MPLAB® Harmony 应用程序帮助

MPLAB Harmony 集成软件框架 v1.07

© 2017 Microchip Technology Inc. 版权所有

应用程序帮助

本部分介绍 MPLAB Harmony 中包含的各种应用程序演示。

说明

应用程序决定如何使用 MPLAB Harmony 库 （器件驱动程序、中间件和系统服务）来完成一些有用的功能。在 MPLAB Harmony 系统中，可能有一个主应用程序、多个独立应用程序，也可能有一个或多个特定于操作系统 （Operating System，OS）的应用程序。应用程序通过定义良好的接口与 MPLAB Harmony 库交互。应用程序可能在严格查询环境中工作，它们可能由中断驱动、在特定于 OS 的线程中执行，也可能编写得非常灵活，可以针对其中的任何环境轻松配置。应用程序通常属于以下类别之一。

演示应用程序

演示应用程序 （随 MPLAB Harmony 提供或单独安装）用于演示一个或多个 MPLAB Harmony 库的典型或有趣的使用模型。演示应用程序可以演示实际问题的现实解决方案。

示例应用程序

示例应用程序是随 MPLAB Harmony 提供的一些非常简单的应用程序，用作说明如何使用库的各个功能的示例。它们通常无法独立实现任何有用的功能，主要是一些说明如何使用库的文档。

音频演示

本部分介绍在 MPLAB Harmony 安装中提供的音频演示。

简介

MPLAB Harmony 音频演示帮助。

说明

本帮助文件包含关于 MPLAB Harmony 音频演示应用程序 （包含在 MPLAB Harmony 库分发包中）的说明和关联信息。

演示

说明如何运行演示应用程序。

audio_microphone_loopback

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

在该演示应用程序中，使用 AK4642 编解码器驱动程序来设置 AK4642 编解码器，它是音频编解码器子板 AK4642EN 上的板载器件，可以通过子板的板载麦克风接收音频数据并通过板载耳机将这些音频数据发送出去。音频编解码器子板 AK4642EN （可从 Microchip 购买）可与 PIC32 蓝牙音频开发工具包连接。

该演示应用程序还说明了对于需要在编解码器和 PIC32 单片机之间进行双向数据通信的应用程序，如何配置 I2S 客户端和设置 DMA 通道。

关于驱动程序的更多信息，包括配置详细信息以及可用的 API，请参见 *Framework Help （框架帮助）* > *Driver Libraries Help （驱动程序库帮助）* > *Decoder Libraries Help （解码器库帮助）* > AK4642 Codec Driver Library （AK4642 编解码器驱动程序库）部分。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `audio_microphone_loopback.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/audio/audio_microphone_loopback`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`audio_microphone_loopback.X`	`<install-dir>/apps/audio/audio_microphone_loopback/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
bt_audio_dk_16bit	bt_audio_dk	该演示在 PIC32 蓝牙音频开发工具包和音频编解码器子板 AK4642EN 上安装的 PIC32MX450F256L PIM 上运行。默认配置为使用 16 位数据宽度、 48000 Hz 采样频率和 I2S 音频协议。 在 MHC 中，可以在 I2S 驱动程序中更改音频协议，编解码器驱动程序会自动配置为对应于I2S协议集的适当音频数据格式。 可以将数据宽度更改为 24 位，这是编解码器支持的。应用程序用户应选择对应于 24 位宽度的 SPI 通信宽度设置和 I2S 音频数据格式。指定 I2S 格式之后，编解码器驱动程序会自动将自身配置为使用正确的数据格式。
pic32mx270f512l_pim_bt_audio_dk	pic32mx270f512l_pim+bt_audio_dk	该演示在 PIC32 蓝牙音频开发工具包和音频编解码器子板 AK4642EN 上安装的 PIC32MX270F512L PIM 上运行。默认配置为使用 16 位数据宽度、 48000 Hz 采样频率和 I2S 音频协议。 在 MHC 中，可以在 I2S 驱动程序中更改音频协议，编解码器驱动程序会自动配置为对应于I2S协议集的适当音频数据格式。 可以将数据宽度更改为 24 位，这是编解码器支持的。应用程序用户应选择对应于 24 位宽度的 SPI 通信宽度设置和 I2S 音频数据格式。指定 I2S 格式之后，编解码器驱动程序会自动将自身配置为使用正确的数据格式。

配置硬件

本部分介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包与用于蓝牙音频开发工具包的 PIC32MX270F512L PIM 以及音频编解码器子板 AK4642EN （见**注**）

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙音频开发工具包与 PIC32MX450/470F512L 接插模块（Plug-in Module，PIM）以及音频编解码器子板 AK4642EN（见**注**）

不需要硬件相关配置或跳线设置更改。

注： PIC32 蓝牙音频开发工具包包含一个音频 DAC 子板；但是，必须将该音频 DAC 子板更换为音频编解码器子板 AK4642。

运行演示

本部分说明如何运行演示。

说明

在使用该演示之前，建议先查看发行说明来了解任何已知问题。

重要提示！

执行以下操作来运行演示：

1. 编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。
2. 将扬声器或耳机连接到音频编解码器子板 AK4642EN 上的 line-out 线连接器。
3. 板载麦克风会开始捕捉周围的音频，并开始通过编解码器将它传送到微处理器，然后回送到编解码器耳机，您可通过收听耳机中的声音来观察麦克风输入。

audio_tone

本部分提供关于 MPLAB Harmony 编解码器驱动程序演示应用程序 （包含在 MPLAB Harmony 库分发包中）的说明和信息。

说明

在该演示应用程序中，使用编解码器驱动程序来设置开发板上的编解码器。该演示会发送出所生成的音频波形（正弦波音和 chirp 波），其参数可通过图形用户界面 （Graphic User Interface， GUI）基于 MPLAB Harmony 图形库图形和板载按钮来修改。根据对应于所显示参数的声音输出来指示是否成功。

要进一步了解 MPLAB Harmony 编解码器驱动程序、配置编解码器驱动程序和编解码器驱动程序提供的 API，请参见编解码器驱动程序库。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `audio_tone.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/audio/audio_tone`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`audio_tone.X`	`<install-dir>/apps/audio/audio_tone/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
bt_audio_dk_16bit_48000_RJ	bt_audio_dk	该演示在 PIC32 蓝牙音频开发工具包和音频 DAC 子板上的 PIC32MX470F512L 器件上运行。其配置为使用 16 位数据宽度、 48000 Hz 采样频率和右对齐音频协议来产生正弦波音或 chirp 波信号。
bt_audio_dk_24bit_44100_I2S	bt_audio_dk	该演示在 PIC32 蓝牙音频开发工具包和音频 DAC 子板上的 PIC32MX470F512L 器件上运行。其配置为使用 24 位数据宽度、 44100 Hz 采样频率和右对齐音频协议来产生正弦波音或 chirp 波信号。

配置硬件

本部分介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包和 PIC32 音频 DAC 子板

不需要硬件相关配置或跳线设置更改。

运行演示

本部分说明如何运行演示。

说明

可以生成连续正弦波音以及有限长度正弦波音和 chirp 波。表 2 总结了可以用于控制音频输出波形特性的按钮操作。图 2 说明了与按钮控件进行交互的显示功能的位置，以及它如何反映音频输出的当前状态。

编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。

1. 将扬声器或耳机连接到音频 DAC/ 编解码器子板的 line-out 线连接器 （见图 1）。
2. 在开始时，可以听到连续的 440 Hz 正弦音，数字仪表则通过显示 “f1 Hz” 来指示。
3. SW4 用于打开或关闭音频，这通过声音输出和显示 “Sine Tone - OFF”（正弦音调 - 关）和 “Sine Tone - ON”（正弦音调 - 开）来指示
4. 通过按下 SW1 来选择可修改的参数，这会循环选择显示为 “f1 (Hz)”、“f2 (Hz)” 和 “t (ms)” 的参数。开始时选择的是 “f1 (Hz)” 参数。可以使用 SW5/SW6 来更改任何选定参数的值，它们分别用于增大和减小参数值。“f2 (Hz)” 参数是 chirp 波最终频率。“t (ms)” 参数是信号的持续时间 （以毫秒为单位）。

注： 长按 SW1 （5 秒）会将选定的参数值设置为其最小允许值。长按 SW2 将设置为最大值。

5. 正弦音或 chirp 波模式通过按下 SW2 来选择。选择新模式时，输出会自动关闭。按下 SW4 会基于参数的当前设置启动声音输出。

注：

1. 当 “t (ms)” 参数显示为 99999 时，对于任一模式，都会以 “f1 (Hz)” 频率产生连续正弦音。当按下 SW3 来递增值时，会发出有限长度的 chirp 波或正弦音，从 0 ms 开始，每次递增 / 递减对应的步阶为 10 ms。
2. 长按 SW5/SW3 会加快选定值的递增 / 递减速度。

6. 所显示的采样频率可以通过探测音频 DAC/ 编解码器子板 （PIC32 蓝牙音频开发工具包配置）的 “LRCK 引脚点” 进行验证，如图 1 所示。
7. 连续正弦音输出的信号频率可以通过探测音频 DAC/ 编解码器子板的 “Line Out 点” 进行验证，如图 1 所示。频率应与 “f1 (Hz)” 显示值相同。有限长度正弦音和 chirp 波参数也可以通过使用存储示波器探测该点来进行验证。

图 1：PIC32 蓝牙音频开发工具包上的音频 DAC 子板

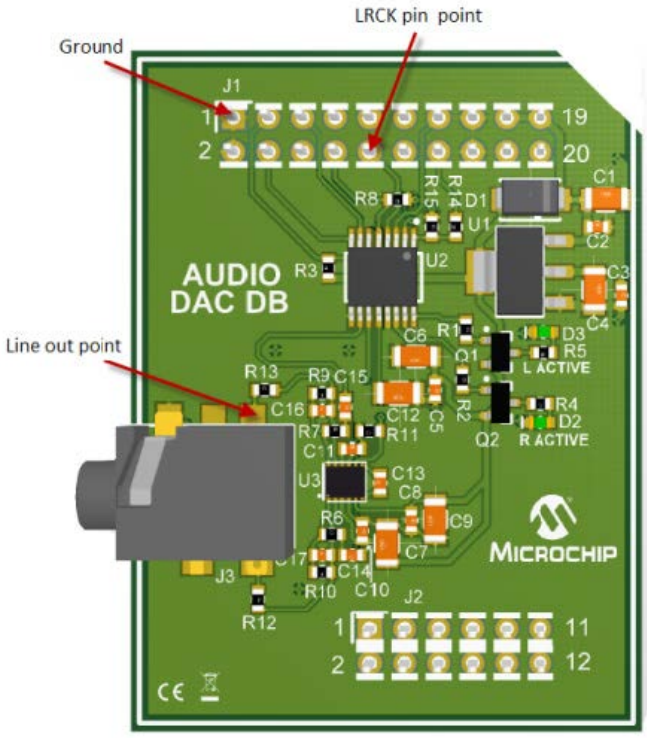
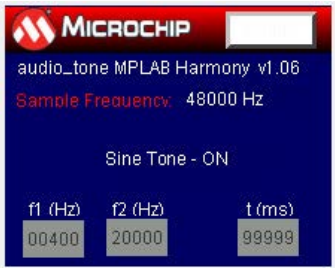


