# AN89661

# 使用 EZ-USB® FX3S™ 设计 USB RAID 1 磁盘

作者：Hingkwan Huen
参考例程：有
相关器件系列： EZ-USB® FX3S™
软件版本：N/A
相关应用笔记：AN75705 — FX3 入门

AN89661 描述了如何使用 EZ-USB® FX3S™设计并实现 USB 独立冗余磁盘阵列（RAID）1 级磁盘。它包含用于服务器启动存储应用的 USB RAID 1 级磁盘的设计示例，该示例有助于您开发服务器启动存储解决方案。

## 目录

1. 简介 ..... 1
2. RAID 简介 ..... 2
3. 系统概述 ..... 2
4. 功能概述 ..... 4
5. 硬件 ..... 4
6. 固件 ..... 6
   6.1 固件源 ..... 6
   6.2 回调操作 ..... 8
   6.3 批量存储操作 ..... 8
   6.4 自动重建 ..... 10
7. 操作流程 ..... 10
8. 性能 ..... 11
   8.1 USB⇔SD 卡 ..... 11
   8.2 SD 卡⇔SD 卡 ..... 11
9. 总结 ..... 11
附录 A：FX3S 片上 RAID USB 启动盘电路板布局视图 ... 12
附录 B：FX3S 片上 RAID USB 启动盘原理图 ..... 13
文档修订记录 ..... 14
全球销售和设计支持 ..... 15

## 1. 简介

USB 标准允许通过一个简单的标准化的接口连接多个外设。多年以来，USB 作为一个可靠、可扩展、快速、廉价、低功耗的可快捷插拔接口，普遍用在各种应用中。该应用笔记介绍了一种使用了 USB 的应用，即服务器上的 RAID 启动盘。

在各种服务器应用中，服务器虚拟化是信息技术的发展趋势，并成为服务器市场中增长最快的领域。服务器虚拟化是指使用软件将一个物理服务器分为多个虚拟服务器的过程。服务器虚拟化要求提供一个区别于主存储服务器的快速、可靠、紧凑的启动盘。

EZ-USB FX3S 是一个带有集成了双 SD 接口的片上 RAID 控制器，非常适合于低成本的 RAID 启动盘。该控制器不但作为一个超速 USB 批量存储设备，而且还通过片上 RAID 功能管理存储冗余状况，从而提高了它的可靠性。

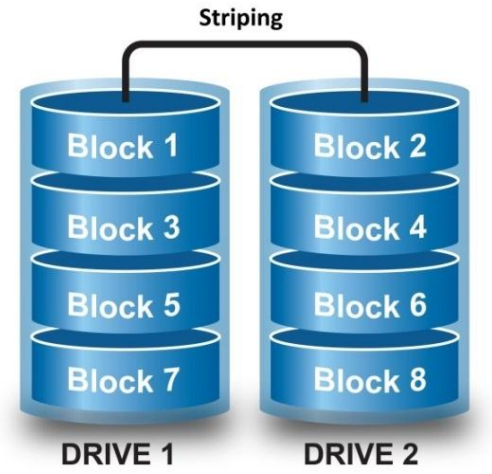
本应用笔记中的示例介绍了如何通过使用 EZ-USB FX3S 设计 USB RAID 1 磁盘应用在服务器启动存储。RAID 1 完全在 FX3S 中管理。该应用也可以作为标准的使用两个 SD 卡的批量存储设备。能够以现货供应形式购买 FX3S 片上 RAID USB 棒套件。可在本应用笔记的附录中查看完整的源代码。

## 2. RAID 简介

RAID 是一种存储技术，能够将多个存储盘组件组合成一个逻辑单元。 根据冗余情况和所需性能，可以按不同方式（即为 RAID 级别）将数据发送到磁盘。不同方案或架构的名称为“RAID 级别，”或“RAID”后加上某个数字。FX3S 支持 RAID 0 和 RAID 1。

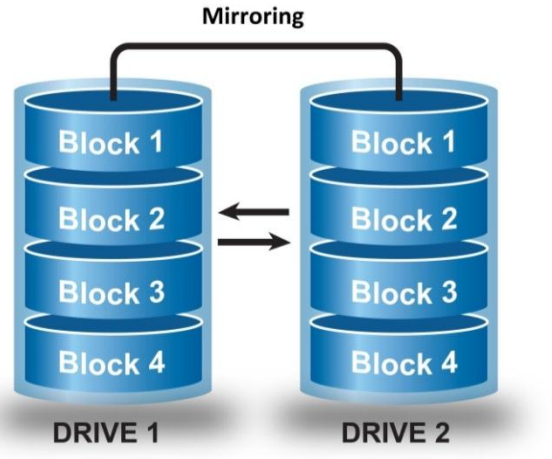
RAID 0（块级条带）没有冗余空间。因为它在两个物理存储组件间插入数据，从而提高存储性能，但它不具备容错功能。如果磁盘发生任何故障，都会破坏存储阵列。典型情况下，RAID 0 架构支持两个磁盘，如 图 1 所示。

图 1. RAID 0 级




对于 RAID 1（镜像），相同数据被写进两个磁盘，从而生成“一组镜像”，如 图 2 所示。两个盘中的任何一个都包含被请求的数据，服务读取请求。写入请求将更新该两个磁盘的条带。 创建一个阵列至少要用两个磁盘。

图 2. RAID 1 级




## 3.系统概述

图 3 显示的是使用 RAID 启动磁盘的服务器高级框图。出现服务器虚拟化技术后，单一物理服务器通过使用虚拟化软件可管理多个虚拟服务器。虚拟化软件通常位置是在与服务器主要存储器分离开的启动盘上。为了提高可靠性，RAID 1 配置的启动盘。通过使用图 4 中显示的 EZ-USB FX3S 片上 RAID USB 棒来实现 RAID 1 启动磁盘（图 3 中蓝框显示的部分）。

FX3S USB RAID 1 磁盘是主服务器系统的初始启动源。通过与平台控制器集线器（PCH）连接的快速 USB 3.0 接口，可快速启动服务器系统。典型的服务器系统还能通过使用连接至电路板管理控制器（BMC）的独立控制连接来管理 RAID 操作。

FX3S 片上 RAID USB 棒为 BMC 连接提供了一个独立的多媒体卡（MMC）从设备接口。因为 BMC 控制链接协议是由特定供应商提供的，所以本应用笔记不再介绍该协议。但您可以在 FX3S 固件中轻松添加并自定义该特性。要获取执行特定协议的详情，请在 cypress.com/go/support 网站上创建技术支持需求。

