。

本示例使用了两个帧描述符来支持两个分辨率和帧频集。其图像的特性包含在三个描述符中，如下面三个表格所示的内容 (只有相关的字节偏移量显示)。

Table 1　　VS 格式描述符的值

VS 格式描述符的字节偏移量	特性	超速值	高速值
23-24	长宽比	16:9	4:3

Table 2　　第一个 VS 帧描述符的值

VS 帧描述符字节偏移量	特性	超速值	高速值
5-8	分辨率 (宽、高)	0x780，0x438 (1920×1080)	0x140，0xF0 (320×240)
17-20	图像的最大尺寸，单位为字节	0x3F4800 (1920×1080×2)	0x25800 (320×240×2)
21-24 或 26-29	帧间隔，单位为 100 ns	0x51615 (30 fps)	0x1B207 (90 fps)

Table 3　　第二个 VS 帧描述符的值

VS 帧描述符字节偏移量	特性	超速值	高速值
5-8	分辨率 (宽、高)	0x500，0x2D0 (1280×720)	0x280，0x1E0 (640×480)
17-20	图像的最大尺寸，单位为字节	0x1C2000 (1280×720×2)	0x96000 (640×480×2)
21-24 或 26-29	帧间隔，单位为 100 ns	0x28B0A (60 fps)	0x28B0A (60 fps)

请注意，多字节数值将被存储并优先发送最低有效位 (LSB) (例如，低位优先)。因此，如果第一个超速帧频为 30 fps，那么帧间隔为：

$$FrameInterval = \frac{1}{30\ fps \times 100ns} = 333333 = 0x51615$$

它被存储为描述符 0x15、0x16、0x05 和 0x00。可以修改该设计，通过更改上述三个表格中的输入项来支持不同的图像分辨率。

4.3.2　UVC 特定的请求

UVC 规范使用了 USB 控制端点 EP0，使之能够向 UVC 器件进行控制和传输所请求的通信。这些请求用于发现并更改与视频相关的控制属性。UVC 规范将这些与视屏相关的控制内容定义为性能。通过这些性能您可以更改图像属性或输送视频。

性能可以是视频控制属性 (如亮度、对比度和色调)，也可以是视频流模式的属性 (如颜色格式、帧大小和帧频)。通过 USB 配置描述符的 USB 特定部分，可以记录其性能。每个性能都带有其属性。各项性能的属性如下所述：

- 最小值
- 最大值
- 最小值和最大值之间值的数量
- 默认值
- 当前值

SET 和 GET 是两种 UVC 特定的请求类型。SET 用于更改属性的当前值，而 GET 则用于读取属性。

下面是 UVC 特定请求的列表：

- SET_CUR 是 SET 请求的唯一类型。
- GET_CUR 用于读取当前值。
- GET_MIN 用于读取支持的最小值。
- GET_MAX 用于读取支持的最大值。
- GET_RES 用于读取分辨率 (步长值表示最小值和最大值之间受支持的数值)。
- GET_DEF 用于读取默认值。
- GET_LEN 用于读取属性大小，单位为字节。
- GET_INFO 用于查询特定性能的状态或支持情况。

USB 视频类型 (UVC)

针对已给的属性，UVC 规范将上述请求定义为强制或可选的。例如，如果 SET_CUR 请求对于一种特殊性能是可选的，则通过 GET_INFO 请求可确定它是否存在。如果摄像机不支持性能的某个请求，在主机向摄像机发出请求时必须通过搁置控制端点来指出这一点。

这些请求中的字节字段符合其目标性能。这些字节字段分层次，它具有与**第 4.3.1 节**中所述的 UVC 特定描述符相同的结构。第一层用以识别接口 (视频控制接口还是视频流接口)。

如果第一层确认该接口是视频控制，则第二层将确认终端或者单元，并且第三层确认该终端或单元的性能。例如，如果目标性能为亮度控制，则：

- 第一层用于视频控制
- 第二层用于处理单元
- 第三层用于亮度控制

如果第一层确认该接口为视频流，则第二层将为 PROBE (探针) 或 COMMIT (提交)。这样便不存在第三层。

当主机需要 UVC 器件启动输送或更改流模式时，主机会先确定器件是否支持新的流模式。要确定该项，主机会发出一系列的 SET 和 GET 请求，并将第二层设为 PROBE。器件可接受或拒绝对流模式进行的更改。如果器件接受更改请求，则主机将通过发出 SET_CUR 请求并把第二层设为 COMMIT 进行确认。