图 2：音调图形显示



控件说明

下表介绍了在演示支持的配置中使用的控件。

表 1：LED 控件

控件	bt_audio_dk
LED1	红色 LED D5，位于电路板的最上方。
LED2	红色 LED D6，位于电路板的最上方。
LED3	红色 LED D7，位于电路板的最上方。
LED4	红色 LED D8，位于电路板的最上方。
LED5	红色 LED D9，位于电路板的最上方。

表 2：按钮控件

控件	说明
SW1	f1/f2/t 参数选择 （长按 —— 参数最小值）。
SW2	chirp 波 / 正弦波模式选择 （长按 —— 参数最大值）。
SW3/SW5	参数递增 / 递减 （长按 —— 加快递增 / 递减速度）。
SW4	音频开 / 关 **注：**播放有限长度的波形，然后关闭。

mac_audio_hi_res

演示仿真 USB 扬声器的 USB 音频 2.0 设备。

说明

该演示应用程序使用 USB 音频 2.0 设备类来实现扬声器。该演示属于 Beta 版本，将在未来的 MPLAB Harmony 发行版中更新。

注： 该演示只能用于 Apple Mac 个人计算机，如带有 iTunes 11.2.1 或 VLC 媒体播放器 V2.2.1 的 Apple MacBook Air OS X 10.9.4。任何低于所列版本的设备都可能无法使用该演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了高分辨率音频演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `mac_audio_hi_res.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/audio/mac_audio_hi_res`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`mac_audio_hi_res.X`	`<install-dir>/apps/audio/mac_audio_hi_res/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `/firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_bt_audio_int_dyn	pic32mz_ec_pim+bt_audio_dk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙音频开发工具包上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MZ2048ECH144 音频 PIM。
pic32mz_ef_pim_bt_audio_dk	pic32mz_ef_pim+bt_audio_dk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙音频开发工具包上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MZ2048EFH144 音频 PIM。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包和 PIC32MZ2048ECH144 音频接插模块 （PIM）或 PIC32MZ2048EFH144 音频接插模块 （PIM）

1. 将 PIC32MZ 音频 PIM 插到 PIC32 蓝牙音频开发工具包上。
2. PIC32 蓝牙音频开发工具包上的开关 S1 应设置到 PIM_MCLR。
3. 将耳机连接到 PIC32 蓝牙音频开发工具包上的音频 DAC DB 的插孔中。

运行演示

介绍如何编译并运行高分辨率音频演示。

说明

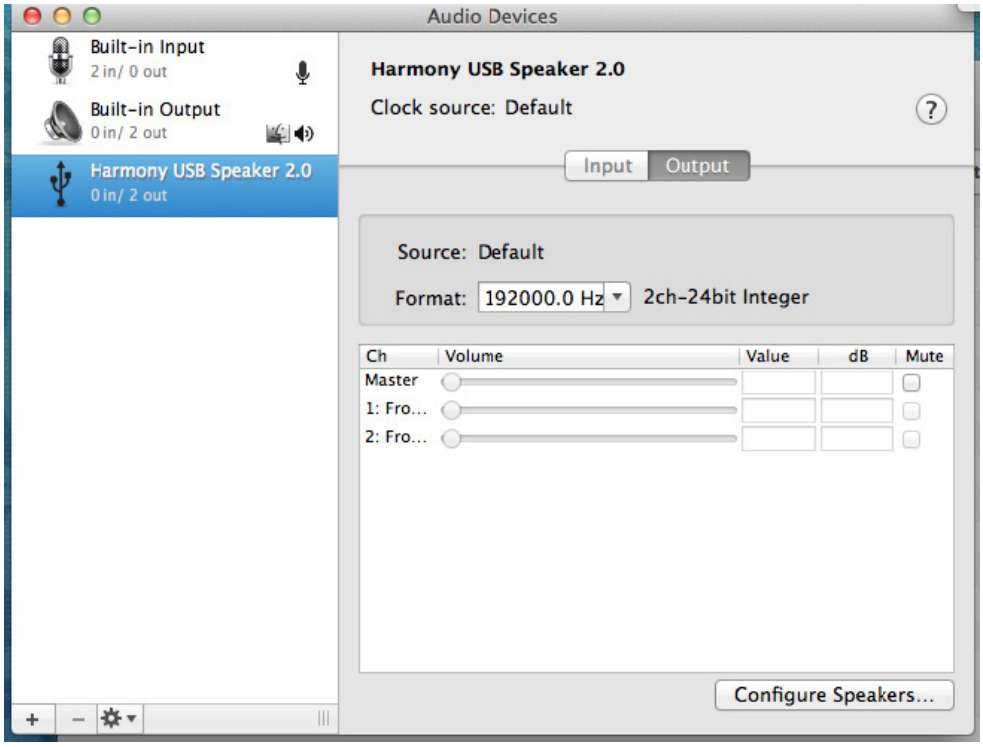
该演示在插入支持 USB 音频 2.0 设备的计算机时用作一个扬声器。

注：

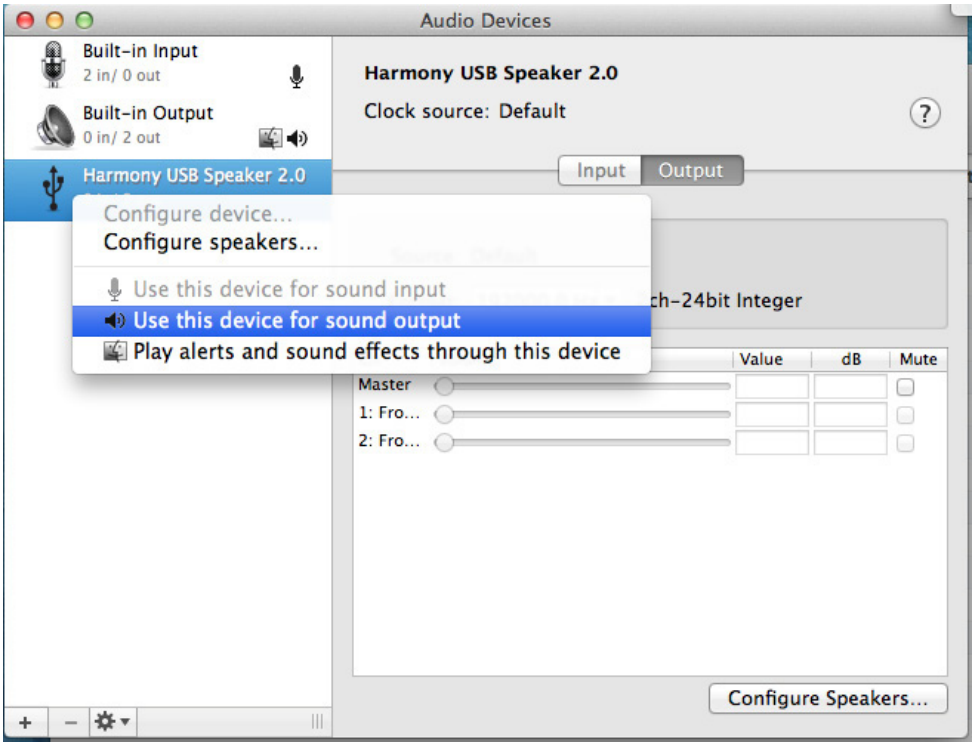
1. 在发布时，只有 Apple MAC 个人计算机本机支持音频 2.0 USB 设备。
2. 该演示已在 Windows® 7 上使用第三方音频 2.0 USB 设备驱动程序进行了测试。

执行以下操作来运行演示：

1. 编译演示应用程序并对器件编程。
2. 将设备与计算机 （例如 Apple Mac book）连接。
3. 使用将声音输出到扬声器的计算机功能。在带有 OS X 10.9.4 的 Apple Mac book 上，可以使用 Audio MIDI Setup （音频 MIDI 设置）应用程序，如下所示：
   - 打开 Audio MIDI Setup



   - 右键单击左侧列出的 **Harmony USB Speaker 2.0** （Harmony USB 扬声器 2.0）
   - 选择 **Use this device for sound output** （使用该设备作为声音输出）