本应用笔记中的示例固件执行了一个普通的 RAID 1 解决方案，但不使用 BMC 控制。

图 4 中显示的 EZ-USB FX3S 片上 RAID 棒硬件由 Pactron 生成，并可从 pactronstore.com/products/cypress-fx3s.html 上获取该硬件。

本应用笔记的附录提供了硬件原理图和示例固件源代码。

图 3. 服务器概述

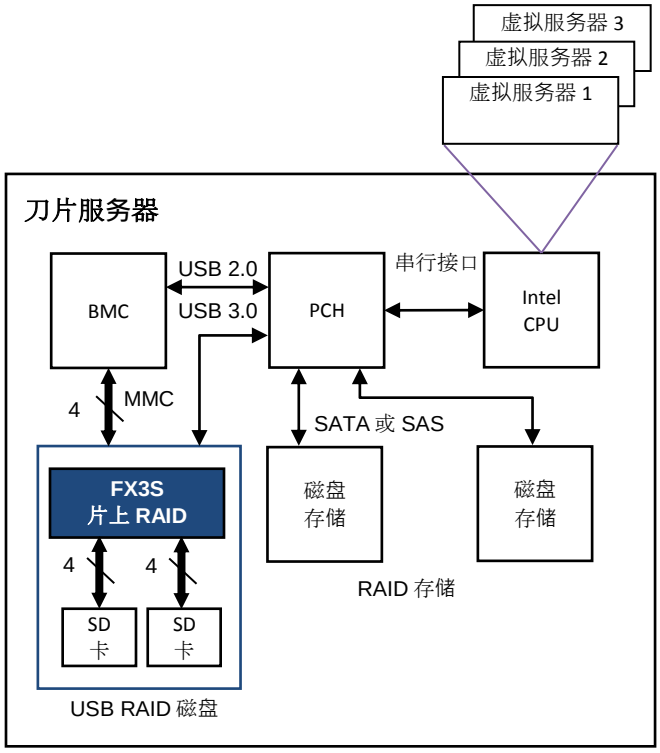
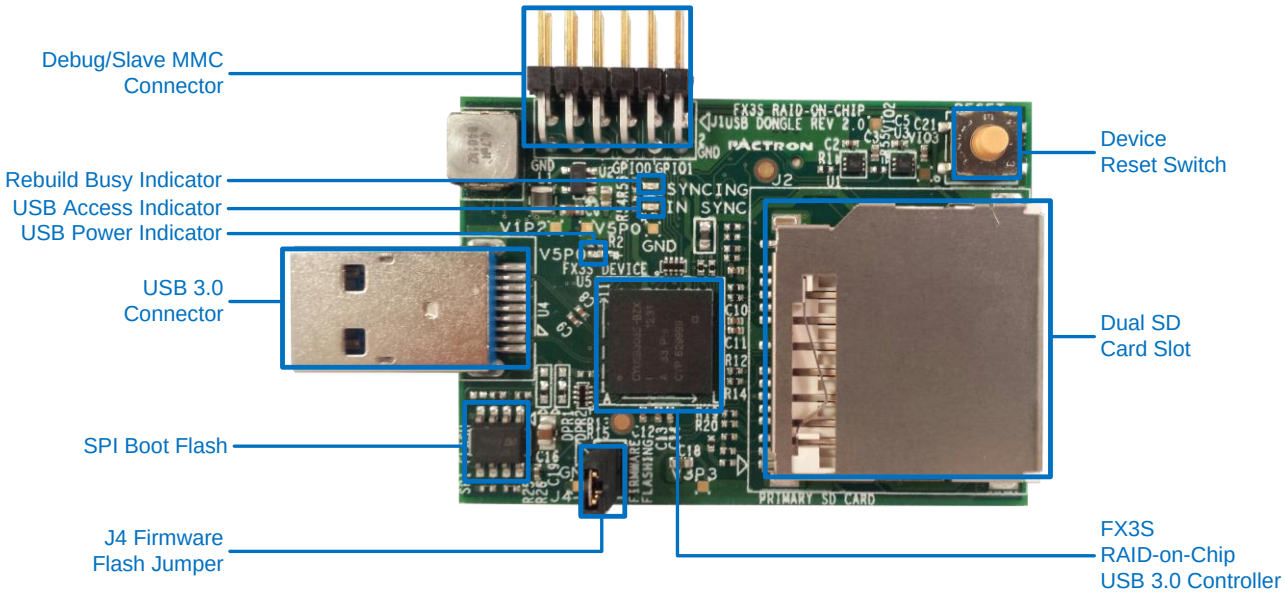



图 4. EZ-USB FX3S 片上 RAID USB 棒



## 4. 功能概述

FX3S USB RAID 1 磁盘作为一个标准的 USB 3.0 批量存储设备使用。本应用笔记提供的示例固件使用了两个 SD 卡来执行 RAID 1 功能。在 USB 主机 PC 上，该设备作为单个镜像的存储卷被枚举。在 FX3S 内管理所有 RAID 操作。用户不需要外部 RAID 处理。

FX3S 能识别两个 SD 卡：主卡和附属卡。对于读取操作，返回到 USB 主机的数据是来自主卡的。如果主卡发生故障或被取出，则会继续从附属卡读取数据。对于写操作，来自 USB 主机的数据可同时被写入到主卡和附属卡内。如果某个卡发生故障或被取出，则该卡上的数据传输将被中止，但另一个卡仍会正常执行。在两种情况下，每当两个卡的一个发生错误时，FX3S 维持无缝的 USB 磁盘操作。

RAID 1 示例固件支持当替换主卡或附属卡时自动重建存储体。运行磁盘时，虽然取出了某一个卡，但 USB 主机不会发现任何操作中断。如果插入新卡，通过将有效卡的全部内容复制到新插入的卡内，RAID 1 固件自动执行重建过程。

在重建过程中，LED 重建忙碌指示灯仍然亮着，如 图 4 所示。从 USB 主机到 SD 卡的存储访问被阻止，直到完成这个重建过程为止。USB 主机的读/写请求暂时被拒绝，但它会持续发送请求，直到该重建过程完成为止。主机操作系统自动执行该操作，而不需用户干预。

总之，FX3S USB RAID 1 磁盘执行下面各操作：

- 作为标准的批量存储设备使用，并以 USB 超速、高速或全速运行。
- 使用两个 SD 3.0 (UHS-I)卡
- 如果两个 SD 卡的空间大小不同，将报告并使用空间更小的那个卡。
- 在普通的批量存储操作中，执行数据镜像操作
- 如果移除某个 SD 卡，不会中断对另一个卡的访问
- 替换某个卡时，自动执行数据是同步化操作
- 处理 SD 卡的热插拔事件（移除和插入）