下面三个流程图显示的是主机与 UVC 器件间的互动情况。

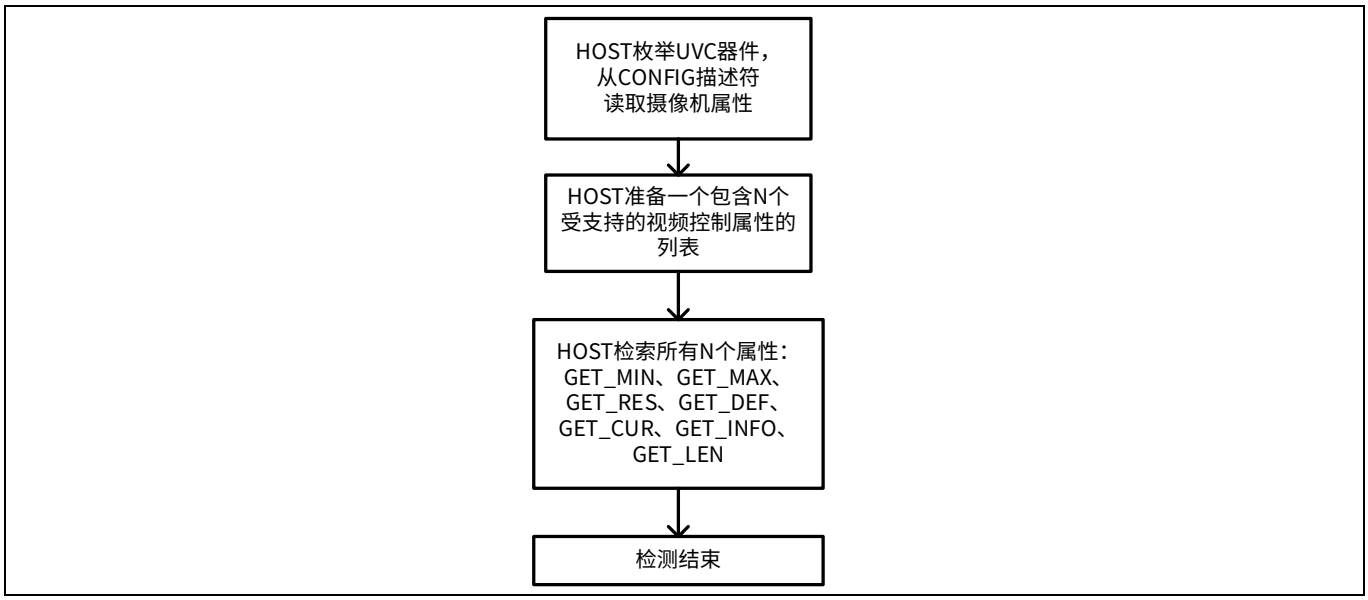


Figure 15　　UVC 枚举以及探索流程

当 UVC 插入到 USB 时，主机将枚举它，并发现摄像机支持的属性的细节 (**Figure 15**)。

在视频运行期间，摄像机的操作者可能会在主机应用所提供的显示对话框中更改摄像机的属性 (如亮度)。**Figure 16** 显示了此种交互情况。

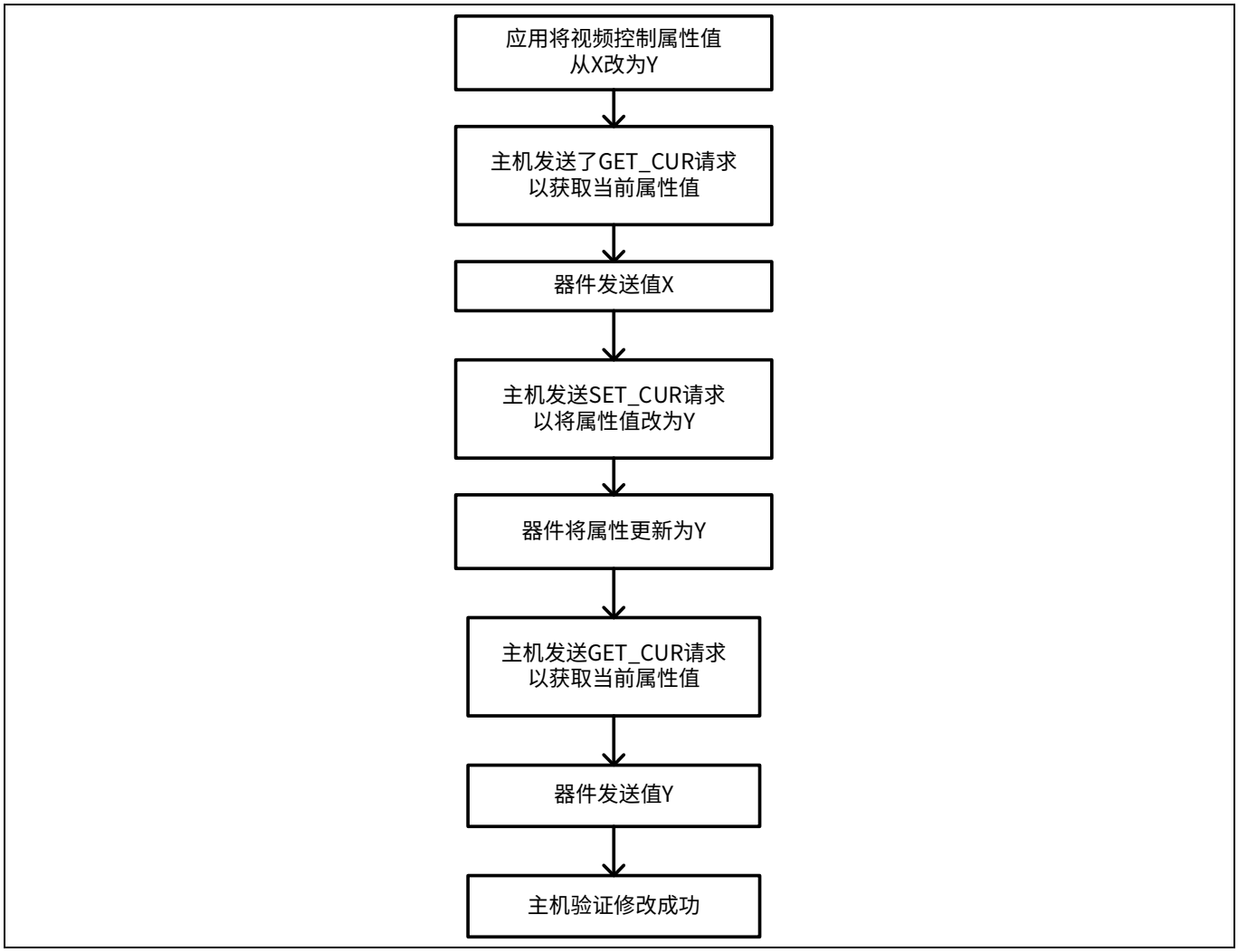


Figure 16　　主机应用更改了摄像机的设置

进行输送前，主机应用将发出一些探针请求，用以发现可能的流模式。确定默认流模式后，主机会发送一个 COMMIT 请求，该请求包含帧和带有所需分辨率和帧频的帧 ID。**Figure 17** 显示了该过程。主机将结构与 COMMIT 请求一同发送，该结构显示在 **Table 4** 中。

在该端点上，UVC 驱动程序可以从 UVC 器件输送视频。

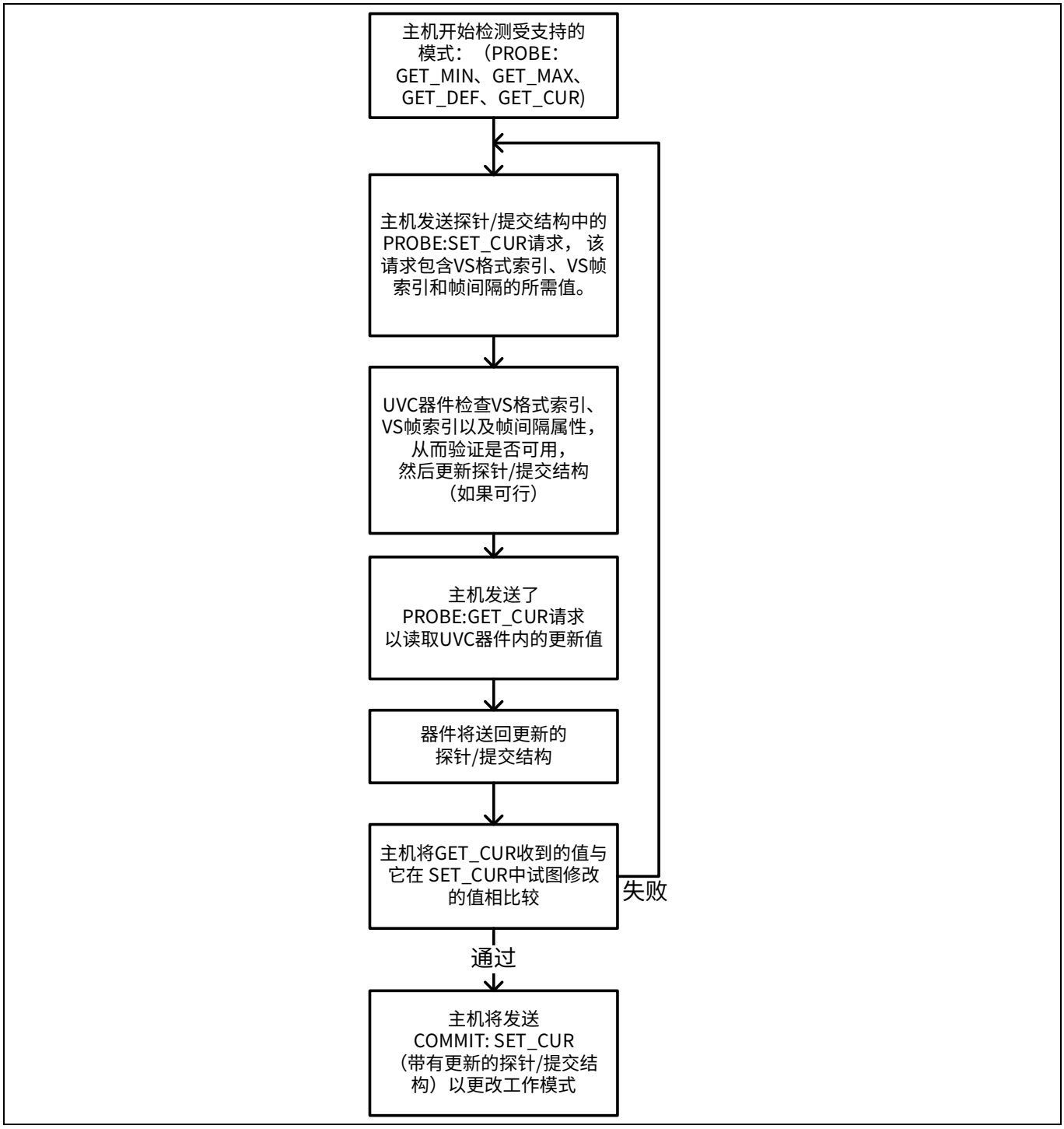


Figure 17　　主机与摄像机之间进行预输送时的对话框

Table 4　　启动 1080p @ 30fps 流前由主机发送的 Commit (提交) 结构值

探针/提交结构的字节偏移量	特性	数值
2	格式索引	1
3	帧索引	1
4-7	帧间隔，单位为 100 ns	0x51615 (30 fps)
18-21	图像的最大尺寸，单位为字节	0x3F4800 (1920×1080×2)

控制请求：亮度、PTZ、色调、饱和度以及其它因素

图像传感器支持对图像特性进行控制，如亮度、色调、饱和度等，并且可以通过添加外部硬件来支持 PTZ (平滑、倾斜和缩放)。

要想使 CX3 固件支持该特性，则必须修改视频控制描述符，使其支持特定的控制。例如，为支持亮度控制，必须设置处理单元描述符中 bmControls 字段的位 0 的值。更多详细信息，请参考**第 4.1 节**中介绍的 UVC 规范。

现在主机可以将请求直接发送到特定的终端。请查看**第 4.3.2 节**中的内容，了解主机可以发送的其它请求类型。通过 CyCx3AppUSBSetupCB 函数，可以处理这些请求。

接收到请求时，固件可以向图像传感器或与其关联的硬件发送请求。该特性是应用特定的。

串流请求：探针与提交控制

CyCx3AppUSBSetupCB 函数将处理与串流相关的请求。当 UVC 驱动程序需要从 UVC 器件输送视频时，首先要进行协商。在该阶段，驱动程序将发出一个 PROBE (探针) 请求，比如：GET_MIN、GET_MAX、GET_RES 和 GET_DEF。作为响应，CX3 固件会返回一个 PROBE (探针) 结构。该结构包含了 USB 描述符的索引，包括视频格式和视频帧、帧频、帧的最大尺寸以及负载大小等 (即 UVC 驱动程序在每次转换中可获取的字节数量)。

在示例中，GET_CUR 请求被处理，然后将检查当前的有效连接，并发送相应的探针结构。

处理完成 GET_CUR 后，将处理 SET_CUR 请求。在开始进行串流阶段时会发送这些请求。

COMMIT 控制的 SET_CUR 请求表示请求完成后主机将开始传输视频。根据主机所发送的帧描述符索引 (请参考 **Table 4**，加深了解由主机所发送的数据结构)，配置图像传感器以发送带有请求的分辨率和帧频的视频数据，然后设置相应的 MIPI CSI-2 接口参数。这时视频流才会开始。