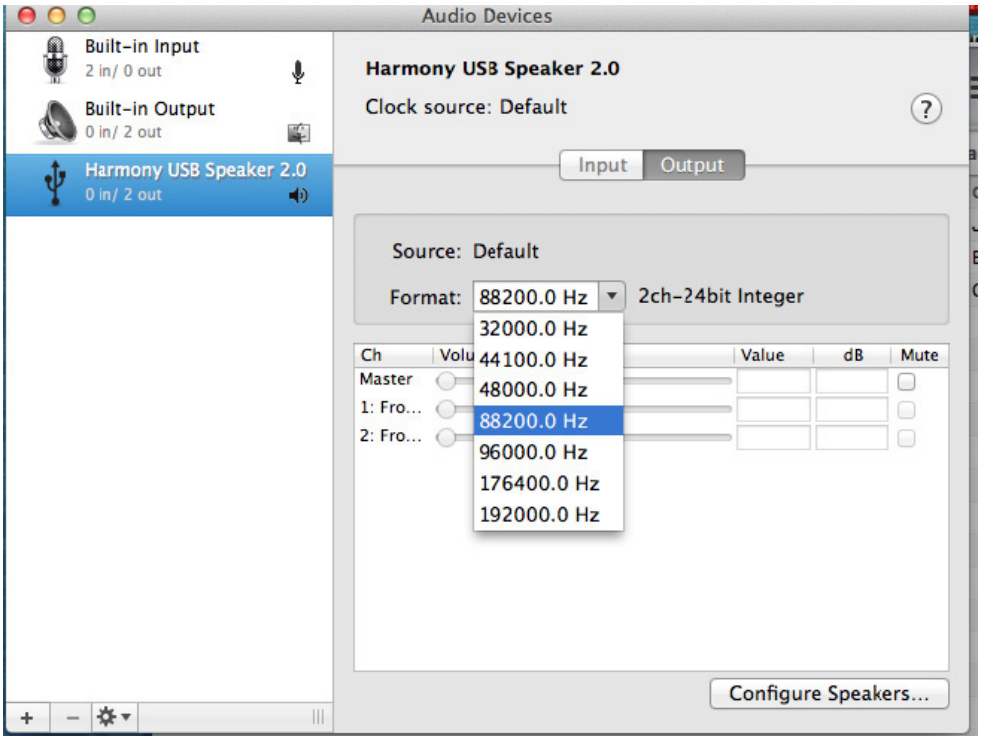


4. 打开音频播放器 （iTunes 或 VLC）并播放您选择的音频。
5. 功能单元只支持静音控制。正在播放的音频也可以使用开发板上的开关 SW4 来变为静音。
6. 音频音量可以通过计算机媒体播放器（Apple Mac book 上的 iTunes 或 VLC）控制，也可以通过开发板上的开关 SW1 和 SW2 控制。
7. 请注意，一些应用程序会在打开或关闭时锁定某声源（例如一些 Web 浏览器或插件）。因此，如果在查看网页或播放视频时插入扬声器，则要直到关闭并重新打开浏览器，声音才会被重定向到基于 USB 的扬声器。
8. 在该演示中创建的音频设备具有以下特性：
   - 支持的采样速率：
     - 32000 Hz
     - 44100 Hz
     - 48000 Hz
     - 88200 Hz
     - 96000 Hz
     - 176400 Hz
     - 192000 Hz
   - 2 通道 （立体声）
   - PCM 格式 （每个采样 24 位）

采样速率

该演示应用程序允许更改默认采样速率 （设置为 192000 Hz）。这可以在 Apple Mac book 上使用以下过程来完成。

1. 如果正在播放音频，则停止它 （在 iTunes 上选择 PAUSE）。
2. 打开 Audio MIDI Setup 应用程序。
3. 选择左侧列出的 Harmony USB Speaker 2.0。
4. 在右窗格中，从 *Format* （格式）下拉菜单中选择所需的采样速率，如下图所示。



5. 验证板上显示的采样速率已改变。
6. 在音频播放器上选择 “PLAY”（播放），使用更改后的采样速率播放音频。

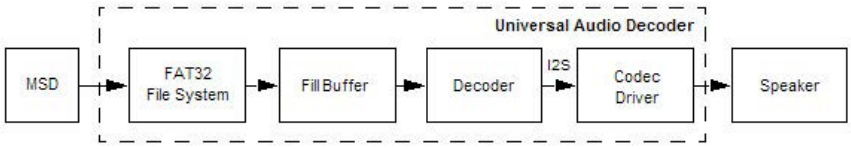
universal_audio_decoders

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

通用音频解码器应用程序会将开发板配置为处于 USB 主机模式。该应用程序支持 FAT32 文件系统。当海量存储设备连接到开发板时，应用程序会开始扫描根目录。在扫描完成后，将打开并播放列表中的第一首曲目。填充缓冲区程序会读取数据块，作为受支持文件格式的解码器的输入。解码器会对数据包解码，并将第一个音频数据帧通过 I2S 驱动程序发送给编解码器驱动程序，输出将可通过扬声器听到。以下框图给出了应用程序的图形表示形式。

按钮控件支持遍历目录树和播放来自其他目录或子目录的音频文件。默认情况下，应用程序仅支持 WAVE （.wav）格式的文件。



如果以插件形式添加了解码器库和受支持的源文件，则除了 WAVE 格式，应用程序还可以支持 MP3、AAC、WMA、ADPCM 和 Speex。MP3、 AAC 和 WMA 属于高级软件包，必须购买。详情请参见 Microchip 高级 MPLAB Harmony 音频网页（www.microchip.com/pic32harmonypremiumaudio）。

购买之后，可以按使用 MHC 选择解码器所述，将 MP3、 AAC 和 WMA 解码器模块添加到应用程序中。

如果任意解码器不受支持或已使用 MHC 删除，则不会扫描该特定文件格式。

音频格式	软件包	采样速率 （kHz）	说明
ADPCM	免费	8, 16	自适应增量脉冲编码调制 （Adaptive Delta Pulse Code Modulation， ADPCM）是 Microsoft 波形 （.wav）文件格式的一个子类。在该演示中，它只会对具有 WAV 文件头的 ADPCM 音频解码。该格式的扩展名为 pcm 或 PCM。
Speex	免费	8, 16	Speex 是针对语音设计的开源 / 免费软件的无专利保护音频压缩格式。在该演示中，只能对处于 Ogg 容器内的 Speex 位流解码。该格式的扩展名为 spx 或 SPX。
WAVE	免费	8 至 96	WAVE 文件格式是 Microsoft Windows 用于存储数字音频数据的本机文件格式。
AAC	高级 （必须购买）	8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 64, 88.2 和 96	AAC 格式是音频数据的一种有损数字压缩格式，带有 ADTS 文件头。 AAC 解码器支持 MPEG-2 和 MPEG-4 AAC。为确保 AAC 音频文件可用于 AAC 解码器，可以总是通过某种 MPEG-2 或 4 AAC 编码器将所有音频文件都转换为 MPEG-2 或 4 AAC 文件，一种已知可用的编码器是 FAAC （Freeware Advanced Audio Coder）。
MP3	高级 （必须购买）	32, 44.1 和 48	MPEG1 Layer 3 是音频数据的一种有损数字压缩格式。
WMA	高级 （必须购买）	8, 11.025, 16, 22.05, 32, 44.1 和 48	Windows 媒体应用程序（Windows Media Application， WMA）格式允许使用有损压缩算法存储数字音频。WMA解码器支持ASF容器格式。可以从Microsoft网站下载Windows Media Encoder 9 Series，用于将任意音频文件转换为 WMA v9.2 文件，从而可用于该 WMA 解码器。

注： AAC 和 MP3 解码器库具有两个版本：PIC32MX （用于 PIC32MX 器件）和 microAptiv （用于具有 microAptiv 内核的 PIC32MZ 器件）。对于 PIC32MX 器件，在 MHC 中选择 AAC 或 MP3 库时，会在项目中自动添加 PIC32MX 版本的库。对于具有 microAptiv 内核的 PIC32MZ 器件，会在项目中添加 microAptiv 版本的库。理论上来说，MHC 会自动添加正确的库；但是，务必确保在不同设备上使用正确的库。

每个配置下的 `system_config.h` 中的 DISK_MAX_DIRS 和 DISK_MAX_FILES 宏决定在每一级扫描的最大目录数，以及演示总共扫描的最大歌曲数量。对于每个配置，值可能不同。例如，对于 PIC32MX270F512L PIM， DISK_MAX_FILES 可以大约为 600，而对于 PIC32MZ EF 入门工具包， DISK_MAX_FILES 最大可以为 800。

关于解码器库的更多信息，请参见库框架帮助中的 AAC 解码器库、 MP3 解码器库和 WMA 解码器库，以及 Microchip 高级 MPLAB Harmony 音频网页 （www.microchip.com/pic32harmonypremiumaudio）。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `universal_audio_decoders.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/audio/universal_audio_decoders`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`universal_audio_decoders.X`	`<install-dir>/apps/audio/universal_audio_decoders/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
ak7755_bt_audio_dk	bt_audio_dk	该配置在 PIC32 蓝牙音频开发工具包和 AK7755 编解码器子板上运行。
bt_audio_dk	bt_audio_dk	该配置在 PIC32 蓝牙音频开发工具包上运行。
270f512lpim_bt_audio_dk	pic32mx270f512l_pim+bt_audio_dk	该配置在 PIC32MX270F512L PIM 和 PIC32 蓝牙音频开发工具包上运行。
pic32mz_ec_pim_bt_audio_dk	pic32mz_ec_pim+bt_audio_dk	该配置在 PIC32MZ 音频 PIM 和 PIC32 蓝牙音频开发工具包上运行。
pic32mz_da_sk_meb2	pic32mz_da_sk+meb2+wvga	该配置在PIC32MZ DA入门工具包和多媒体扩展板II上运行。

使用 MHC 选择解码器

本主题介绍如何使用 MHC 选择解码器。

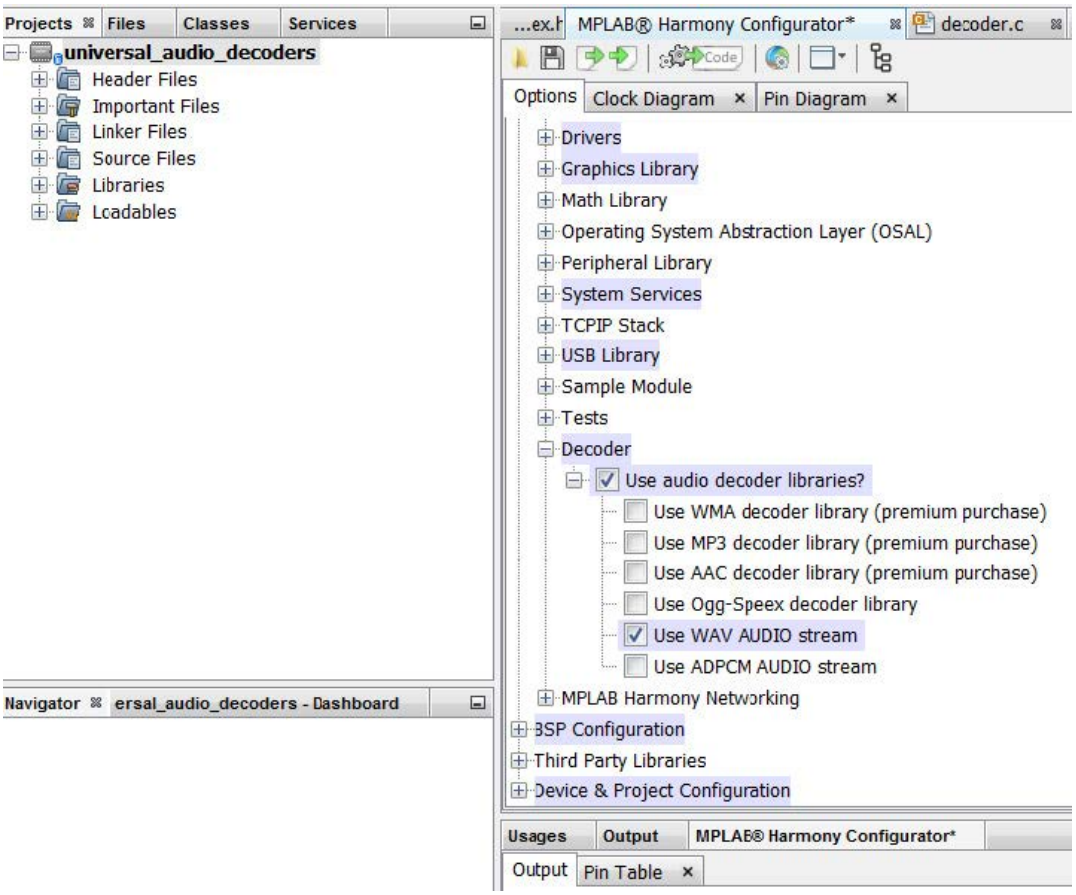
说明

该应用程序支持 WAVE、压缩 MP3、AAC（ADTS 文件头）、WMA（v9.2）、ADPCM（WAV 文件头）和 Speex（Ogg 容器）音频格式。

注： MP3、 AAC 和 WMA 解码器库属于高级软件包，必须购买。详情请参见 Microchip 高级 MPLAB Harmony 音频网页（www.microchip.com/pic32harmonypremiumaudio）。