## 5. 硬件

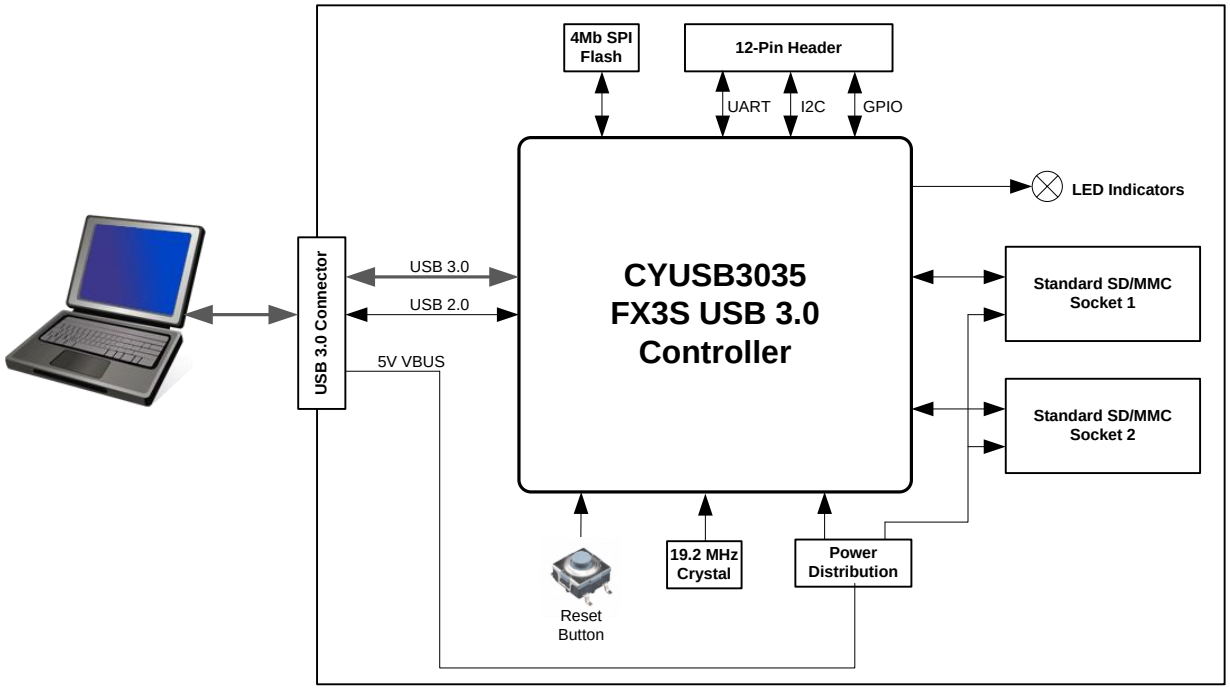
图 5 中显示的 USB 片上 RAID USB 棒硬件设计使用了带有内置 RAID 管理的 FX3S 来执行超速 USB 批量存储设备。电路板上装有下面各个组件：

- 支持超速、高速和全速的 USB 3.0 外设接口
- 两个独立的存储端口，每个端口支持 SD/SDIO 3.0 (UHS-I)和嵌入式 MMC 4.41 器件
- 4 Mb 的 SPI 闪存，用于 FX3S 固件存储
- 19.2 MHz 晶振输入作为系统时钟源使用
- 12 引脚插头，包括用于调试的 UART 和一个 MMC 4.2 接口（不使用于本应用笔记中的 RAID 1 示例）
- 电源、USB 访问和重建操作的 LED 指示灯
- 硬件系统复位按键

通过 USB 接口，总线给片上 RAID USB 棒供电。所有 FX3S 和 SD 卡电源轨均来自 5 V VBUS（由 USB 主机提供）。

FX3S 包括一个 200 MHz ARM9 MCU， 大小为 512KB 的系统存储器和一个非常适合本应用的完整的外设阵列。虽然已经设置了电路板，使之从板上 SPI 闪存器件中引导加载所需的固件，但为了开发系统，您还可以配置电路板，使之通过 USB 进行引导加载。

图 5. 硬件概述


4Mb SPI Flash
12-Pin Header
UART
I2C
GPIO
LED Indicators
USB 3.0 Connector
USB 3.0
USB 2.0
5V VBUS
CYUSB3035
FX3S USB 3.0
Controller
Standard SD/MMC Socket 1
Standard SD/MMC Socket 2
19.2 MHz Crystal
Power Distribution
Reset Button

# 6. 固件

应用笔记和示例固件项目文件位于 FX3MSC_RAID1 文件夹压缩文件内。该示例项目可无需修改地编译并生成一个二进制固件文件。通过使用赛普拉斯 USB Control Center（控制中心）可以将该二进制文件加载到 FX3S 内。进行下载前，请确保已经打开了电路板上的 J4，以允许 FX3S 进入 Bootloader 模式。

更多有关如何使用 FX3S 固件项目进行操作以及如何使用赛普拉斯 USB Control Center 下载固件，请参阅 AN75705 — EZ-USB FX3 入门。

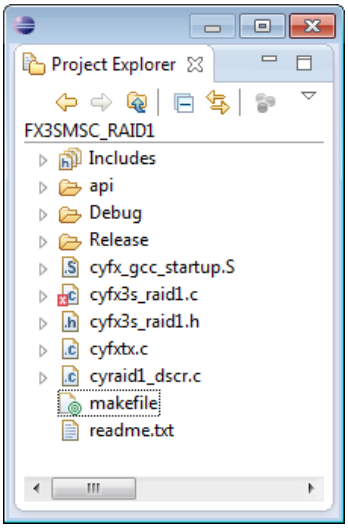
## 6.1 固件源

RAID 1 固件应用运行于实时操作系统（RTOS）顶层，这样可以有效地管理 FX3S 内部资源。 FX3S 固件通过一个应用编程接口（API）库函数集与 FX3S 硬件外设通信。该函数集抽象了低级设备，从而大大简化开发工作。