视频数据格式：YUY2

UVC 规范只支持一个视频数据的颜色格式子集。因此，您需要选择一个能输送符合 UVC 规范要求颜色格式的映像传感器。本应用笔记包含了一个被称为 YUY2 的非压缩颜色格式。大部分 (而不是所有) 图像传感器均支持这种格式。YUY2 颜色格式是 YUV 颜色格式中 4:2:2 的采样版本。亮度值 Y 是对所有像素进行采样的，而色度值 U 和 V 是仅针对偶像素进行采样的。这样会创建“宏像素”，每个宏像素描述共使用了 4 个字节大小的两个图像像素。请注意，所有其他字节都是 Y 值，而 U 和 V 值仅用于表示偶像素。

Y0，U0，Y1，V0　　(头两个像素)

Y2，U2，Y3，V2　　(接下来两个像素)

Y4，U4，Y5，V4　　(接下来两个像素)

请参考**维基百科**，了解有关颜色格式的详细信息。

Note: *UVC 规范不支持 RGB 格式。但是，对于 Windows，微软提供了扩展型的 UVC 规范，用于支持其他多种格式，包括 RGB888 和 RGB565。在视频串流格式描述符中添加了相应的 GUID 来指定受支持的格式。更多信息，请参考**有关媒体类型的微软网页**。*

Note: *尽管 UVC 规范不支持单色图像，但通过将 8 位 RAW 数据作为 Y 值，并将所有 U 和 V 值设置为 0x80，仍能够将一个 8 位的单色图像显示为 YUY2 格式。此操作由图像传感器/ISP 处理。*

USB 视频类型 (UVC)

UVC 视频数据标头

UVC 类别非压缩视频负载需要一个 12 字节的标头数据。标头数据描述了所传输图像数据的属性。例如，它包括一个“新帧”位，图像传感器控制器 (CX3) 可以为每个帧切换该位。CX3 代码还可以设置标头数据中的错误位，以表示在传输当前帧的过程中发生的错误。每个 USB 传输操作都需要 UVC 数据标头。请参考 UVC 规范，了解详细信息。

Table 5 显示的是 UVC 视频标头数据的格式。

Table 5 UVC 视频标头数据格式

字节偏移	字段名称	说明
0	HLF	标头长度字段指定了标头的长度，单位为字节
1	BFH	位字段标头表示图像数据的类型、视频流的状态以及其他字段的存在情况
2-5	PTS	呈现时间戳表示源时钟时间，单位为本地器件时钟周期
6-11	SCR	参考源时钟表示系统时间时钟和 USB 帧开始 (SOF) 标志计数器

HLF 的值始终为 12。PTS 和 SCR 字段都是可选的。固件示例以 0 填充这些字段。位字段标头 (BFH) 将保存变化值，直到帧结束为止。**Table 6** 显示的是 BFH 格式 (BFH 是视频标头数据的一部分)。

Table 6 位字段标头 (BFH) 格式

位偏移	字段名称	说明
0	FID	帧标识符位在每个图像帧的起始边界上进行切换，在图像帧的其余部分该位保持不变
1	EOF	帧结束位指示视频结束，仅在属于图像帧的最后一个 USB 传输操作中设置该位
2	PTS	呈现时间戳位指示了 UVC 视频数据标头中 PTS 字段是否存在 (1 表示存在)
3	SCR	源时钟参考位指示了 UVC 视频数据标头中 SCR 字段是否存在 (1 表示存在)
4	RES	保留，将其设置为 0
5	STI	静态图像位指示视频取样是否属于静态图像
6	ERR	错误位表示在器件的传输过程中是否发生错误
7	EOH	标头结束位，如果被设置，表示 BFH 字段的结束

Figure 18 显示的是如何将这些标头数据添加到本应用的视频数据中。应针对所有 USB 批量传输操作添加 12 字节的标头。在这里，每个 USB 传输操作共有 16 个批量数据包。USB 3.2 Gen 1 的批量数据包的大小为 1024 个字节。

USB 视频类型 (UVC)

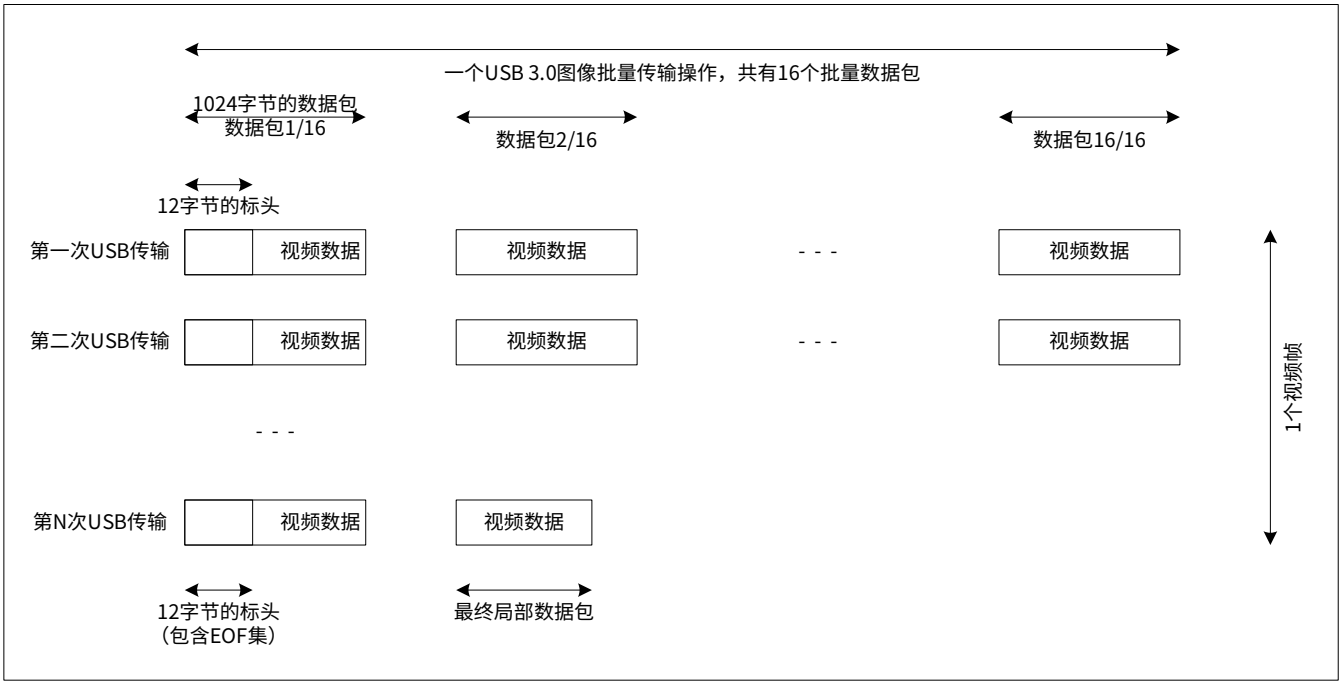


Figure 18 UVC 视频数据传输

5 CX3 固件

本应用笔记使用了安装 FX3/CX3 SDK 时所提供的 *cycx3_uvc_ov5640* 示例项目。首先，固件将初始化 CX3 CPU 并配置它的 I/O。然后，它将调用函数 (CyU3PKernelEntry)，以启动 ThreadX RTOS。RTOS 将自身初始化、创建几个内部线程，然后调用 CyFxApplicationDefine，使用户能够创建特定于应用的线程。

本示例创建了两个应用线程：uvcAppThread 和 uvcMipiErrorThread。RTOS 将调配资源执行这些应用线程，并处理执行应用线程函数的顺序 (分别执行 CyCx3UvcAppThread_Entry 和 CyCx3UvcMipiErrorThread)。**Figure 19** 显示的是基本的程序结构。

本应用笔记所描述的示例固件仅使用了 OV5640 图像传感器来执行基本的图像串流功能。如果您使用了其它传感器，或要求使用 OV5640 的其它功能，请务必修改相应的 *cycx3_uvc.c* 文件。**Table 7** 汇总了各种代码模块以及用于实现每个模块的各个函数。