音频格式可以根据需要使用 MHC 选择，如下所示：

1. 在 MPLAB X IDE 中打开项目，并根据硬件选择项目配置。
2. 打开 MHC 并选择解码器，如下图所示。



3. 选择 **Use audio decoders libraries?** （使用音频解码器库？），下拉菜单中会列出可用解码器的列表。
4. 选择所需的解码器，并单击 **Generate** （生成）将支持的解码器文件和库添加到项目中。

配置硬件

本部分介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙音频开发工具包和 PIC32MZ 音频 PIM 或 PIC32MX270F512L 接插模块 （PIM）

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包和 MEB II 以及 5" WVGA PCAP 显示板 （见**注**）

不需要硬件相关配置或跳线设置更改。

注： 关于获取 5" WVGA PCAP 显示板的信息，请与您当地的 Microchip 销售办事处联系。

运行演示

本部分说明如何运行演示。

说明

注： 在运行该演示之前，建议先查看发行说明中的音频演示部分来了解任何已知问题。

编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。这些配置会设置目标处理器、电路板和要在输出接口中使用的编解码器。

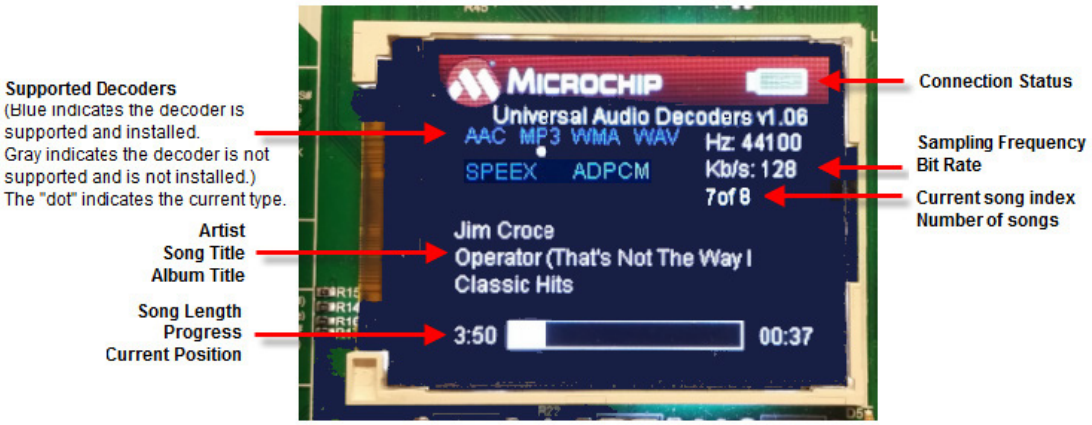
详情请参见编译应用程序。

执行以下操作来运行演示：

1. 将扬声器或耳机连接到 PIC32 音频 DAC 子板 （PIC32 蓝牙音频开发工具包随附）上的 line-out 线连接器或位于目标硬件上的音频连接器。
2. 将电源连接到电路板。系统将处于等待 USB 连接的状态。在处于等待状态期间， LED1、 LED2 和 LED3 将点亮。



3. 连接包含受支持音频格式的歌曲的 USB 海量存储设备。默认情况下，该应用程序可以流式传输 WAVE （.wav）格式的音频文件。
4. 当连接 USB 设备后，系统将扫描音频文件。在扫描期间， LED1 和 LED3 将点亮。
5. 扫描完成时，可以从连接到电路板的扬声器或耳麦听到音频输出。对于 WAVE 音频，LED1 将点亮；对于 AAC，LED2 将点亮；对于 MP3， LED3 将点亮。



LED 状态

状态	LED1	LED2	LED3
等待 USB	点亮	点亮	点亮
扫描文件	点亮	熄灭	点亮
WAVE 音频流	点亮	熄灭	熄灭
AAC 音频流	熄灭	点亮	熄灭
MP3 音频流	熄灭	熄灭	点亮

WMA 音频流	熄灭	点亮	点亮

演示控件

元件	标签	PIC32 蓝牙音频开发工具包	PIC32MZ EC 入门工具包和 PIC32MZ EF 入门工具包
开关	SW1	N/A	下一曲目 （切换） / 快进 （长按）
开关	SW2	N/A	播放 / 暂停 （切换）
开关	SW3	下一曲目 （切换） / 快进 （长按）	上一曲目 （切换） / 快退 （长按）
开关	SW4	播放 / 暂停 （切换）	N/A
开关	SW5	上一曲目 （切换） / 快退 （长按）	N/A

usb_headset

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

在该演示应用程序中，系统被配置为 USB 耳麦。系统将与 USB 主机（如个人计算机）连接，后者支持 USB 设备类耳麦。嵌入式系统将使用 USB 音频设备端点进行枚举，使系统能够从 USB 端口输出音频。它使用符合标准 USB 1.0 的全速实现。

嵌入式系统将通过音频 USB 接口获取数据，并对其设置格式，从而使用编解码器驱动程序进行输出。编解码器驱动程序会设置音频输出接口、时序、 DMA 通道和缓冲区，从而能够输出连续的音频流。数字音频通过 I2S 数据通道以 48 kHz （典型值）进行处理。

然后，输出音频流通过硬件连接 （通常连接到耳机或扬声器）以模拟格式提供。

该系统支持多种可应对不同硬件平台和编解码器的配置。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `usb_headset.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/audio/usb_headset`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`usb_headset.X`	`<install-dir>/apps/audio/usb_headset/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
bt_audio_dk	bt_audio_dk	该配置在 PIC32 蓝牙音频开发工具包上运行。
bt_audio_dk_ak4642	bt_audio_dk	该配置在 PIC32 蓝牙音频开发工具包和音频编解码器子板 AK4642EN 上运行。

pic32mx270f512l_pim_bt_audio_dk	pic32mx270f512l_pim+bt_audio_dk	该配置在使用 PIC32MX270F512L PIM 配置的 PIC32 蓝牙音频开发工具包上运行。

配置硬件

本部分介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包。

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙音频开发工具包和音频编解码器子板 AK4642EN。

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙音频开发工具包和 PIC32MZ270F512L PIM。

不需要硬件相关配置或跳线设置更改。

运行演示

本部分说明如何运行演示。

说明

重要提示！ 在使用该演示之前，建议先查看发行说明中的 PIC32 蓝牙音频协议栈演示部分来了解任何已知问题。

编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。这些配置会设置目标处理器、电路板和要在输出接口中使用的编解码器。详情请参见编译应用程序。

执行以下操作来运行演示：

1. 将扬声器或耳机连接到音频 DAC 子板 AK4642EN 上的 line-out 线连接器或位于目标硬件上的音频连接器。
2. 将电源连接到电路板。
3. 通过标准 USB 电缆将使用 USB mini-B 连接器 （设备）的电路板连接到主机计算机。
4. 允许主机计算机应答、安装驱动程序 （如需要）和枚举设备。主机端不需要任何特殊软件。
5. 如果需要，将主机计算机配置为使用 usb_speaker 输出作为选定的音频设备。这可以在系统配置或“控制面板”中完成，具体取决于操作系统。
6. 在主机计算机上播放音频。这可以使用标准媒体播放器完成，也可以借助各种来源 （包括操作系统生成的声音或视频）完成。
7. 可以从连接到电路板的扬声器或耳麦听到音频输出。音量通常由主机调节。

蓝牙演示

本主题介绍 PIC32 Bluetooth® 协议栈库演示。

简介

PIC32 蓝牙协议栈库演示应用程序帮助。

说明

MPLAB Harmony 标准安装包含了一系列与蓝牙相关的固件项目，用于演示 MPLAB Harmony PIC32 蓝牙协议栈库的功能。该库被视为“基本的”蓝牙协议栈，包含称为“数据演示”的基本演示。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些基本蓝牙协议栈固件项目的过程。

除了随 PIC32 蓝牙基本协议栈库提供的数据演示之外，其他“高级演示”（可购买）用于演示 PIC32 蓝牙音频协议栈库的功能。高级演示部分提供了关于如何获取、配置和运行这些演示的信息。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference（卷 IV：MPLAB Harmony 框架参考）中的相关文档。

演示

本部分介绍标准数据演示，以及需购买的高级演示。

演示功能

介绍演示的功能。

说明

基本功能

蓝牙模块

PIC32 蓝牙入门工具包和 PIC32 蓝牙音频开发工具包提供了对于 BlueCore® CSR8811™ 的硬件支持。

CSR8811

CSR8811 是一款用于蓝牙 2.4 GHz 系统的单芯片无线电和基带 IC，包含了最高 3 Mbps 的增强型数据速率（Enhanced Data Rate，EDR）和蓝牙低功耗特性。CSR8811 支持蓝牙 1 类传输，并支持多器件连接。PIC32 蓝牙入门工具包和 PIC32 蓝牙音频开发工具包在其默认配置中使用一个基于 CSR8811 无线电的模块（见**注**）。

注： 使用 CSR8811 器件的 Flairmicro BTM805 模块集成在 PIC32 蓝牙入门工具包中，也集成在 PIC32 蓝牙音频开发工具包上安装的 BTM805 蓝牙子板中。

蓝牙器件 ID

蓝牙软件能够记住并在闪存中存储它成功配对过的最后 10 个唯一蓝牙器件 ID，以便在当前没有活动蓝牙连接时可以更快地自动重新连接。如果当前连接的用户智能手机关闭了蓝牙并之后重新使能，如果它处于通信范围内或当它回到通信范围内时，将自动重新连接。