图 6 显示了示例 RAID 1 固件项目中的文件和目录：

- Includes：包含C库头文件和项目特定头文件的目录。
- api：包含FX3S API库的目录（二进制格式）。
- “Debug”和“Release”：可通过调试配置或发布配置来编译项目。“Debug”和“Release”目录包含相应编译配置的输出。
- *cyfx_gcc_startup.S*：FX3S器件上ARM9内核的启动代码。该汇编源文件位于GNU汇编器语句后。自我引用的GNU是GNU中非UNIX的缩写。它是自由软件基金会开发的UNIX兼容软件系统。
- *cyfx3s_raid1.c*：RAID 1 USB批量存储器件示例的主要实现。
- *cyfx3s_raid1.h*：RAID 1批量存储应用的常量定义和函数声明。
- *cyfxtx.c*：FX3S API库要求的ThreadX RTOS包装器和工具函数。
- *cyraid1_descr.c*：USB描述符定义。
- *makefile*：用于管理和组织代码编译的“shell”指令文件。
- *readme.txt*：FX3MSC_RAID1项目的简单说明。

图 6. RAID 1 固件源




所有的 RAID 函数在文件 cyfx3s_raid1.c 中实现。表 1 总结了这些函数。

表 1. cyfx3s_raid1.c 中执行 RAID 1 应用的函数

| 函数类型 | 参考章节 | 函数名称 | 说明 |
|---|---|---|---|
| RTOS | 6.2 | main() | 该函数是初始化 ARM 内核和 C 库后的固件 C 代码输入点。使用它可以设置缓存、配置 FX3 I/O 并启动 RTOS 内核。 |
| | 6.2 | CyFxApplicationDefine() | 该函数定义了 RTOS 执行的应用线程。 |
| | 6.2、6.3、6.4 | MscAppThread_Entry() | 该函数是 RAID 1 应用线程，使用它可以执行下面各操作：<br>调用应用级初始化<br>管理批量存储器件的内部状态转换 |
| 批量存储应用级初始化 | 6.3 | CyFxMscApplnInit() | 该函数调用 FX3S 器件级初始化函数，以执行下面各项任务：<br>1. 初始化 GPIO。<br>2. 初始化两个 SD 存储端口。<br>3. 初始化 USB 端口。<br>4. 注册事件回调。<br>5. 注册 USB 描述符。<br>6. 初始化所需要的 DMA 通道。<br>7. 使能 USB 与主机的连接。 |
| | 6.3 | CyFxMscApplnDebugInit() | 该函数初始化 FX3 UART 模块，以打印调试信息。 |
| | 6.3 | CyFxMscApplnDmaInit () | 该函数分配缓冲存储器，并初始化 USB 和存储端口间的 DMA 通道。在普通条件下，当插入两个 SD 卡时，会初始化两个 DMA 配置：<br>对于读操作：一对一的单一通道 DMA，用于将 USB BULK-IN 端点和主 SD 卡连接起来。<br>对于写操作：一对一的多播通道 DMA，用以将 USB BULK-OUT 端点和两个 SD 卡连接起来。 |
| | 6.3 | CyFxMscApplnDmaReInit () | 当两个 SD 卡中的某一个处于断接状态，然后再次连接时，可使用该函数重新配置一对一的多播通道 DMA。 |
| FX3S 器件级初始化 | 6.3 | CyFxRdDmaInit() | 对于读操作，该函数创建一对一的单通道 DMA，用以将 USB BULK-IN 端点和主 SD 卡连接起来。 |
| | 6.3 | CyFxWrErrDmaInit() | 当两个 SD 卡中的某一个处于断接状态时，可使用该函数调用多传输通道 DMA，并将其重新配置为单通道 DMA。 |
| | 6.3 | CyFxMscApplnSibInit() | 该函数初始化两个 SD 存储控制器。 |
| | 6.3 | CyFxMscApplnGpioInit () | 该函数初始化应用所需要的 GPIO。 |
| 批量存储应用回调 | 6.2 | CyFxMscApplnUSBEventCB() | 该回调函数处理普通的 USB 事件（如复位，连接与断接等）。 |
| | 6.2 | CyFxMscApplnUSBSetupCB() | 进行枚举和控制传输过程中，该回调函数会处理 USB 设置事件。 |
| | 6.2 | CyFxMscApplnSibCB() | 该回调函数处理存储端口事件（如插入和移除卡，数据传输完成等）。 |
| | 6.2 | CyFxMscApplnDmaCb() | 该回调函数处理一对一的单通道 DMA 事件。 |
| | 6.2 | CyFxMscApplnMultiDmaCb() | 该回调函数处理一对一的多播通道 DMA 事件。 |
| 批量存储应用操作 | 没有引用 | CyFxMscApplnResetCtlr() | 该函数复位对 USB 数据路径的清除。 |
| | 6.3 | CyFxMscApplnParseCbw() | 该函数解析并处理所接收的存储指令。 |
| | 没有引用 | CyFxMscApplnSendDataToHost() | 该函数对发送至 BULK-OUT 端点的存储指令执行状态分为不同的阶段，以供主机读取。 |
| | 没有引用 | CyFxMscApplnSendCsw() | 该函数对发送至 BULK-OUT 端点的存储指令执行状态分为不同的阶段，以供主机读取。 |
| | 6.4 | CyFxMscApplnQueryDevStatus() | 该函数将探测两个 SD 存储端口，并报告其状态。 |
| | 没有引用 | CyFxAppLedOn() | 该函数通过设置 GPIO 来打开 LED。 |
| | | CyFxAppLedOff() | 该函数通过设置 GPIO 来关闭 LED。 |

## 6.2 回调操作

由于 RTOS 管理系统资源，RAID 1 固件执行将由事件驱动。由外设（USB 和 SD 端口）和内部 DMA 系统生成事件。在初始化固件时，通过注册回调函数来处理这些事件。

当开始执行 RAID 1 固件时，该固件会先执行一系列 ARM9 内核、GNU 工具链库和 RTOS 的初始化，然后才会进入 *cyfx3s_raid1.c* 中的 main()函数。通过调用 main(). 中的 CyU3PKernelEntry()， 可以启动 RTOS。在 RTOS 启动线程调度前，至少要创建一个线程，以执行应用任务。在 RAID 1 示例项目中，将使用 MscAppThread_Entry()应用线程。通过配置外设接口，并注册用来处理 USB、存储和 DMA 事件的事件回调函数，可以启动该线程。这些回调函数包括：