Table 7 示例项目文件

文件	说明
cyfxtx.c	无需改动。将该文件使用到项目中。 它包含了各种变量和函数。RTOS 和 CX3 API 库将这些变量用于存储器映射，而 CX3 API 库将这些函数用于存储器管理。
cycx3_uvc.c	UVC 应用的主源文件。修改代码以支持其他控制功能 (如亮度、PTZ 等) 或添加各种视频数据流模式的支持时，需要修改该文件。 它包括下面各函数： • *main*：启动 CX3 器件、设置缓存、配置 CX3 I/O 并启动 RTOS 内核。 • *CyFxApplicationDefine*：定义 RTOS 执行的两个应用线程 • *CyCx3AppThread_Entry*：该函数由第一个应用线程执行。它将调用 CX3 内部模块的初始化函数、枚举器件，并处理器件暂停和帧完成任务。 • *CyCx3AppMipiErrorThread*：该函数由第二个应用线程执行。它将等待指示 MIPI CSI-2 控制器中发生错误的事件，然后使用 CyU3PMipicsiGetErrors 函数读取这些事件。 • *CyCx3AppDebugInit*：初始化用于打印调试信息的 CX3 UART 模块 • *CyCx3AppInit*：初始化 CX3 GPIO 模块、处理器模块或 PIB (GPIF II 是 PIB 的一部分)、I2C/CCI 接口、MIPI CSI-2 Rx 接口以及传感器 (将超速模式中的配置设置为 1080p 30fps)。它还会初始化用于枚举的 USB 模块、用于 USB 传输的端点配置存储器，并创建 DMA 通道配置 (用于将数据从两个 GPIF II 套接字传输到 USB 套接字)。 • *CyCx3AppGpifCB*：处理由 GPIF II 状态机生成的 CPU 中断 • *CyCx3AppDmaCallback*：跟踪从 CX3 到主机的输出视频数据。它将标头添加到图像数据中，并通知第一个应用线程来表示帧完成。 • *CyCx3AppUSBSetupCB*：处理所有由主机所发送的控制请求，设置表示已接收到主机所发送的 UVC 特定请求事件，并检测串流停止的时间它还会处理启动和停止视频串流时所需要的 UVC 类型特定的探针控制请求，并提交控制请求。 • *CyCx3AppUSBEventCB*：处理各个 USB 事件，如暂停、线缆断开连接、复位和恢复等事件。 • *CyCx3AppErrorHandler*：错误处理函数。这是一个占位符函数，它允许您执行错误处理 (若需要)。 • *CyCx3AppAddHeader*：在有效串流期间将 UVC 标头添加到视频数据中。

CX3 固件

文件	说明
cycx3_uvcdscr.c	包含 UVC 应用的 USB 枚举描述符。如果需要修改帧速率、图像分辨率、位深度或支持的视频控制，则需要修改该文件。UVC 规范包含了所有需要的详细信息。
cycx3_uvc.h	包含各个用于修改应用行为的开关，以便打开/关闭帧计数和 MIPI 错误线程的调试打印。 包含的各常量通常用于 *cycx3_uvc.c* 以及 *cycx3_uvcdscr.c* 文件。
cyfx_gcc_startup.s	该汇编源文件包含了 CX3 CPU 启动代码。它具有用于设置堆栈和中断向量的函数。 无需改动。

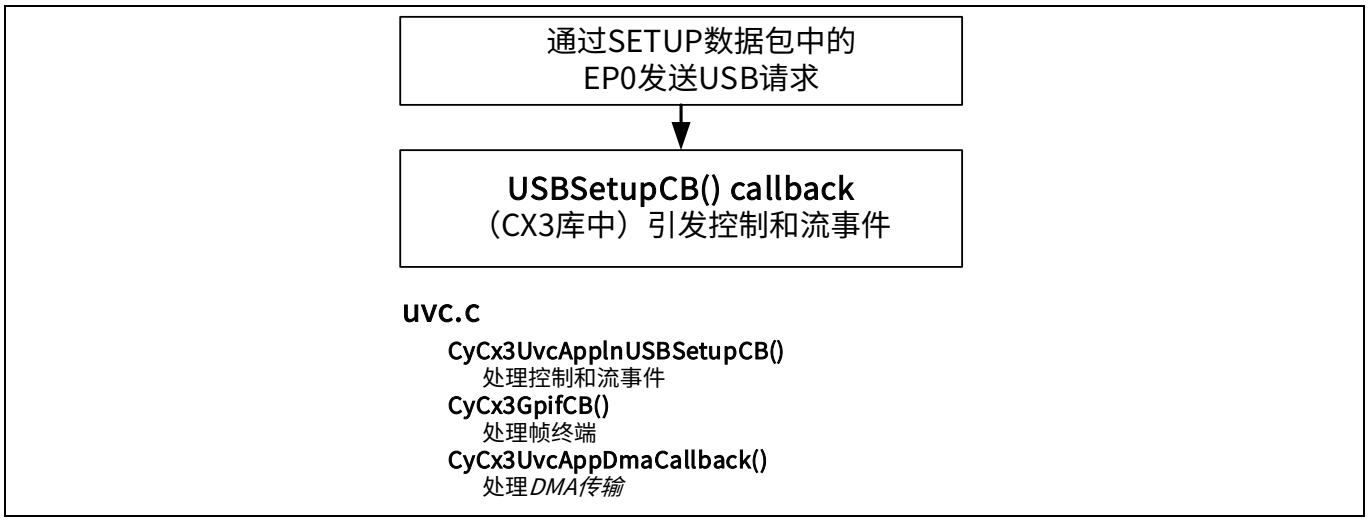


Figure 19　　高级摄像机项目结构

5.1　　应用线程

两个应用线程实现并行功能。

uvcAppThread (应用线程) 会初始化 CX3 外设并处理两个内部事件：*USB 暂停*和 *DMA 通道复位*。

uvcMipiErrorThread 监控着 MIPI CSI-2 错误，并会返回错误的计数值。

5.1.1　　USB 暂停事件

当主机将 CX3 设置为暂停状态时，会触发 USB 暂停事件 (CX3_USB_SUSP_EVENT_FLAG)，该事件执行了以下各项操作：

1. 禁用 MIPI CSI-2 控制器的时钟，使接口进入低功耗模式
2. 图像传感器处于睡眠状态
3. CX3 内核进入低功耗暂停模式，并且 USB 总线活动作为唤醒源

当检测到 USB 总线上发生的活动时，器件将被唤醒。唤醒器件后，图像传感器被上电，并将重新激活 MIPI CSI-2 控制器。

5.2 DMA 通道复位事件

如果帧传输执行的时间过长，过时数据将被删除，并将重新启动视频流。通过 CX3_DMA_RESET_EVENT 标志，可以实现上述操作。当 500 毫秒的定时器 (作为看门狗) 溢出时，将设置该标志。

每一个完整的帧被发送给主机时，该定时器将被复位；如果帧被卡在 DMA 缓冲区内较长时间，则定时器会溢出。如果将 DMA 缓冲区提交给主机失败 (使 DMA 通道失序)，也会重启视频流。

如果使能了 CX3_ERROR_THREAD_ENABLE 预处理宏，则会创建另一个线程 (uvcMipiErrorThread) 来监控 CSI-2 错误。

该线程的入口函数 (CyCx3UvcMipiErrorThread) 会定期调用 CyU3PMipicsiGetErrors，从 MIPI CSI-2 控制器中得到错误的计数值。如果观察到空白屏幕，则可以使用错误计数器进行调试 (有关如何处理 MIPI CSI-2 错误计数器的详细信息，请查看**第 8 节**)。

5.3 处理 UVC 请求

CX3 固件通过控制端点 (EP0) 处理 UVC 特定的控制请求 (**第 4.3.2 节**中所描述的 SET_CUR、GET_CUR、GET_MIN 以及 GET_MAX)。类别特定的控制请求由 CyCx3AppUSBSetupCB (CB=Callback) 函数处理。每当 CX3 接收到这些控制请求的其中一个时，该函数将对接收到的设置数据进行解码并处理它。

对定向于视频串流接口的探针控制请求 (请查看**第 4.3.2.2 节**)，它将相应的探针控制结构发送给主机。如果发出提交控制请求，将开始串流视频。

所有针对视频控制接口的请求 (Get Status 请求除外) (请查看**第 4.3.2.1 节**) 均被停止 (USB 主机收到 STALL 握手数据包)，这是因为本示例不支持视频控制特性。如果您的示例支持亮度、曝光度、PTZ 或其它控制性能，务必通过此处控制这些性能。

5.4 初始化

CyCx3UvcAppThread_Entry 函数将调用 CyCx3AppDebugInit 来初始化 UART 的调试能力，并调用 CyCx3AppInit 来初始化其余所需模块、DMA 通道以及 USB 端点。

5.5 枚举

在 CyCx3AppInit 函数中，调用 CyU3PUsbSetDesc 函数会使 CX3 枚举为 UVC 器件。在 *cycx3_uvcdscr.c* 文件中定义了 UVC 描述符。这些描述符的定义用于以非压缩的 YUY2 格式传输每像素为 16 位的图像传感器，其分辨率为 30 fps 时的 1920 x 1080 像素和 60 fps 时的 1280 x 720 像素。若需要修改这些设置，请参考**第 4.3.1 节**中的内容。

5.6 配置 MIPI CSI-2 控制器

CyCx3AppInit 函数首先初始化与 MIPI CSI-2 控制器连接的 I^2C、GPIO 以及 PIB 模块。

创建 DMA 通道后，通过调用 CyU3PMipicsiGpifLoad 函数 (其参数为数据总线宽度和缓冲区大小)，可以加载 GPIF II 状态机。对于生产套接字，作为参数传入的缓冲区大小就是 DMA 缓冲区大小。因此，被加载的值为：

dma_buffer_size – (dma_header_size + dma_footer_size)