注： 目前，由于MPLAB Harmony中的限制，演示不支持SPI闪存，因此在发生电源或硬件复位时不会存储或恢复配对信息。

蓝牙器件地址

当开发工具包上电时，它会为所有给定的开发工具包硬件生成随机的唯一蓝牙器件地址。或者，用户也可以在设计时，在开发工具包的应用程序代码中指定蓝牙器件地址。

器件地址是一个 6 字节的十六进制值。宏 BT_DEVICE_ID 定义十六进制值的前 4 个字节，BT_DEVICE_ID_2LSB 定义十六进制值的最后 2 个字节。通过使能BT_DEVICE_ID_2LSB_RANDOMIZE，可以随机生成器件地址的最后 2 个字节。这些宏在`btconfig.h`中定义。

建议不要在设计和开发期间设置特定的硬编码器件地址，因为如果有另一个具有相同蓝牙器件地址的开发板处于通信范围内，则可能会遇到蓝牙连接问题。

其他蓝牙资源

提供关于其他蓝牙演示资源的信息。

说明

除了演示部分中介绍的演示功能之外，还提供了其他蓝牙演示资源。

其他 Android 应用程序

此外，也可以使用 Bluetooth SPP-pro 应用程序来执行蓝牙数据智能手机演示。该应用程序可以从以下网址下载：

https://play.google.com/store/apps/details?id=mobi.dzs.android.BLE_SPP_PRO&hl=en

安装之后，可以在该应用程序的终端仿真器的命令行模式 / 键盘模式下发送和接收命令。命令将使用下表所示的格式发送和接收。

Windows 手机应用程序

对于 Windows 手机，可以通过数据终端发送和接收命令。该应用程序可以从以下网址下载：

http://www.windowsphone.com/en-us/store/app/bt-terminal/09d679af-bdd8-40b2-b54e-56d68aeb03e0

安装之后，可以从该应用程序的终端仿真器中发送和接收命令。命令可以使用下表所示的格式发送和接收。

蓝牙数据 Windows 个人计算机演示设置

使用十六进制文件 `data_temp_sens_rgb.X.production.hex` 对器件编程。

1. 在 Windows 个人计算机上，选择*开始* > *控制面板* > *硬件和声音* > *添加设备*。此时会显示可用蓝牙设备的列表。
2. 从列表中，选择 **BTSK**。
3. 打开一个终端仿真器。对于本示例，使用的是 PuTTY。该 Windows 应用程序可从以下网址获取：
   http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
4. 需要在 PuTTY 和 PIC32 蓝牙入门工具包开发板之间建立连接。这可以通过在 PuTTY 配置窗口中选择 “Serial”（串行）连接类型和分配给电路板的 COM 端口来完成。所分配的 COM 端口可以通过*设备管理器* > *端口*确定。
5. 连接建立之后，可以发送以下命令。

命令	说明
R	将 LED 设定为 100% 的红色
G	将 LED 设定为 100% 的绿色
B	将 LED 设定为 100% 的蓝色
r	将 LED 设定为 50% 的红色
g	将 LED 设定为 50% 的绿色
b	将 LED 设定为 50% 的蓝色

数据演示

本主题介绍如何运行在本 MPLAB Harmony 发行版中免费包含的 PIC32 蓝牙基本协议栈库 “数据演示”应用程序。

data_basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示通过蓝牙链路执行全双工数据传输。该演示使用户可以执行终端仿真，回显从 Android 智能手机上通过蓝牙连接发送到开发板的字符，然后再发送回到智能手机仿真应用程序屏幕。

从智能手机到开发板的数据传输通过用户发送 （通过从 Android 智能手机应用程序中输入字符）来演示。数据接收通过闪烁两个 LED 来指示：从熄灭变为点亮，然后恢复为熄灭状态。

从开发板到智能手机的数据传输通过用户发送 （通过按下开发板顶部的按钮）来演示。智能手机会将接收到的数据显示在其屏幕上，显示为 “Button 1”和 “Button 2”等。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了蓝牙演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `data_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/bluetooth/data_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`data_basic.X`	`<install-dir>/apps/bluetooth/data_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
bt_audio_dk	bt_audio_dk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙音频开发工具包上运行演示。
pic32mx270f512l_pim_bt_audio_dk	pic32mx270f512l_pim+bt_audio_dk	选择该 MPLAB X IDE 项目配置可以在使用 PIC32MX270F512L PIM 的 PIC32 蓝牙音频开发工具包上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙音频开发工具包和 PIC32MX270F512L 接插模块 （PIM）

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行蓝牙基本数据演示。

说明

该演示执行基本的 SPP 全双工数据传输。

注： 在运行演示之前，需要先安装 Bluetooth SPP-pro Android 应用程序。关于详细信息，请参见演示部分中的*其他蓝牙资源 > 其他 Android 应用程序*。

运行演示

1. 编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。
2. 当 LED1 和 LED2 点亮时，表示蓝牙设备演示正在运行。
3. 在您的智能手机上使能蓝牙。
4. 在您的智能手机上启动 Android 应用程序。
5. 扫描可用的蓝牙设备。目标蓝牙设备也应显示在您的智能手机的可用设备列表中。
6. 目标蓝牙设备的名称将为以下之一：

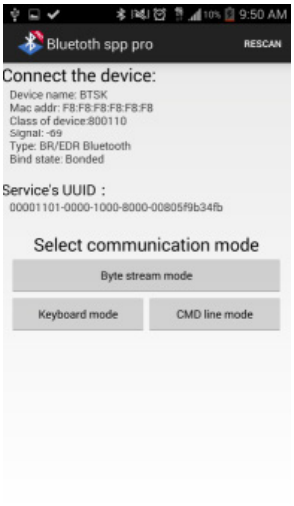
配置	器件名称
bt_audio_dk	BTAD
pic32mx270f512l_pim_bt_audio_dk	BTAD

注： 有时，蓝牙设备的名称尚未解析，将显示为“null”。一段时间后名称会从“null”更改为前面列出的特定于配置的名称。可见 MAC 地址的前 8 位是固定的，后 4 位可变 （12:34:45:78:XX: XX）。

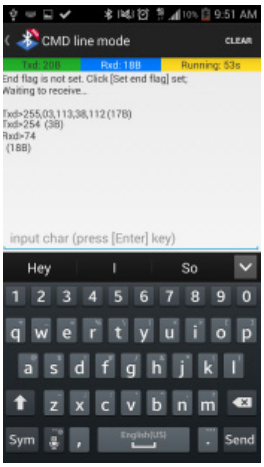
7. 选择要进行配对的器件并连接。



8. 如果连接成功，屏幕顶部会显示消息 “connected to <Device Name>”（已连接到 <设备名称>）。
9. 选择 “CMD Line Mode”（命令行模式）选项卡，输入字符并按下 “Send”（发送）按钮。蓝牙设备接收到字符时，会通过 LED 指示，即 LED4 和 LED5 会从熄灭切换为点亮，然后恢复为熄灭。每次接收到数据时，蓝牙设备都会重复该序列。



10.此外，也可以从蓝牙设备向连接的智能手机发送数据。这可以通过按下开发板上的开关 SW1 至 SW5 来完成。按下开关时，两个 LED（LED4 和 LED5）会从熄灭切换为点亮，然后恢复为熄灭。智能手机会接收数据，并对于 SW1/S1 开关显示 “Button 1”，对于 SW2 开关显示“Button 2”，对于 SW3 开关显示“Button 3”，对于 SW4 开关显示“Button 4”，对于 SW5 开关显示“Button 5”。

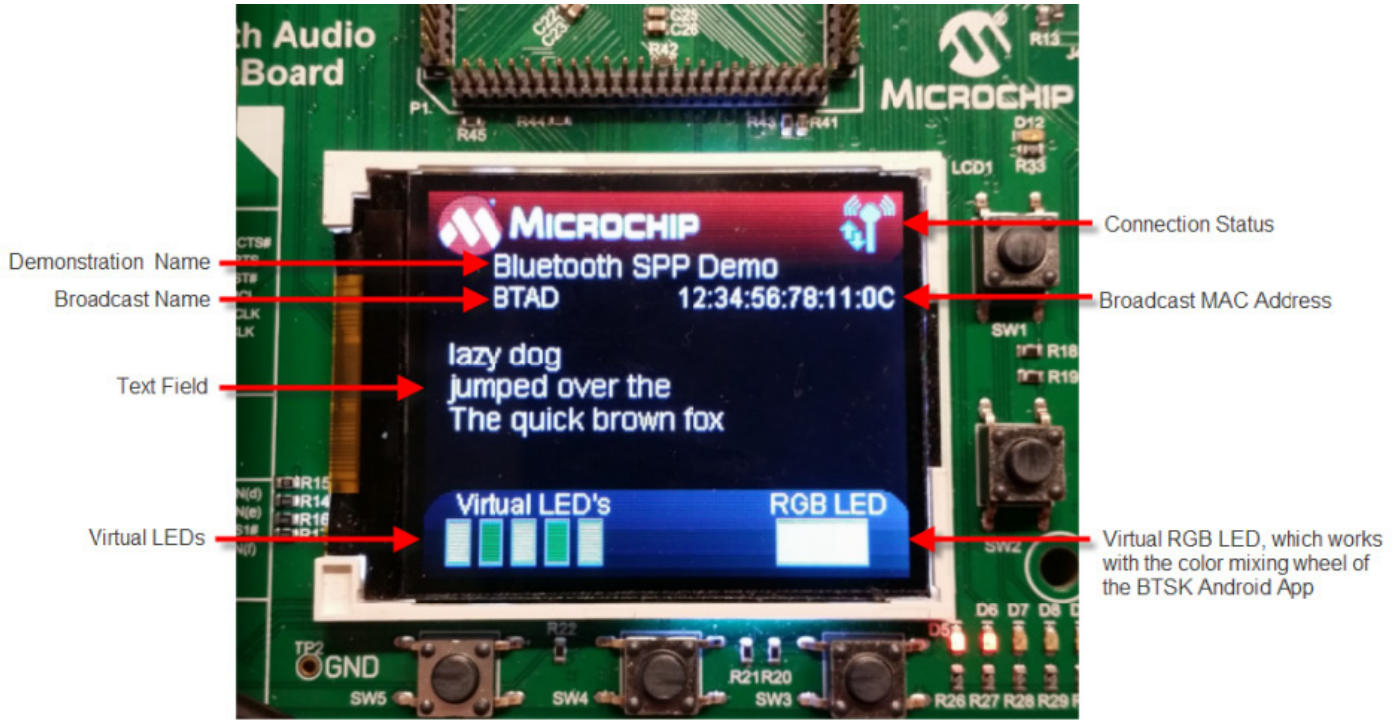


下表介绍了在演示支持的配置中使用的控件。

控件	bt_audio_dk
LED1	PIC32 蓝牙音频开发工具包开发板上的红色 LED D5。
LED2	PIC32 蓝牙音频开发工具包开发板上的红色 LED D6。
LED3	PIC32 蓝牙音频开发工具包开发板上的红色 LED D7。
LED4	PIC32 蓝牙音频开发工具包开发板上的红色 LED D8。
LED5	PIC32 蓝牙音频开发工具包开发板上的红色 LED D9。
SW1/S1	PIC32 蓝牙音频开发工具包开发板上的 SW1 开关。
SW2	PIC32 蓝牙音频开发工具包开发板上的 SW2 开关。
SW3	PIC32 蓝牙音频开发工具包开发板上的 SW3 开关。
SW4	PIC32 蓝牙音频开发工具包开发板上的 SW4 开关。
SW5	PIC32 蓝牙音频开发工具包开发板上的 SW5 开关。