- CyFxMscApplnUSBEventCB: 该回调函数会处理下面各 USB 事件：
  - *CY_U3P_USB_EVENT_SUSPEND*： USB 暂停事件。器件进入低功耗模式。
  - *CY_U3P_USB_EVENT_DISCONNECT*：USB 断连事件。器件与主机断连。
  - *CY_U3P_USB_EVENT_RESET*：USB 复位事件。主机接收一个复位。
  - *CY_U3P_USB_EVENT_CONNECT*：USB 连接事件。器件连接至主机。
  - *CY_U3P_USB_EVENT_SETCONF*：主机“配置”器件，以完成枚举过程。
- CyFxMscApplnUSBSetupCB：回调函数处理主要在枚举过程中发生的 USB 设置事件：
  - CY_FX_MSC_USB_STANDARD_REQ：接收来自主机的标准设置请求。
  - CY_FX_MSC_USB_CLASS_REQ：接收来自主机的特定类别的请求。
  - CY_FX_MSC_USB_VENDOR_REQ：接收来自主机的供应商特定请求。使用自定义供应商主机驱动器时，该函数才会使用。
- CyFxMscApplnSibCB ：该回调函数处理存储端口事件：
  - *CY_U3P_SIB_EVENT_XFER_CPLT*：完成了 DMA 传输。
  - *CY_U3P_SIB_EVENT_INSERT*：检测到两个 SD 插座中有一个有存储卡插入。
  - *CY_U3P_SIB_EVENT_INSERT*：检测到两个 SD 插座中有一个有存储卡移除 。
  - *CY_U3P_SIB_EVENT_DATA_ERROR* 两个卡中的一个发生了错误。
  - *CY_U3P_SIB_EVENT_ABORT*：读/写操作已被中止。
- CyFxMscApplnDmaCb：该回调函数会处理一对一的 DMA 通道的事件。在一对一的通道中，有一个数据发送端和一个数据接收端。在下面各种情况下，通常会发生有关一个到一个通道的事件：通过 USB 接收批量存储指令块包（CBW）、通过 USB 发送一个批量存储指令状态包（CSW），或从某个存储端口进行读取操作。
  - *CY_U3P_DMA_CB_SEND_CPLT*： 已完成从 DMA 输出数据
  - *CY_U3P_DMA_CB_RECV_CPLT*： 已完成向 DMA 输入数据。
- CyFxMscApplnMultiDmaCb：该回调函数处理一对多的 DMA 通道的下面事件。在一对多的传输中，有一个数据提供端和多个数据接收端（在这种情况下，接收端有两个 SD 卡）。当执行存储写入指令，将来自 USB 的数据写入两个存储卡内时，通常会发生这些事件。
  - *CY_U3P_DMA_CB_RECV_CPLT*： 已完成从 DMA 传出数据。
  - *CY_U3P_DMA_CB_PROD_EVENT*：已完成向 DMA 传入数据。

## 6.3 批量存储操作

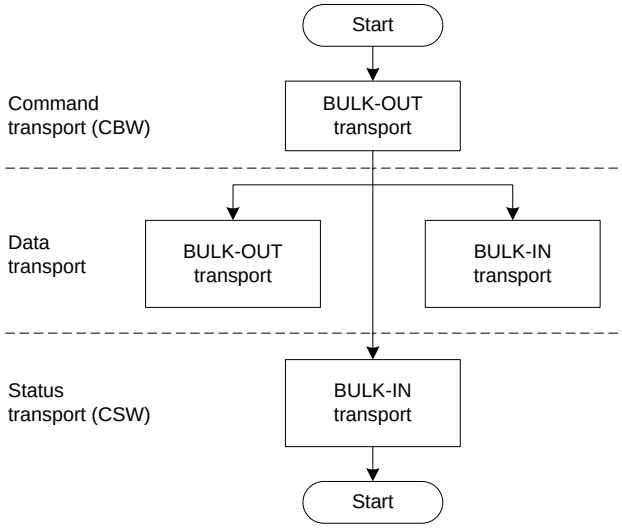
RAID 1 固件完成器件级和应用级初始化序列后，将继续在 MscAppThread_Entry()上执行操作，并执行无限循环来等待器件生成 USB 批量存储应用事件。

RAID 1 固件应用将使用 bulk-only 传输（BOT）来执行标准的 USB 批量存储类别（MSC）。BOT 定义 MSC 操作的三个基本阶段：

- 指令传输：主机 PC 通过 BULK-OUT 传输将指令发送给器件。该指令数据包被定义为 CBW，并且 BOT 必须始终以 CBW 开始。
- 数据传输：用以在主机和器件间传输数据。例如，对于读/写指令，将在数据传输阶段发送不同存储区内的实际数据。该过程使用了多总线的数据传输。在数据传输中，可通过 BULK-OUT 或 BULK-IN 传输形式进行数据传输。
- 状态传输：通过 BULK-IN 传输形式将来自器件的指令执行结果传输给主机。该状态数据包由 CSW 定义。BOT 始终必须以 CSW 结束。

图 7 显示的是高级 MSC 阶段转换。

图 7. USB MSC 发送阶段




为支持标准的 MSC 协议，将在 MscAppThread_Entry()中处理下面各事件：

- *CY_FX_MSC_RESET_EVENT*：批量存储器件已被复位。
- *CY_FX_MSC_SETCONF_EVENT*：已完成枚举操作，并且需要设置器件，以进行 MSC 操作。
- *CY_FX_MSC_CBW_EVENT*：已接收到 MSC CBW 数据包。接收到该事件时，将解析并执行该指令（使用 SCSI 格式）。
- *CY_FX_MSC_DATASENT_EVENT*：在数据传输阶段，数据已被发送给主机，并且固件会使器件进入状态传输阶段。
- *CY_FX_MSC_SIBCB_EVENT*：该事件要求批量存储器件设置存储端口，以进行下一个数据传输。
  - 如果器件处于指令传输阶段，固件将设置接收缓冲器，以接受来自主机的 CBW 数据包。
  - 如果器件处于数据传输阶段，根据当前指令的数据方向，固件将设置发送或接收缓冲器。
  - 如果器件处于状态传输阶段，则固件将更新 CSW 数据包，并将数据包发送至主机。

RAID 1 固件通过内部状态机管理批量存储器件。在图 7 中显示了每个 MSC 传输状态的定义。下面 C 代码示例显示的是如何在固件中定义这些状态：

```
typedef enum
{
    /*    Inactive    state,    waiting    for
SET_CONFIG. */
    CY_FX_MSC_STATE_INACTIVE = 0,

    /* Waiting to queue a CBW command. */
    CY_FX_MSC_STATE_CBW,

    /* transitional state that it tells USB
host it is ready to receive a CBW command.
*/
    CY_FX_MSC_STATE_WAITING,

    /* Waiting to complete data transfer for
a command. */
    CY_FX_MSC_STATE_DATA,

    /* Waiting to send CSW for a command. */
    CY_FX_MSC_STATE_STATUS,

    /*  Waiting  for  host  to  read  out  CSW
packet. */
    CY_FX_MSC_STATE_CSW
} CyFxMscFuncState;
```