所选的总线宽度要与通过 CSI-2 发送的图像数据类型相匹配。**SDK API 指南**中定义了每种受支持的图像数据类型需要的总线宽度。

这时状态机将被启动，然后被暂停，直到 USB 主机发送图像串流请求为止。

CX3 固件

加载并启动状态机后，可以初始化 MIPI CSI-2 控制器。该操作通过调用 CyU3PMipicsiInit 执行。通过置位 XRES 引脚，该函数将复位图像传感器。要想使传感器退出复位，请使用 CyU3PMipicsiSetSensorControl 函数，取消置位 XRES 引脚。

然后通过 CyU3PMipicsiSetIntfParams 函数配置 MIPI CSI-2 控制器。该函数使用了包含配置参数的 CyU3PMipicsiCfg_t 类结构。

英飞凌提供了一个 CX3 配置工具 (**Figure 20** 至 **Figure 23** 所示的屏幕截图)，以便生成 CX3 项目。使用该工具可以根据用户输入生成完整的 CX3 固件项目 (包含 UVC/供货商类别描述符、处理 UVC 请求、视频串流 DMA 引擎以及 MIPI CSI-2 控制器配置)。访问该工具并执行操作的流程详细介绍在 **Steps to set up MIPI CSI camera solution with CX3 – KBA225748** 中。请访问 *Help > Help Contents > Cypress EZ-USB ™Guides > EZ-USB ™Suite Guide > CX3 Configuration Utility* 路径下的赛普拉斯 EZ-USB™套件 (Eclipse)，查看 CX3 配置工具帮助。