演示显示

下图说明了运行演示时的显示格式。



连接状态

显示屏上的连接状态图标的颜色指示演示的蓝牙状态，如下表所述。

图标颜色	显示示例	说明
灰色		蓝牙未配对且未连接。
白色		蓝牙已配对但未连接。
蓝色		蓝牙已配对且已连接。

演示命令

下表列出了控制演示不同方面的命令。请注意，所有命令都区分大小写，并且在默认情况下，不会在 Text Field （文本字段）中显示演示可识别的命令。要查看发送的所有文本，必须先发送 “DISPLAY_ALL”或 “DAI”命令。

通过蓝牙发送的命令

命令	快捷命令	操作	示例
DISPLAY_ALL	DAI	显示所有文本。	DISPLAY_ALL
DISPLAY_ALL_STOP	DAS	停止显示所有消息，仅显示无法识别的命令 （默认）。	DISPLAY_ALL_STOP
ledx on （x = 1-5）		点亮指定的虚拟 LED （LED1 = 左， LED5 = 右）。	led3on
ledx off （x = 1-5）		熄灭指定的虚拟 LED （LED1 = 左， LED5 = 右）。	led3off
ledx toggle （x = 1-5）		切换指定虚拟 LED 的状态 （LED1 = 左， LED5 = 右）。	led3toggle
Lx （x = 1-31）		使用虚拟 LED 显示二进制模式 （MSB = 左， LSB = 右）。	L21
255,03,R,G,B		将虚拟 RGB LED 的颜色修改为指定的 RGB 值，范围分别为 0-254。	255,03,127,127,127

data_temp_sens_rgb

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示通过蓝牙链路执行全双工数据传输。该演示使用户可以使用 Microchip 的 PIC32 蓝牙入门工具包应用程序通过基于 Android 操作系统的智能手机发送和接收 SPP 数据。 PIC32 蓝牙入门工具包 Android 应用程序使用户可以通过基于 Android 操作系统的智能手机向 Microchip 的 PIC32 蓝牙入门工具包发送和接收 SPP 数据。更多信息，请参见 *“PIC32 Bluetooth Starter Kit User's Guide ”*（DS70005190）。应用程序文件 BTSK_Android_App.apk 位于 MPLAB Harmony 安装的 `<install-dir>/apps/bluetooth/data_temp_sens_rgb/android_app` 文件夹。

从 Android 手机到 PIC32 蓝牙入门工具包的数据传输通过用户发送红色、绿色和蓝色数据来演示，即用户需要通过操作 Android 应用程序的直观 GUI 上的 “COLOR”（颜色）选项卡来生成颜色。接收到所生成的颜色后，将通过 PIC32 蓝牙入门工具包上的 Cree 高输出多色 LED 来显示。

从 PIC32 蓝牙入门工具包到 Android 手机的数据传输通过用户请求温度测量值来演示，即用户需要操作 Android 应用程序的直观 GUI 上的 “TEMPERATURE”（温度）选项卡。当 PIC32 蓝牙入门工具包应用程序接收到请求时，会通过读取入门工具包上的温度传感器来发送最新值。 Android 应用程序会接收温度值，并在动态更新的 “TEMPERATURE”选项卡上显示它们。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了蓝牙演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `data_temp_sens_rgb.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/bluetooth/data_temp_sens_rgb`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`data_temp_sens_rgb.X`	`<install-dir>/apps/bluetooth/data_temp_sens_rgb/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_bt_sk	pic32mx_bt_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙入门工具包上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 蓝牙入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行蓝牙温度传感器和 RGB 数据演示。

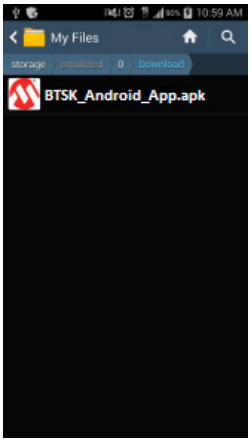
说明

该演示可以执行温度传感器和 RGB 数据的 SPP 数据传输 / 接收。

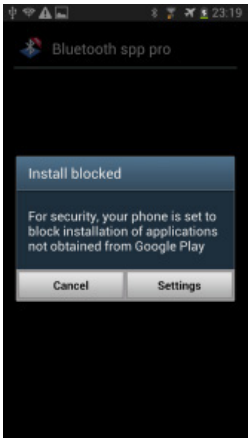
注： 在运行演示之前，必须先在 Android v4.0 或更高版本智能手机上安装 PIC32 蓝牙入门工具包 Android 应用程序。

安装 PIC32 蓝牙入门工具包 Android 应用程序

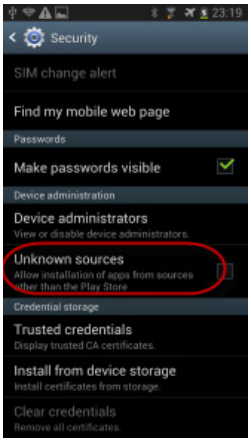
1. 将 Android 应用程序 BTSK_Android_App.apk 安装到 Android 4.0 或更高版本智能手机中。该文件位于以下 MPLAB Harmony 安装文件夹：`<install-dir>/apps/bluetooth/data_temp_sens_rgb/android_app`。
2. 使用 mini-B USB 连接器将 Android 设备连接到计算机。
3. 建议将 Android 应用程序复制到 Android 设备的 `Download` 文件夹。
4. 在 Android 设备上，选择 *My Files （我的文件）>All Files （所有文件）>Download （下载）>BTSK_Android_App.apk*。



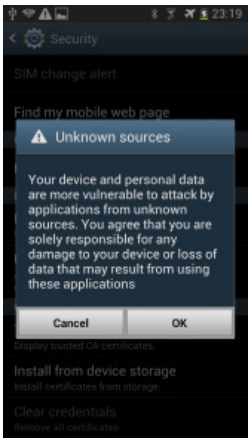
5. 选择 .apk 文件之后，会显示警告消息 “blocking installation”（阻止安装）。



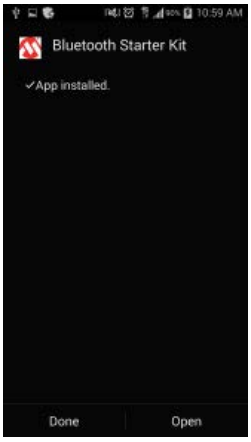
6. 选择 **Settings** （设置），此时会显示安全窗口。
7. 选择安装来自未知源的应用程序的选项。



8. 选择该选项后，会显示一个警告，如下图所示。

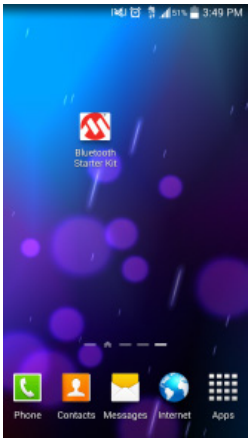


9. 选择 **OK** （确定）后，会出现一个请求确认安装应用程序的窗口。
10. 安装完成后，选择 **Open** （打开）来运行该应用程序，如下图所示。

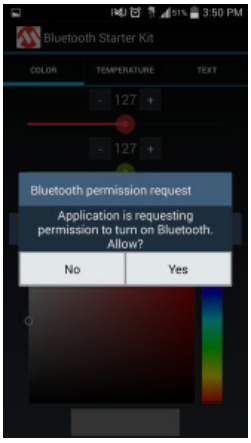


运行温度传感器和 RGB 演示

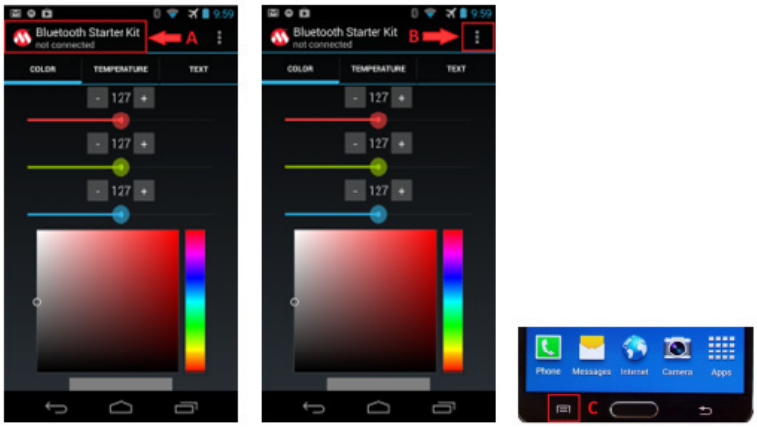
1. 编译并使用十六进制文件 `data_temp_sens_rgb.X.production.hex` 对目标器件编程。
2. 选择 Android 设备上的 PIC32 蓝牙入门工具包，此时会显示一个窗口，如下图所示。



3. 在应用程序窗口中选择 Bluetooth Starter Kit （蓝牙入门工具包）图标。
4. 如果出现提示，则通过选择 **Yes** （是）开启蓝牙。