当进行 FX3S USB RAID 磁盘枚举时，USB 主机将 CBW 数据包发送到 BULK-OUT 端点。FX3S 将 CBW 数据保存到 glMscCbwBuffer 缓冲器内。当器件准备好执行指令时，器件的当前状态将转换为 CY_FX_MSC_STATE_CBW ，将产生 MSC CBW 事件 CY_FX_MSC_CBW_EVENT_FLAG 。在 USB 成功枚举或完成最后的指令后，器件状态将被设置为 CY_FX_MSC_STATE_CBW。

在 MscAppThread_Entry() 检测到 CY_FX_MSC_CBW_EVENT_FLAG 事件后，它会调用 CyFxMscApplnParseCbw() ，以分析 glMscCbwBuffer 中存储的指令。根据指令类型，器件状态将被转换为 CY_FX_MSC_STATE_DATA 或 CY_FX_MSC_STATE_STATUS 状态。

如果使用 CY_FX_MSC_SCSI_READ_10 或 CY_FX_MSC_SCSI_WRITE_10 指令，将产生存储回调事件 CY_FX_MSC_SIBCB_EVENT_FLAG。MscAppThread_Entry()函数通过设置 SD 卡和 USB 端点间的合适 DMA 通道来处理该事件。当 USB 主机启动传输时，数据将通过 DMA 通道输入到 SD 卡或从 SD 卡内输出，直至达到数据计数为止。当 DMA 传输完成时，两个 DMA 回调函数中的某一个将接收传输完成事件，并产生 MSC 数据发送事件 CY_FX_MSC_DATASENT_EVENT_FLAG 。MscAppThread_Entry() 函数处理 MSC 数据发送事件后，指令状态将被更新。器件状态变成 CY_FX_MSC_STATE_STATUS。

当器件处于 CY_FX_MSC_STATE_STATUS 状态，将构建 MSC CSW 数据包，并将该数据包发送给 USB 主机。发送完 CSW 后，器件将返回到 CY_FX_MSC_STATE_CBW 状态，并准备好执行下一条指令。

### 6.4 自动重建

RAID 1 固件支持自动重建存储体，以恢复存储冗余状态。MscAppThread_Entry() 通过监控下面两个状态标志来管理存储体的重建情况：过期状态和存储卡的状态，每个存储被设置为两种状态中的一种。

每当卡操作失败或从插座取出存储卡时，过期状态标志将被设置。一般在成功重建后，当两个卡都正常工作，并都被同步化时，该标志将被清除。默认情况下，插入两个 SD 卡后，在初始通电时，会设置该过期标志。固件始终假定在初始通电条件下两个 SD 卡是同步的。推荐您在初始通电后格式化磁盘，以确保在启动时两个卡均被同步化。

当插入（CY_U3P_SIB_EVENT_INSERT）或移除（CY_U3P_SIB_EVENT_REMOVE）某个卡时，卡存在标志将被更新。这些插入/取出事件由存储回调函数 CyFxMscApplnSibCB()处理。如果移除某个存储卡时，它的存在标志将被清除。当插入某个卡时，将产生 CY_U3P_SIB_EVENT_INSERT 事件，并且 CyFxMscApplnSibCB() 调用 CyFxMscApplnQueryDevStatus() 以检测并初始化该卡。如果初始化成功，存储卡存在标志将被设置。

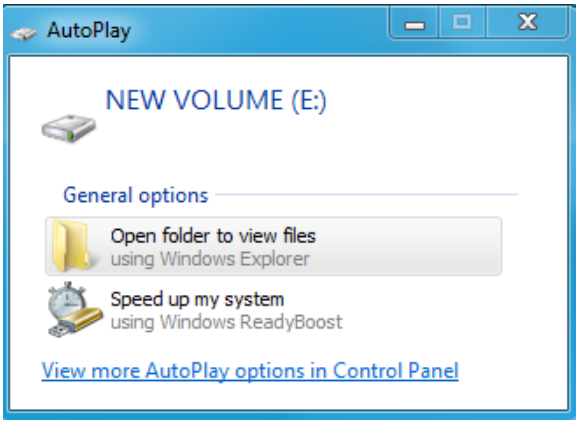
如果 MscAppThread_Entry()检测到卡存在标志和过期状态标志均被设置，那么它会通过调用 CyWbRaidFullCardSync()启动存储体的重建过程。 卷重建过程持续，直到过期状态被清除为止。当成功将 SD 卡中的全部内容传输到新插入的存储卡后，将在恢复冗余状态时清除过期状态。在重建过程中，USB 主机不能访问磁盘，因为主 SD 卡正在服务该过程。在这种情况下，主机继续尝试访问磁盘，直到重建操作成功，并且磁盘再次进入可访问状态为止。

## 7. 操作流程

可通过执行下面的操作流程来测试 FX3S USB RAID 磁盘支持的功能：

1. 将 FX3S USB RAID 磁盘插入到任何 USB 插槽（超速、高速或全速）内。可以看到电源指示 LED 灯亮。当前 USB 对 SD 卡进行的任何访问都会使 LED USB 访问指示灯闪烁。图 4 显示的是这些 LED 指示灯的位置。
2. USB 枚举后，Windows 资源管理器 AutoPlay 窗口中将显示一个磁盘驱动，如图 8 所示。

图 8. Windows 资源管理器 AutoPlay 窗口




3. 使用 FAT 或 FAT32 文件系统对该磁盘进行格式化处理。注意：该磁盘显示为一个单一的磁盘存储体，但实际上，它还隐藏了两个映射 SD 卡。
4. 将视频文件移入卡中，并播放该视频。
5. 在播放视频期间，移除其中一个 SD 卡。可以发现该视频继续播放，而不被中断。这表示 RAID 设计的安全特性 — 当某个卡操作失败或被移除时，系统仍使用单个卡继续运行。
6. 删除第四步骤移除的 SD 卡中的内容，并重新插入该卡。这会触发重建操作。 图 4 中的 LED 重建忙碌指示灯发亮，并且禁止 PC 对卡执行访问。由于内部缓冲，该视频可能播放一段时间，但它最终会停止。
7. 当完成重建操作时，LED 重建忙碌指示灯将被关闭，并且该视频恢复播放。这表示了重建操作不需要用户干预。
8. 按照第四步骤介绍的操作移除同一个 SD 卡，并检查其内容。卡里面的内容应该是与 FX3S 片上 RAID USB 棒上连接的 SD 卡的内容的副本。