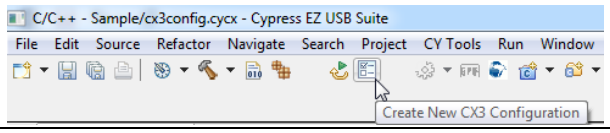


Figure 20　　CX3 配置工具快捷方式

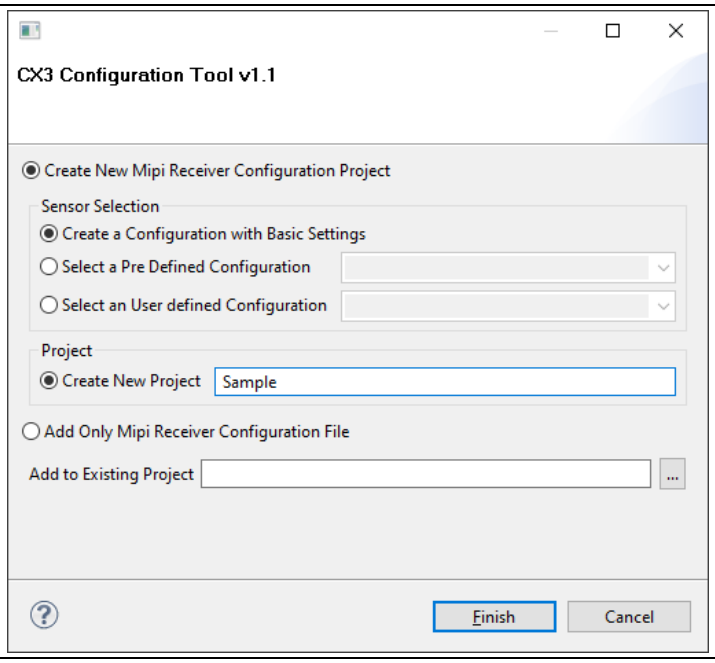


Figure 21　　CX3 配置工具起始页

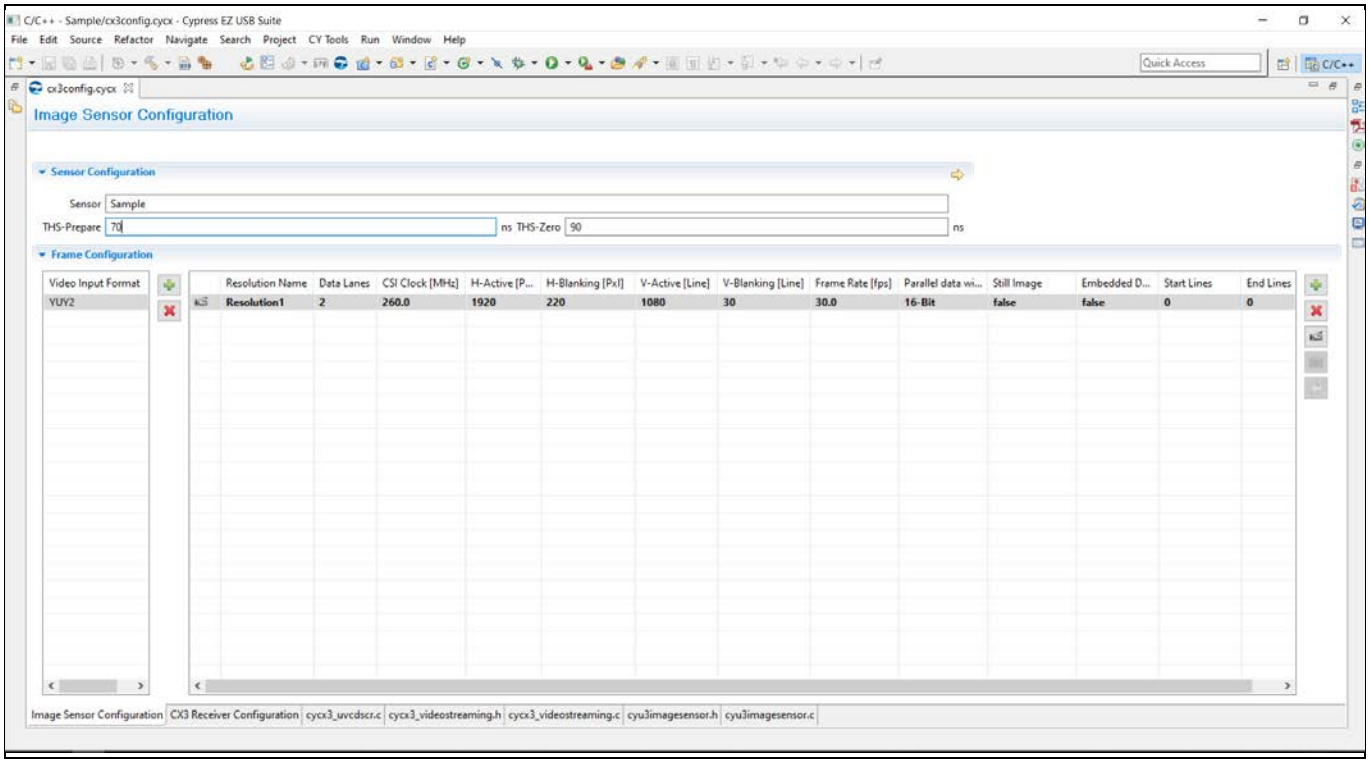


Figure 22 CX3 配置工具的传感器配置页

Note: 從图像传感器供應商取得图像传感器的配置參數

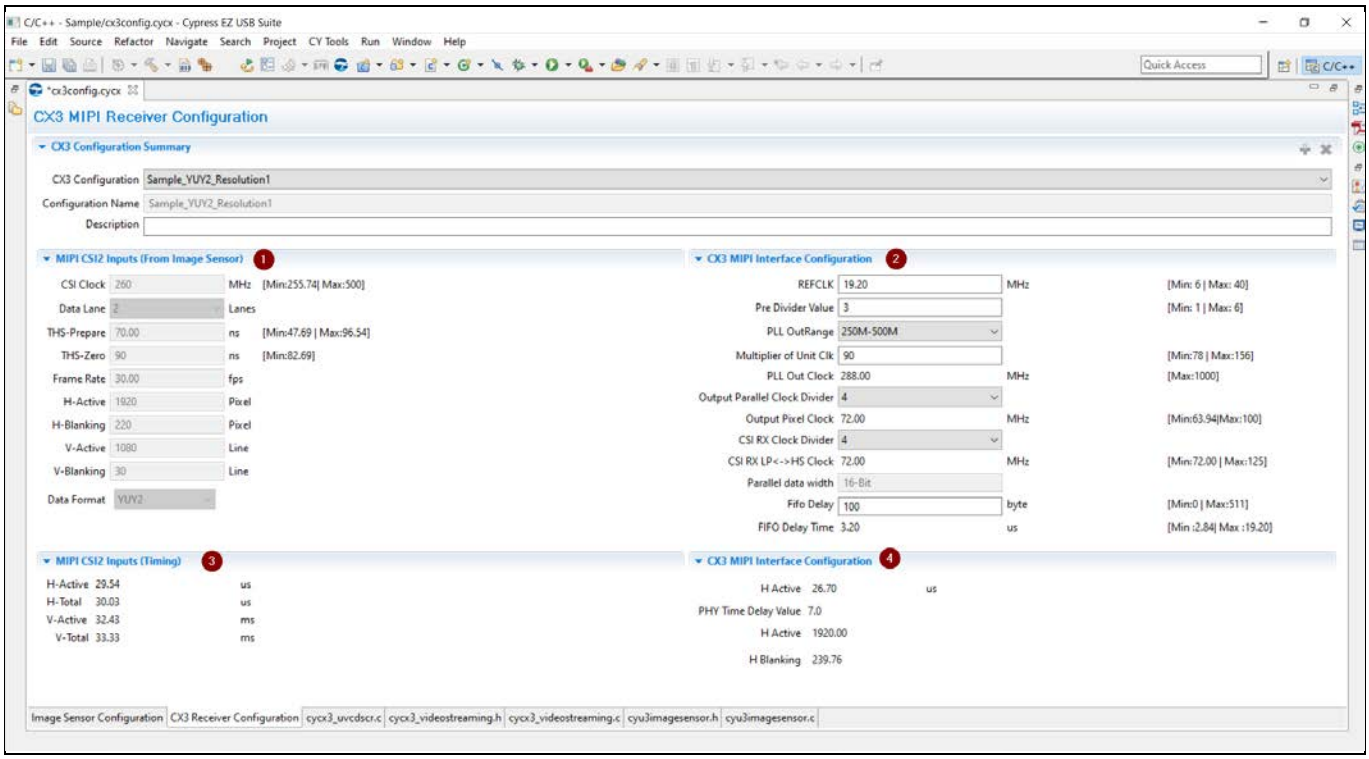


Figure 23 CX3 MIPI 接收器配置工具

5.6.1 CX3 MIPI 接收器配置工具

CX3 配置工具可用于生成一个完整的 CX3 项目或 MIPI 接收器配置 (该配置可添加到某个现存项目中)。本节简单介绍了 MIPI Receiver Configuration (MIPI 接收器配置) 选项卡中的各个字段 (**Figure 23**)。

MIPI CSI2 输入：请参考 **Figure 23** 中所标注的 ‘1’ 。这些字段都是 CSI-2 输入参数。例如，**Figure 23** 显示的是 CSI-2 传感器接口，另外 CSI 时钟频率为 260-MHz 传输 YUY2 Full HD (1080p)，各像素以 H-Blanking 为 220 像素，并且 V-Blanking 为 30 线。这些值要与 CSI-2 接口的实际参数相匹配，因为根据这些输入进行的计算用于验证 MIPI 控制器配置参数。

该节中的所有参数 (包括数据格式) 仅用于计算。本节所提供的数据格式指出了对正在传输的图像格式的一个像素进行编码时所需要的位数量。

Note: 解像度的寬度或行大小, 單位為 Byte, 必須是 4 的倍數。如果不滿足此條件，CX3 的 CSI-2 MIPI 控制器模塊將在行數據添加額外的填充字節。

目前，该工具不适用 THS-Prepare 和 THS-Zero 值字段。要想调整 THS-Prepare 和 Zero 值，SDK 应支持一个 API ‘CyU3PMipicsiSetPhyTimeDelay’ 。请参考 FX3 SDK **API 指南**，深入了解该 API。PHY 时间延迟为 9 (上述 API 的第二个参数)，与大部分传感器/ISP 配合使用。

MIPI CSI2 输入时序：请参考 **Figure 23** 中所标注的 ‘3’ 。这些时序都是根据 CSI-2 输入参数计算得出的。

CX3 MIPI 接口配置：请参考 **Figure 23** 中所标注的 ‘2’ 。本节所提供的参数用于配置 MIPI CSI-2 控制器中使用的三个时钟 (PLL 时钟，CSI RX LP <-> HS 时钟，并行输出像素时钟)。更多有关这些时钟的信息，请参考 **CX3 技术参考手册**中的第 1.7 节。

输出像素时钟 (PLCK)、数据格式和 Fifo 延迟都是关键参数。这些参数决定了 CX3 中的 MIPI 控制器是否能够正常操作。

输出像素时钟参数的范围由工具指定。

数据格式参数决定了并行输出接口的数据总线宽度。请参考 CX3 技术参考手册中的第 1.6 节，了解受支持的数据格式列表以及每种格式所支持的并行输出数据总线的宽度。

设置时，FIFO 延迟参数将在并行输出的每个线开头添加一个延迟。当输出并行像素时钟设置值大于该工具所建议的输出并行像素 PCLK ‘最小值’ 时，需要使用 Fifo 延迟参数。**Figure 23** 介绍了 FIFO 延迟的基本概念。必须指出的是，FIFO 延迟参数的最小值和最大值不是通过对 MIPI H-有效时间和输出并行 H-有效时间进行简单减法计算得出的。在 MIPI 控制器中，只有一个行缓冲区可用，在计算 FIFO 延迟时间时，需要计算该缓冲区。该工具负责进行此项调整。

Note: 发生下一个 MIPI 线数据之前，应该通过并行接口完成发送当前的 MIPI 线数据。如果违反该规则，可以在视频帧中观察到垂直分割。因此，要谨慎选择使用 FIFO 延迟。

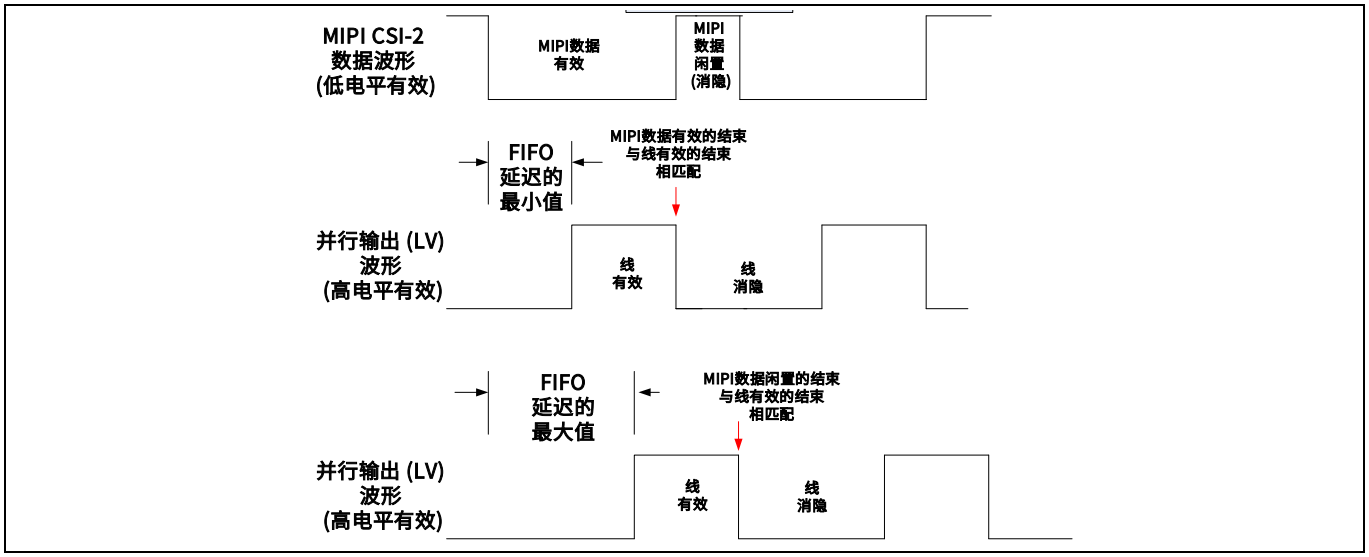


Figure 24 FIFO 延迟的基本概念

Note: *更多詳細資訊請參閱* ***Analysis of CX3 clocking parameters – KBA226758*** *和* ***analysis of CX3 video timing parameters – KBA226779***

要將其他傳感器, 設置為任何其他格式的串流傳輸，請在 **FX3 SDK** 中創建一個新項目，並在您看到傳感器配置頁面時選擇輸入視頻格式 (如圖 22 所示)。RGB 可以和 YUV 格式, 使用相同的串流傳輸方式，具體取決於像素中的字節數。要獲取更多詳細信息, 關於 RGB 格式的串流傳輸，請參閱 **FX3 SDK** 中的其他示例項目。要將 MJPEG 格式的串流, 以不同的解像率來傳輸, 並將 CX3 Device 與 ISP (圖像信號處理器) 連接，請參閱 THEIA-CAM CX3 Base 的代碼, 它來自 Thin Solutions。請注意，您需要購買 THine ISP 13MP 參考設計套件 (RDK) 以獲取基於 CX3 的 **THEIA-CAM** 的韌體代碼。

如果您需要帶有 OV5640 傳感器的 MJPEG 韌體範例，您必須與 OmniVision 公司簽署 NDA 並開立 **Technical Support** case.。

5.7 配置图像传感器

配置 MIPI CSI-2 接口后，需要相应配置图像传感器。在 CyCx3AppInit 函数快要结束时，应完成配置操作。英飞凌提供了一个预编译库，该库包含了用于初始化和配置 OV5640 图像传感器的各个函数。