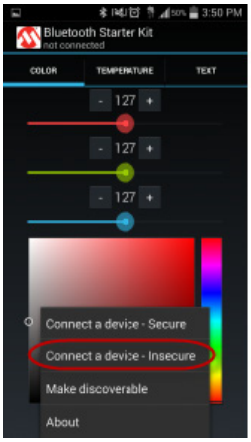


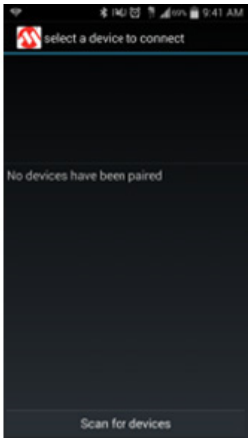
5. 根据您所使用的手机和 Android 版本，可以通过 3 种方法来执行接下来的步骤。 **[A]** 按下 Microchip 徽标和文字 “Bluetooth Starter Kit”， [B] 按下屏幕菜单按钮 **[B]**（如果支持），或 **[C]** 按下菜单按钮（硬件）。通常，如果手机具有硬件按钮，则不会有屏幕按钮，反之亦然。打开菜单后，选择 **Connect a device - Insecure** （连接器件 - 不安全）。



Method [A] Method [B] Method [C]

6. 选择 “Connect a device - Insecure”选项，此时会出现一个窗口，显示已配对设备的列表，以及扫描设备的选项，如下图所示。



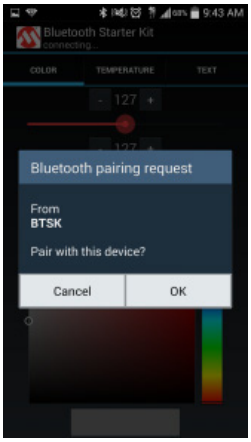


7. 此时会显示已配对设备的列表。如果这是第一次连接 PIC32 蓝牙入门工具包，则请选择 **Scan for Devices**（扫描设备）。应注意，有时名称尚未解析，将显示为“null”。一段时间后名称会从“null”更改为“BTSK”。使用代码对器件编程时， MAC 地址的前 8 位是固定的，后 4 位可变（12:34:45:78: XX: XX）。

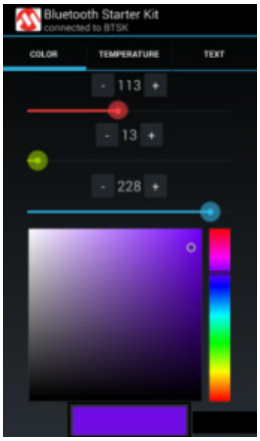




8. 当选中时， Android 应用程序会配对并连接 PIC32 蓝牙入门工具包。接受任何配对请求，如下所示。



9. 状态指示器将确认连接已成功，如下图所示。

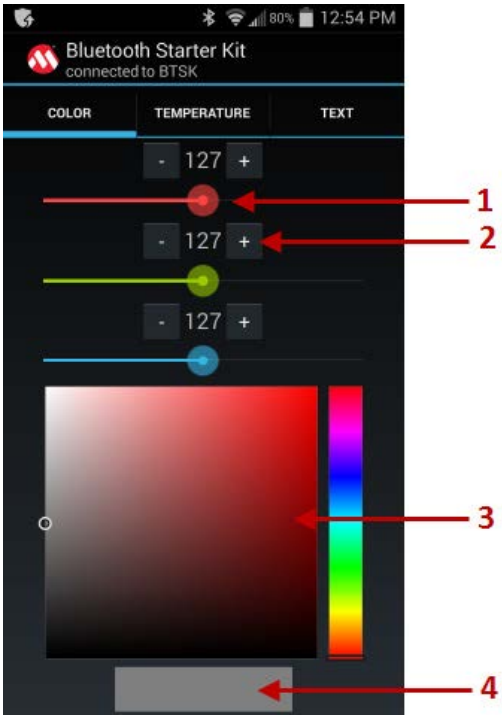


注： 如果应用程序无法连接，则检查 PIC32 蓝牙入门工具包是否已上电、处于通信范围内，且未连接到另一个蓝牙设备。

LED 颜色控件

该应用程序包含 3 个分别对应于红色 （R）、绿色 （G）和蓝色 （B）的颜色滑块。 LED 的颜色在修改滑块 **(1)** 时设定。此外，也可以使用递增按钮或递减按钮 **(2)** 来修改 LED 的颜色。类似地，可以通过基于调色板 **(3)** 上的色调和饱和度选择颜色来修改颜色。滑块和递增 / 递减按钮会通过 SPP 向 PIC32 蓝牙入门工具包发送命令，从而设定 LED 颜色。 R、 G 和 B 颜色的范围分别为 0 至 254。颜色条 **(4)** 会指示 RGB 组合产生的修改后的颜色。LED 会实时变化，以近似显示颜色条中的颜色。LED 颜色将具有集成的三个彩色二极管未经校准的特性。

该应用程序演示的是 Android 设备与开发板之间的全双工数据传输。每次修改颜色时，都会通过 SPP 以字符串格式向开发板发送命令。发送的命令可以在 Text （文本）菜单中查看。



温度传感器

在 Android 应用程序中选择 Temperature（温度）菜单时，将出现如下图所示的窗口。图中会显示华氏温度读数，它在接收到新的读数时自动更新。该图可沿 X 方向缩放和滚动。最后接收到的读数还会用大字体的文本显示。

PIC32 蓝牙入门工具包具有一个温度传感器，当选择 Start （启动）时，默认情况下温度会周期性地每秒更新一次。通过使用递增或递减按钮 **(1)**，可以在 250 毫秒至 8 秒范围内修改更新速率。此外，还可以通过应用程序中提供的滑块 (2) 修改持续时间。设置持续时间之后，请选择 **Set Timer** （设置定时器）来启动对应于修改后的持续时间的周期性更新。温度更新可以通过选择 **Stop** （停止）暂停。Start/Stop 和 Set Timer 会通过 SPP 全双工传输向 / 从 PIC32 蓝牙入门工具包发送 / 接收命令。发送和接收的命令可以在 Text 菜单中查看。



文本控件

LED 控件

LED 颜色可以使用应用程序的终端仿真器通过发送命令来修改。选择 Text 菜单可以查看应用程序的终端窗口。要修改 LED 颜色，可以使用 `255, 03, R, G, B` 格式发送命令，其中：

- `255` 是用于修改 LED 颜色的命令。
- `03` 决定要发送的字节数，当前它是 3 （R, G, B）
- `R` 指定红色的值，范围为 0-254
- `G` 指定绿色的值，范围为 0-254
- `B` 指定蓝色的值，范围为 0-254

例如， `255, 03, 127, 60, 128` （R = 127， G = 60， B = 128）

命令 “r” 或 “g” 或 “b” 会分别将 LED 设置为 50% 的红色、绿色或蓝色。类似地，命令 “R” 或 “G” 或 “B” 会分别将 LED 设置为 100% 的红色、绿色或蓝色。请参见**表 1：文本命令**来了解详细信息。

温度控件

通过 SPP 数据传输向 PIC32 蓝牙入门工具包发送命令 254。该命令会按请求向 Android 设备终端仿真器发送一次当前温度。类似地，向 PIC32 蓝牙入门工具包发送命令 253 时，将周期性地（默认情况下为每秒一次）接收到当前温度。更新速率可以通过命令 252 加上以毫秒为单位的时间来修改。

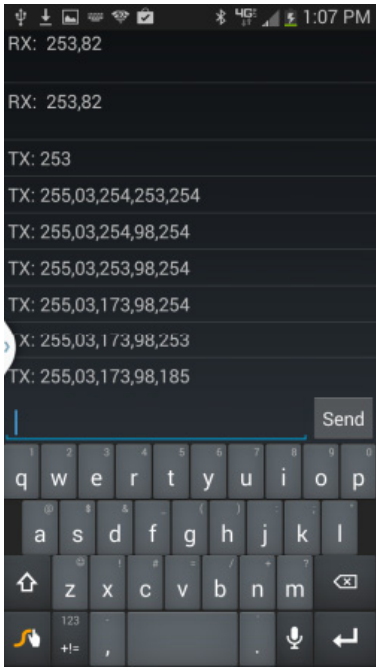
252—— 命令更改定时器更新速率。周期性温度更新可以通过发送命令 253 停止。

请参见**表 1：文本命令**来了解详细信息。

表 1：文本命令

特性	命令	发送格式	接收格式	说明	示例
LED	255	255,03,R,G,B	N/A	根据 R、 G 和 B 的值 （范围分别为 0-254）修改 LED 颜色。	255,03,127,127,127
LED	R	R	N/A	将 LED 设定为 100% 的红色	R
LED	G	G	N/A	将 LED 设定为 100% 的绿色	G
LED	B	B	N/A	将 LED 设定为 100% 的蓝色	B
LED	r	r	N/A	将 LED 设定为 50% 的红色	r
LED	g	g	N/A	将 LED 设定为 50% 的绿色	g

LED	b	b	N/A	将 LED 设定为 50% 的蓝色	b
温度	254	254	华氏温度	当发送 254 命令时， PIC32 蓝牙入门工具包会按请求发送一次当前温度	254
温度	253	253	253，华氏温度	当发送 253 时， PIC32 蓝牙入门工具包会每秒更新温度一次	发送：253 接收：253,80
温度	252	252，以毫秒为单位的速率	253，温度	将 LED 设定为 100% 的绿色	发送：253,399 接收：253,80
温度	253	253	N/A	周期性显示可以通过重新发送 253 暂停	N/A



高级演示

本主题提供关于如何获取、编译、配置和运行需购买的 “高级”演示的信息。

说明

关于购买高级演示的信息，请参见 Microchip 高级 MPLAB Harmony 音频网页 （www.microchip.com/pic32harmonypremiumaudio）。

a2dp_avrcp

注： “高级演示”未包含在 MPLAB Harmony 标准发行版中，必须另外购买。详情请参见 Microchip 高级 MPLAB Harmony 音频网页 （www.microchip.com/pic32harmonypremiumaudio）。

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示对来自任意智能手机（即 Apple、Samsung 和 Google 等）、个人计算机或支持蓝牙的设备的无线蓝牙音频执行流式传输。该演示支持以下特性：

- A2DP
- AVRCP
- SSP

- SBC 解码器

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了高级演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `a2dp_avrcp.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/bluetooth/premium/audio/a2dp_avrcp`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`a2dp_avrcp.X`	`<install-dir>/apps/bluetooth/premium/audio/a2dp_avrcp/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
ak7755_bt_audio_dk	bt_audio_dk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙音频开发工具包和 AKM AK7755 编解码器上运行演示。
bt_audio_dk	bt_audio_dk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙音频开发工具包上运行演示。
pic32mz_da_sk_meb2	pic32mz_da_sk+meb2	选择该 MPLAB X IDE 项目配置可以在连接到多媒体扩展板 II （Multimedia Expansion Board II， MEB II）的 PIC32MZ DA 入门工具包上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙音频开发工具包和 AK7755 编解码器