# 8. 性能

RAID 1 固件在 EZ-USB FX3S 片上 RAID USB 启动磁盘上运行时，本节中介绍的特性是在下述条件中测得的：

- 测量工具：CrystalDiskMark v3.0.2 x64
- SD 卡类型：SanDisk Extreme Pro 8 GB
- USB 主机：Lenovo ThinkPad T430，主机运行 Windows 7 x64 的 Intel Ivy 集成了 USB 3.0

普通情况下，推荐使用相同的 SD 卡，以获得最佳性能。如果使用的两个卡大小和速度都不一样，会按空间较小的卡使用于逻辑 RAID 1 磁盘存储体。性能也受速度较低的卡的限制。

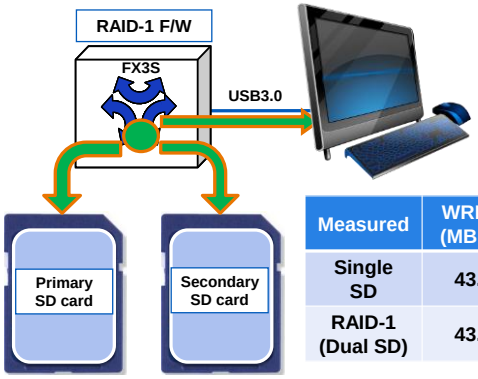
## 8.1 USB⇔SD 卡

USB 3.0 主机使用普遍的磁盘基准测试工具（CrystalDiskMark）来读/写驱动，从而测量 USB 到 SD 卡的传输性能。可从下面的链接中下载该工具

crystalmark.info/software/CrystalDiskMark/index-e.html。

图 9 显示的是预期 USB⇔ SD 卡的性能。

图 9. USB 3.0⇔SD 性能



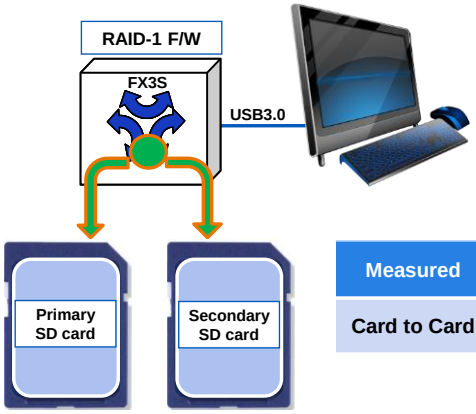

| Measured | WRITE (MBps) | READ (MBps) |
|---|---|---|
| Single SD | 43.2 | 43.6 |
| RAID-1 (Dual SD) | 43.7 | 43.7 |

## 8.2 SD 卡⇔SD 卡

先移除两个 SD 卡中的某一个然后再插入它，这样可以启动 SD 卡至 SD 卡间的传输。这样可以触发重建操作，有效卡的全部内容将被传输到新插入的存储卡内。

图 10 显示的是预期 SD⇔ SD 卡的性能。

图 10. SD⇔SD 新能




| Measured | WRITE (MBps) |
|---|---|
| Card to Card | 43.5 |

# 9. 总结

本应用笔记介绍了用于服务器启动存储应用的片上 RAID USB 批量存储器件（使用 EZ-USB FX3S）。本文档还附带了一个相关的电路板设计原理图和示例固件源代码。