对于其它的图像传感器，*cyu3imagesensor.c* 文件包含几个辅助函数用于配置图像传感器。

使用 CX3 的 I^2C 主模块来配置图像传感器。使用 *cyu3imagesensor.c* 文件中的 SensorWrite2B、SensorWrite、SensorRead2B 以及 SensorRead 函数，通过 I^2C 对图像传感器配置进行读写操作。

SensorWrite2B 和 SensorWrite 函数将调用 CyU3PI2cTransmitBytes 标准 API，向图像传感器内写入数据。SensorRead2B 和 SensorRead 函数将调用 CyU3PI2cReceiveBytes 标准 API，读取图像传感器内的数据。有关这些 API 的更多详细信息，请查阅 **FX3 SDK API 指南**中的内容。

5.8 启动视频流

USB 主机应用 (如 VLC 播放器、e-CAMView、经典媒体播放器或 VirtualDub) 位于 UVC 驱动器之上，以便将 USB 接口和 USB 备用设置组合到一个流视频 (通常为接口 0 备用设置 1)，并发送 PROBE/COMMIT 控制请求。该主机指示符表示快要启动一个视频数据流。发生流事件时，USB 主机应用会向 CX3 请求图像数据；CX3 开始将图像数据从图像传感器发送给 USB 3.2 GEN 1 主机。

CX3 固件

当 CX3 接收到流事件时，USB 事件回调 (CyCx3AppUSBSetupCB) 将根据所接受到的帧索引来配置图像传感器和 MIPI CSI-2 接口。配置参数取决于主机要求的分辨率和帧频；下一节 (**第 5.8 节**) 将详细说明该内容。

由于直到主机请求图像数据为止，一直不用运行 GPIF II 状态机，所以它会保持暂停状态。发生流事件时，将调用 CyCx3UvcStart 函数。这样将恢复状态机 (使用 CyU3PGpifSMControl)、唤醒 MIPI CSI-2 接口 (使用 CyU3PMipicsiWakeup) 并给图像传感器上电。

Note: ***SDK API 指南****包含了上下文内容所使用的且与 MIPI CSI-2 和 GPIF II 相关的函数的详细信息。*

5.9 选择并切换帧设置

摄像机支持多种分辨率和帧频。各自视频串流帧描述符中包含了每个受支持的分辨率/帧频的设置，如**第 4.3.1.2 节**中所述的内容。

选中 USB 主机应用中的特定分辨率或帧频设置时，USB 主机会将 SET_CUR 提交控制请求发送给视频流接口。接收结构中的第四个字节表示选中的帧描述符。

在所提供的示例项目中，如果器件在超速模式下运行，则索引 0 支持 720p@60 fps 视频流，并且索引 1 支持 1080p@30 fps 视频流。USB 主机要求使用串流 1080p 视频时，它将发送 SET_CUR 请求并将帧索引设置为‘1’。

CyCx3UvcAppUSBSetupCB 函数将读取该索引，并根据所收到的索引来配置图像传感器和 MIPI CSI-2 控制器，以按照要求的设置情况来串流视频。

5.10 设置 DMA 缓冲区

UVC 规范要求在每个 USB 传输 (在本应用中即为每个 16 KB DMA 传输) 增加一个大小为 12 字节的标头。但是，CX3 架构要求所有与 DMA 描述符相关联的 DMA 缓冲区的大小必须为 16 字节的倍数。

由于在 DMA 缓冲区中保留了供 CX3 CPU 填充的 12 字节，因此 DMA 缓冲区大小不再是 16 字节的倍数。所以，DMA 缓冲区大小应为 16,384 减去 16，而不是减去 12。DMA 大小 (不包括 CX3 固件所添加的 12 字节标头) 为 16,384-16=16,368 字节。DMA 缓冲区的结构如下：12 字节的标头，16,368 视频字节，以及在 DMA 缓冲区终端的 4 个未使用字节。这样，DMA 缓冲区的实现最大空间为 16 x 1024 (16,384) 字节的 USB BULK 突发。

5.11 在视频流期间中处理 DMA 缓冲区

CyCx3AppInit 函数为生成和消耗事件创建了一个具有回调通知的 DMA 手动通道。

消费通知用于跟踪主机读取的数据量。主机读取完整个帧后，将复位跟踪变量并重启 GPIF II 状态机。生成事件通知用于附加 12 字节的标头并将数据传输给主机。

在 DMA 回调中，固件使用 CyU3PDmaMultiChannelGetbuffer 函数来检查 DMA 缓冲区。当 GPIF II 生成的 DMA 缓冲区被调配，或被 CX3 CPU 强制打包时，则 CX3 CPU 可使用它。在有效的帧期间，图像传感器使数据发生转移，并且 GPIF II 将提供填满的 DMA 缓冲区。此时，CX3 CPU 需要将 16,380 字节数据提交给 USB。

通常，帧结束时，最后的 DMA 缓冲区未饱和。在这种情况下，固件必须在生产端上强制打包 DMA 缓冲区，再触发一个生产事件，然后将 DMA 缓冲区中合适的字节数调配给 USB。使用 CyU3PDmaMultiChannelSetWrapUp 调用的强制打包 DMA 缓冲区 (GPIF II 生成) 由 GPIF II 回调函数 CyCx3GpifCB 执行。GPIF II 设置 CPU 中断 (在帧结束时生成) 时，将触发该回调函数。

Note: *UVC 标头包含了有关帧标识符和帧结束标志的信息。在帧的结束部分，固件设置第二个 UVC 标头字节的位 1，并切换它的位 0 (请查看 CyCx3UvcAddHeader)。此外，通过设置“hitFV”变量可表示从图像传感器捕捉的帧已经结束。*

glDmaDone 变量将跟踪 DMA 缓冲区，从而能够确保已经从 CX3 FIFO 读取了所有数据。

5.12 在帧结束时恢复串流

某帧结束时，GPIF II 状态机将生成 CPU 中断，从而启动前面介绍的事件串链。當 Frame 的最後一個 DMA Buffer 在 USB Host 處理完畢後，glDmaDone 變數值, 設置為零。

CX3 固件使用 `CyU3PGpifControlSWInput` API 觸發一個中斷事件，然後重新啟動狀態機以處理下一個傳入的 Frame。

在舊版本的 CX3 固件中，`CyU3PGpifSMSwitch` 函數, 用來將 GPIF II 狀態機重新啟動到 START 狀態。如果套接字 0 读取了帧中最后的缓冲区，GPIF II 状态机以 ALPHA_CX3_START_SCK1 被重启，开始读取套接字 1 上的下一帧，反过来也一样。

此外，UVC 规范要求每一帧都要切换标头中的帧 ID 位。将最后的缓冲区提交给 USB 主机前，通过 CyCx3UvcAddHeader 函数可以完成此操作。

5.13 终止视频流

共有三种方法可终止图像流：

- 断开摄像机与主机间的连接
- 关闭 USB 主机程序
- USB 主机向 CX3 发送请求或暂停请求。

当 FIFO 中有剩余数据时，仍可终止视频流，因此需要清理 FIFO。

终止视频流时，固件将复位与串流相关的变量、复位 DMA 通道并暂停 GPIF II 状态机。它还会断开摄像机和 MIPI CSI-2 接口的电源。通过 CyCx3AppStop 函数，可以实现此操作。

当关闭 USB 主机应用时，它将发送 Windows 平台上的发送清理请求，或发送 Mac 平台上备用设置为 0 的设置界面请求。接收到该请求时，流会停止。当目标为 CY_U3P_USB_TARGET_ENDPT 并请求为 CY_U3P_USB_SC_CLEAR_FEATURE 时，CyCx3AppUSBSetupCB 函数将处理该请求。

终止视频流后，图像传感器被掉电，并通过 CyU3PSysEnterSuspendMode 函数暂停 CX3 内核。当检测到 USB 总线上发生的活动时，CX3 内核将被唤醒，并给图像传感器上电，重启视频流。

使用 CCI (I^2C) 指令或使用 CX3 的 XSHUTDOWN 引脚均可以使图像传感器掉电。该引脚由 CyU3PMipicsiSetSensorControl 函数 (其第一个参数为 CY_U3P_CSI_IO_XSHUTDOWN) 控制。

6 硬件设置

6.1 使用 CX3 RDK 进行测试

通过包含了 CX3 参考设计套件 (RDK) (该套件使用 OmniVision OV5640 图像传感器) 的设置对当前项目进行测试。请访问英飞凌网站 **www.cypress.com/cx3** 下的 **Kits** 选项卡，了解有关该套件的详细信息。

6.1.1 套件购买

1. 通过 **http://www.e-consystems.com/CX3-Reference-Design-Kit.asp** 网站下的 e-Con 系统购买 RDK
2. 英飞凌提供了附加的 CX3 SDK 二进制库，该库可以执行 OV5640 传感器的基本操作。如果需要支持額外的功能，請與 OmniVision 簽署 NDA 並聯繫 **Tech Support**。验证 NDA 后，英飞凌将提供特定于 OmniVision 的相關文件。
3. 使用 USB 3.2 GEN 1 主机使能的电脑进行评估 SuperSpeed 的性能。

CX3 RDK 包含了快速入门指南、硬件用户手册以及固件构建手册，这些材料有助于开始使用 RDK。

6.2 设计自己的电路板

设计自己的电路板时，需要遵循相应指南，以确保电路板正常工作。請參閱 **FX3/FX3S 硬件設計應用指南**，以了解適用於 CX3 的 FX3/FX3S 硬件設計的推薦做法。此外，請參閱 **CX3 硬件常見問題解答 KBA** 了解更多詳細信息。

对于 MIPI CSI-2 信号，需要遵循下述各布线指南：

- 电路板上的走线长度不能超过 100 mm。
- 差分对跟踪的传输线阻抗 (Z0) 为 100 Ω +/- 10%。
- 通道内 (P 和 N 线之间) 的长度偏差要小于 0.5 mm。
- 连接各组件的通道 (两个 MIPI CSI 通道信号对之间) 的长度偏差要小于 1.5 mm。
- P 和 N 信号走线的间隔要等于走线宽度的两倍。

7 基于 UVC 的主机应用

各种主机应用允许您使用 UVC 器件来显示并捕捉视频。在 Windows OS 中 **Media Player Classic** 是個常見的選擇。另外两个 Windows 应用分别为 **VirtualDub** (开源应用) 和 **e-CAMView**。

Linux 系统可以使用 V4L2 驱动程序和 VLC 媒体播放器进行视频流操作。可以在网上下载 VLC 媒体播放器。

Mac 平台可以使用 FaceTime、iChat、Photo Booth 和 Debut Video Capture 软件创建同 UVC 器件相连的接口，以便执行视频流操作。

8 故障排除

如果您对器件或固件有任何问题，首先需要获得更多数据并缩小问题的根源。为了更好实现上述目的，可以使能 UART 和 JTAG 调试。

可以使能 *cycx3_uvc.h* 文件中的“CX3_DEBUG_ENABLED”开关，以打印多种计数器、错误信息以及其它调试数据。使用 UART 线缆或 USB-UART 桥接器将电路板 (或 CX3 RDK) 上的 UART 端口连接到 PC。打开能够访问 PC 上 COM 端口的 Hyperterminal、Tera Term 或其它工具。启动传输前，请按照下面各项内容对 UART 进行配置：115,200 波特、无奇偶校验位、1 个停止位、无流控以及 8 个数据位。这样应足以捕获调试打印。

此外，可以使用 JTAG 调试器逐步调试固件。有关使用 JTAG 进行调试的详细信息，请查看编程员手册的第 12.2.2.3 节“执行与调试”(位于 Start > All Programs > Cypress > EZ-USB™ FX3 SDK 路径下)。

常見的問題解答已收錄在文件中, 請參閱下列的 KBAs:

- **KBA233853 - EZ-USB™ CX3 troubleshooting guide**
- **KBA91297 - CX3 Firmware: Frequently Asked Questions**
- **KBA91295 - CX3 Hardware: Frequently Asked Questions**
- **KBA91298 - CX3 Application Software / USB Driver: Frequently Asked Questions**

下面列出了一些常见的问题、原因以及解决方法。

问题：器件在 PC 上不枚举

原因 1：固件中的 API 可能遇到了错误条件。这样将调用错误的处理器，并且固件可能停止枚举或枚举失败。

解决方案 1：使用 JTAG 调试器或 UART/RS232 线缆并检查所有被调用的返回状态，查看是否发生了失败。然后，使用 API 指南，了解调用失败的原因。

原因 2：电路板的 USB 信号完整性较差。

解决方案 2：读取设计指南应用笔记，并确保电路板设计正确。更多細節請參考 Section 6.2

原因 3：未安装合适的驱动程序。

解决方案 3：在器件管理程序中卸载器件并重新安装它。您也可以修改 VID/PID，强制安装新器件。

原因 4：USB 描述符可能包含了错误数值。

解决方案 4：错误源通常为长度字段。请确保配置描述符中的 wTotalLength 字段中的总长度正确。此外，还要验证类别特定的视频控制和视频串流描述符中相同的字段。

问题：已经枚举器件，但是 USB 主机应用 (如 e-CAMView) 却显示为黑屏幕

原因 1：检查 CyCx3UvcAppThread_Entry 函数中的 glDMATxCount 变量，并验证它的递增情况。如果该变量不递增，可能是由于 CX3 和图像传感器之间的接口发生了故障。

解决方案 1：检查图像传感器是否正确连接到 CX3。然后，验证传感器的初始化情况、MIPI CSI-2 控制器的配置，并且验证是否正确选择了 GPIF II 总线宽度。

原因 2：如果您看到 glDMATxCount 的打印递增，需要验证正在发送的图像数据。首先，检查每帧所输出的总数据量。要想检查该数值，请查找包含了标头所设置的帧结束位的数据包长度。(表头的第二字节为 0x8E 或 0x8F，用以传输结束帧)。对于其它数据包，传输数据的长度等于 DMA 缓冲区大小减去尾部大小。

在一个帧上传输的图像数据 (不包含 UVC 标头) 的总量应为：宽度×高度×2。

解决方案 2：如果总图像尺寸小于 (或大于) 所需大小，则图像传感器所发送的数据大小将小于 (或大于) 所需的数据大小。检查图像传感器和 MIPI CSI-2 控制器配置。

问题：可以在高速模式下串流视频，但不能在超速模式下串联视频

原因：其原因可能是由于 SSTX/SSRX 线上的信号完整性较差。验证超速总线是否遵循 **AN70707** 的指南中的要求。您同时要确保仅使用了得到认证的线缆。

此外，可以使用允许记录 MIPI CSI-2 错误的 CX3_ERROR_THREAD_ENABLE 开关来查看 CSI-2 接口是否发生错误。更多有关错误类型的信息，请查看 **SDK API 指南**。

如果問題仍然沒有解決, 請在 **Infineon Developer Community** 找尋相關問題.

9 总结

本应用笔记描述了如何使用 EZ-USB™ CX3 实现带有 MIPI CSI-2 接口且符合 USB 视频类别的图像传感器。尤其重要的是，它显示了：

- 主机应用和驱动程序如何与 UVC 器件交互作用
- UVC 器件如何管理 UVC 特定的请求
- 如何配置 MIPI CSI-2 接口，以便接收来自通用图像传感器的数据
- 如何显示视频流并修改主机应用中的照相机属性
- 如何查找各种平台上的主机应用，包括开源主机应用项目
- 如何排除故障并调试 CX3 固件 (若需要)

10 相關資源列表

- AN75705 - EZ-USB™ FX3 入门
- AN70707 - EZ-USB™ FX3/FX3S/SX3 硬件设计指南和原理图检查表
- AN75779 - 如何使用 EZ-USB™ FX3 将图像传感器连接到 USB 视频类别 (UVC) 框架内
- Designing FX3/CX3-based USB Type-C products - KBA218460
- EZ-USB™ CX3 THEIA-CAM - 13MP PDAF UVC Camera Solution
- Ascella - CX3 THine ISP 13MP reference design kit (RDK)
- Denebola - USB 3.0 UVC reference design kit (RDK)
- EZ-USB™ CX3 webpage

文档修订记录

版本	提交日期	变更说明
**	2014-05-16	本文档版本号为 Rev**，译自英文版 001-90369 Rev**。
*A	2015-09-25	本文档版本号为 Rev*A，译自英文版 001-90369 Rev*C。
*B	2017-04-26	更新了徽标和版权。
*C	2017-05-31	没有技术更新。 完成日落评论。
*D	2022-06-13	更新至英飞凌模板 翻译自: 001-90369 Rev. *E

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-06-13
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2022 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?
Go to www.infineon.com/support

Document reference
001-92469 Rev. *D

重要提示

本文档所提供的任何信息**绝不应当**被视为针对任何条件或者品质而做出的保证（质量保证）。英飞凌对于本文档中所提及的任何事例、提示或者任何特定数值及/或任何关于产品应用方面的信息均在此明确声明其不承担任何保证或者责任，包括但不限于其不侵犯任何第三方知识产权的保证均在此排除。

此外，本文档所提供的任何信息均取决于客户履行本文档所载明的义务和客户遵守适用于客户产品以及与客户对于英飞凌产品的应用所相关的任何法律要求、规范和标准。
本文档所含的数据仅供经过专业技术培训的人员使用。客户自身的技术部门有义务对于产品是否适宜于其预期的应用和针对该等应用而言本文档中所提供的信息是否充分自行予以评估。

如需产品、技术、交付条款和条件以及价格等进一步信息，请向离您最近的英飞凌科技办公室接洽(**www.infineon.com**)。

警告事项

由于技术所需产品可能含有危险物质。如需了解该等物质的类型，请向离您最近的英飞凌科技办公室接洽。

除非由经英飞凌科技授权代表签署的书面文件中做出另行明确批准的情况外，英飞凌科技的产品不应当被用于任何一项一旦产品失效或者产品使用的后果可被合理地预料到可能导致人身伤害的任何应用领域。