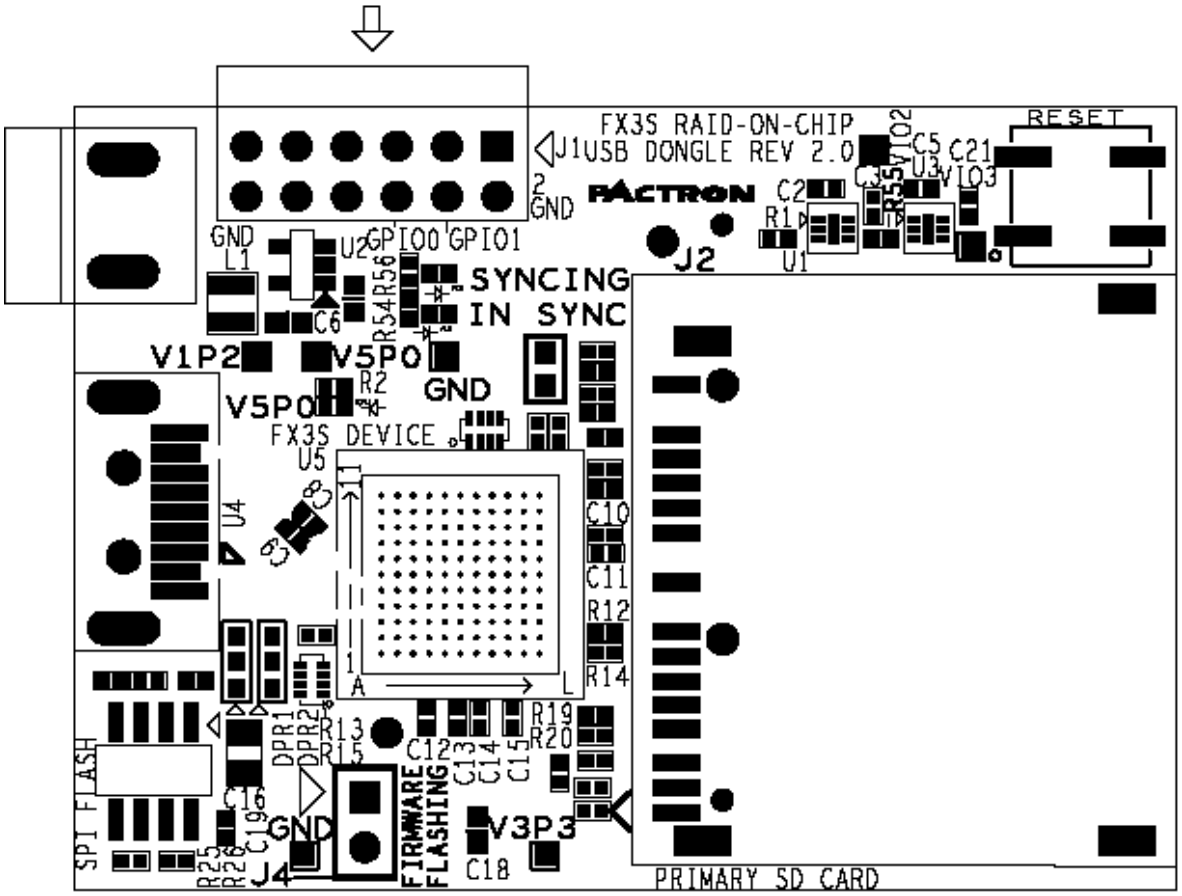
# 10. 关于作者

名称： Hingkwan Huen

职务： 系统工程师负责人

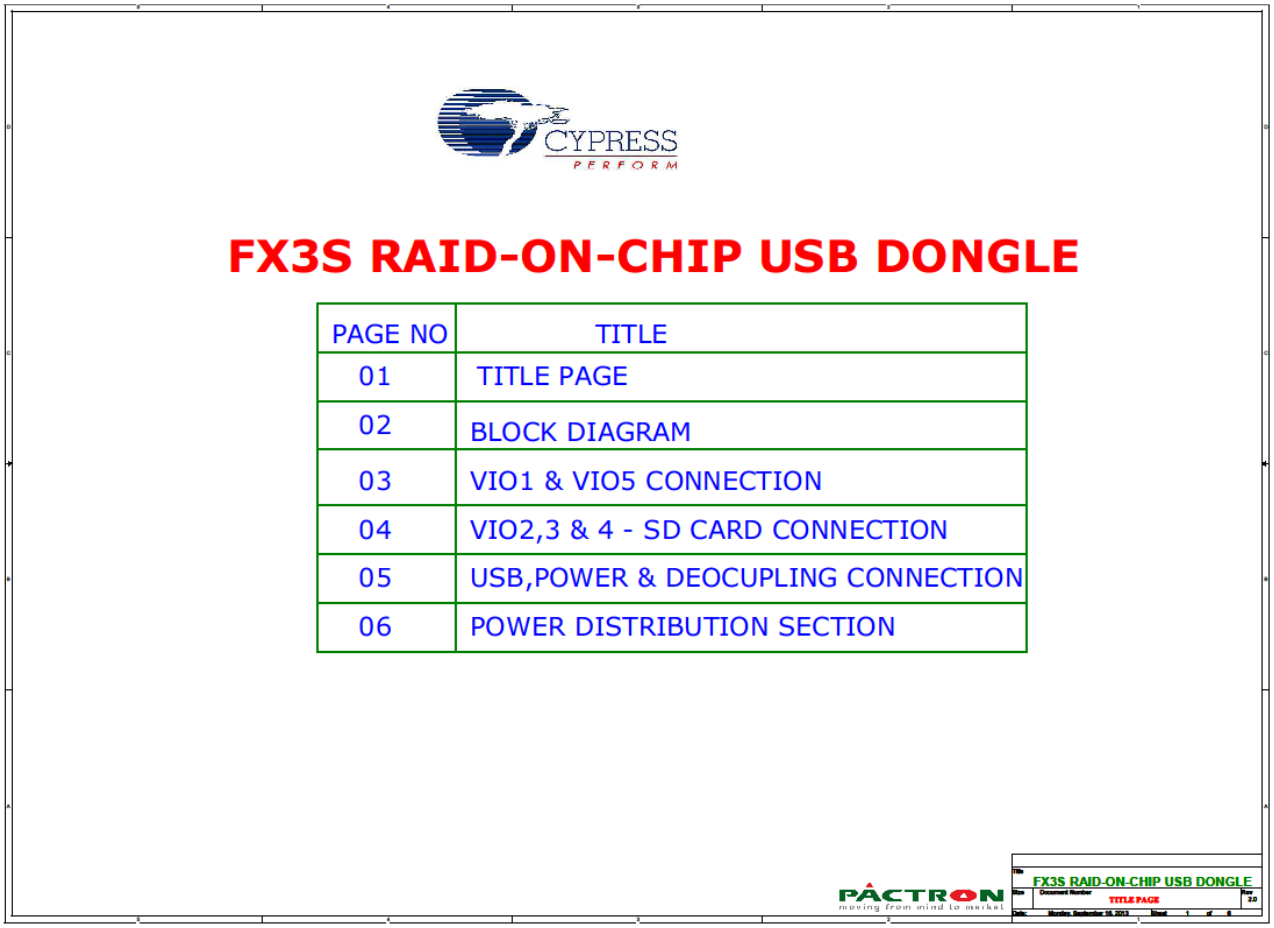
## 附录 A：FX3S 片上 RAID USB 启动盘电路板布局视图

请双击该图像，以打开电路板布局的顶层和底层视图。



## 附录 B：FX3S 片上 RAID USB 启动盘原理图

请双击该图像，以打开电路板原理图文件。



FX3S RAID-ON-CHIP USB DONGLE

| PAGE NO | TITLE |
|---|---|
| 01 | TITLE PAGE |
| 02 | BLOCK DIAGRAM |
| 03 | VIO1 & VIO5 CONNECTION |
| 04 | VIO2,3 & 4 - SD CARD CONNECTION |
| 05 | USB,POWER & DEOCUPLING CONNECTION |
| 06 | POWER DISTRIBUTION SECTION |

# 文档修订记录

文档标题： AN89661 — 使用 EZ-USB® FX3S™的 USB RAID 1 磁盘设计

文档编号：001-92167

| 修订版 | ECN | 变更人 | 提交日期 | 修订说明 |
|---|---|---|---|---|
| ** | 4347623 | HHLL | 04/15/2014 | 本文档版本号为 Rev**，译自英文版 001-89661 Rev*A。 |
| *A | 5709635 | DBIR | 04/25/2017 | 更新到新模板。<br>完成日落评论。 |

# 全球销售和设计支持

赛普拉斯公司拥有一个由办事处、解决方案中心、原厂代表和经销商组成的全球性网络。如欲查找离您最近的办事处，请访问赛普拉斯所在地。

**产品**

| | |
|---|---|
| ARM® Cortex® 微控制器 | cypress.com/arm |
| 汽车级产品 | cypress.com/automotive |
| 时钟与缓冲器 | cypress.com/clocks |
| 接口 | cypress.com/interface |
| 物联网 | cypress.com/iot |
| 存储器 | cypress.com/memory |
| 微控制器 | cypress.com/mcu |
| PSoC | cypress.com/psoc |
| 电源管理 IC | cypress.com/pmic |
| 触摸感应 | cypress.com/touch |
| USB 控制器 | cypress.com/usb |
| 无线连接 | cypress.com/wireless |

**PSoC® 解决方案**

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

**赛普拉斯开发者社区**

论坛 | WICED IoT 论坛| 项目 | 视频 | 博客 | 培训 | 组件

**技术支持**

cypress.com/support

EZ-USB 和 PSoC 是赛普拉斯半导体公司的注册商标且 EZ-USB FX3S 是赛普拉斯半导体公司的商标。

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

赛普拉斯半导体公司，2013-2017 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。
在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。
赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，WICED，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。