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行高级音频演示。

说明

1. 编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。
2. 当 LED2 和 LED3 点亮时，表明该演示正在 PIC32 蓝牙音频开发工具包上运行。
3. 在蓝牙音频设备 （如智能手机）上使能蓝牙。
4. 扫描可用的蓝牙设备。目标蓝牙设备也应显示在您的蓝牙音频设备的可用设备列表中。
5. 目标蓝牙设备的名称将如下：

配置	设备名称
bt_audio_dk	Microchip A2DP
ak7755_bt_audio_dk	Microchip A2DP
pic32mz_da_sk_meb2	BT_DA + MEB-II

注： 有时，蓝牙设备的名称尚未解析，将显示为 “null”。一段时间后名称会从 “null”更改为前面列出的特定于配置的名称。可见 MAC 地址的前 8 位是固定的，后 4 位可变 （12:34:45:78:XX: XX）。

6. 选择要进行配对的器件并连接。
7. 如果您的设备提示输入 PIN，则输入 0000。
8. 如果连接成功，蓝牙音频设备屏幕顶部会显示消息 “connected to <Device Name>”。
9. 将扬声器或耳机连接到开发板的 line-out/ 耳机插孔。
10. 选择音乐曲目，然后点击 **Play** （播放）。

演示控件

蓝牙模式	控件	bt_audio_dk ak 7755_bt_audio_dk	pic32mz_da_sk_meb2
随机播放 （切换） / 强制蓝牙设备取消配对	SW1	开关 SW1，位于电路板的顶部。	N/A
重复曲目 （切换） / 蓝牙设备断开	SW2	开关 SW2，位于电路板的顶部。	N/A
下一曲目 / 快进	SW3	开关 SW3，位于电路板的顶部。	N/A
播放 / 暂停 （切换） / 软静音 （切换）	SW4	开关 SW4，位于电路板的顶部。	N/A
上一曲目 / 倒带	SW5/S1	开关 SW5，位于电路板的顶部。	开关 S1，位于电路板的顶部。不支持入门工具包板上的开关 SW1、 SW2 和 SW3。
N/A	LED1	红色 LED D5，位于电路板的顶部。	红色 LED D3，位于电路板的顶部。
蓝牙设备连接就绪	LED2	红色 LED D6，位于电路板的顶部。	红色 LED D4，位于电路板的顶部。
蓝牙设备连接就绪	LED3	红色 LED D7，位于电路板的顶部。	红色 LED D5，位于电路板的顶部。
音频流指示	LED4	红色 LED D8，位于电路板的顶部。	红色 LED D6，位于电路板的顶部。
N/A	LED5	红色 LED D9，位于电路板的顶部。	红色 LED D7，位于电路板的顶部。
CPU 异常错误	LED1-LED5	红色 LED D5-D9，位于电路板的顶部。	红色 LED D3-D7，位于电路板的顶部。

自举程序演示

本主题介绍自举程序演示。

简介

自举程序演示应用程序帮助。

说明

该分发包包含用于演示 MPLAB Harmony 自举程序功能的固件项目。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的自举程序演示应用程序。

basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示实现了驻留在引导闪存中的自举程序。当自举程序在目标器件上运行时，无需外部编程器或调试器，就可以使用应用程序代码对器件编程。

该自举程序在操作上类似于 AN1388 《PIC32 自举程序》中介绍的自举程序，并与随相关的源代码归档文件提供的个人计算机应用程序配合使用。应用笔记和归档文件可从 Microchip 网站 （www.microchip.com）下载。

作为一种替代方法，可以按 Bootloader Library Help（自举程序库帮助）的 Bootloader Plug-in for MPLAB X IDE（用于 MPLAB X IDE 的自举程序插件）部分所述，使用可用于 MPLAB X IDE 的自举程序插件来烧写自举程序。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了自举程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/bootloader/basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`basic.X`	`<install-dir>/apps/bootloader/basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
i2cslave_pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 I2C 从模式。
udp_pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 UDP 以太网模式。
udp_pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 UDP 以太网模式。
usart_pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 UART 自举程序。
usart_pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 UART 自举程序。 **注：**该演示不依赖于硬件加密模块。
usart_pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 UART 自举程序。 **注：**该演示不依赖于硬件加密模块。
usart_pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ DA 入门工具包上演示 UART 自举程序。 **注：**该演示不依赖于硬件加密模块。
usbdevice_pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示 USB 设备模式。
usbdevice_pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 USB 设备模式。
usbhost_pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示 USB 主机自举程序。
usbhost_pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 USB 主机自举程序。 **注：**该演示不依赖于硬件加密模块。
usbhost_pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 USB 主机自举程序。 **注：**该演示不依赖于硬件加密模块。

配置硬件

介绍如何配置支持的硬件。

说明

以下配置信息用于 UART 模式。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包或 PIC32 以太网入门工具包 II

UART 通信通过 UART2 模块来完成。U2RX 和 U2TX 引脚可以通过入门工具包 I/O 扩展板来访问。一种访问方法是使用 MCP2200 转接模块。将模块的 TX 引脚连接到连接器 J11 上的引脚 46。将模块的 RX 引脚连接到连接器 J11 上的引脚 48。将 GND 引脚连接在一起，以确保共地。

图 1 和图 2 显示了硬件配置和跳线连接的近距离特写。

图 1—— 硬件配置

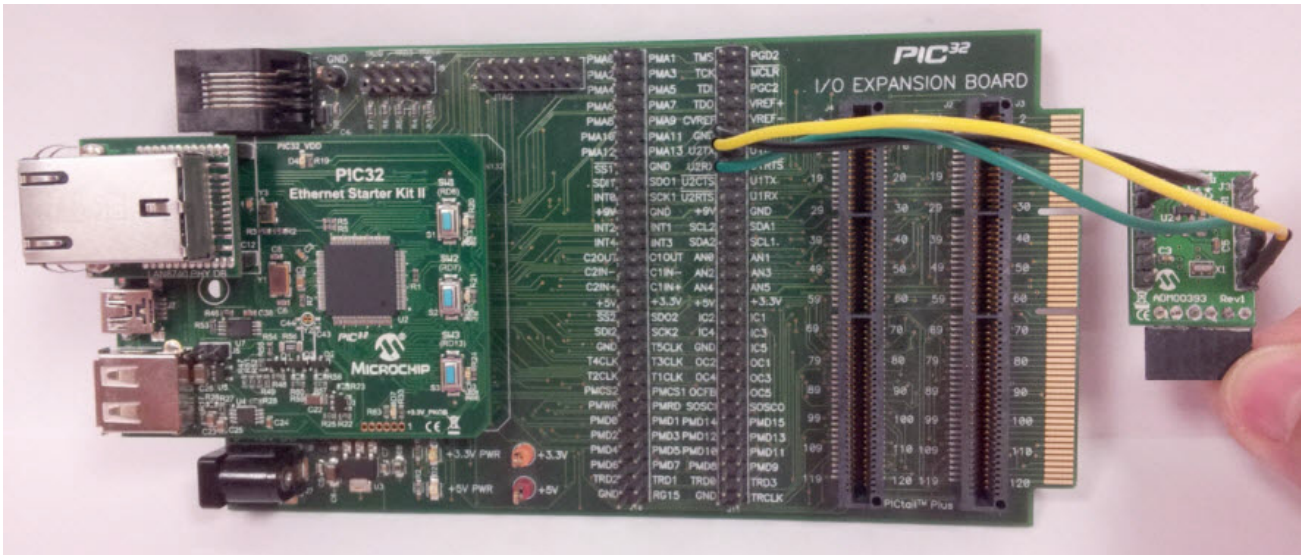
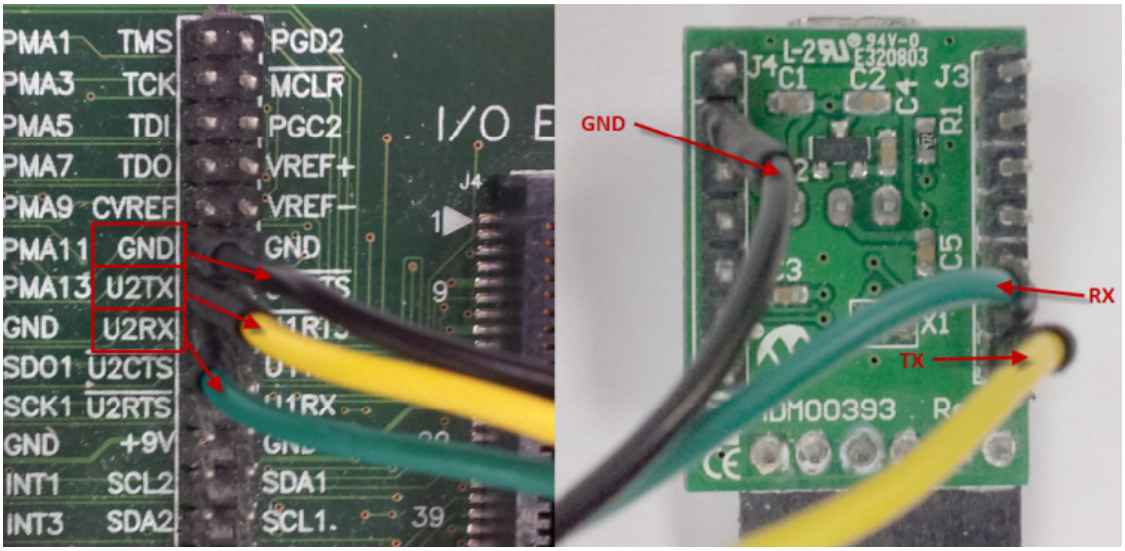


图 2—— 跳线连接



PIC32MZ EC 入门工具包

UART 通信通过 UART1 模块（通过 PPS 连接至 RPF0 和 RPF1）来完成。U1RX 和 U1TX 引脚可以通过 PIC32MZ EC 适配器板来访问。一种访问方法是使用 MCP2200 转接模块。将模块的 TX 引脚连接到适配器背面的 JP3 的 EBID10 引脚。将模块的 RX 引脚连接到适配器背面的 JP1 的 EBID11 引脚。连接模块和入门工具包的地，确保共地。

注： 在 PIC32MZ EC 入门工具包上运行 UDP 自举程序时，自举程序可能由于入门工具包和 LAN8740 PHY 子卡之间的连接问题而无法运行。
关于修正该问题的选项，请参见 LAN8740 PHY 子卡的产品变更通知。该 PCN 可从 PIC32MZ EC 入门工具包网页获取：
http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=Dm320006

图 3 和图 4 显示了硬件配置。

图 3—— 硬件配置 （正面）

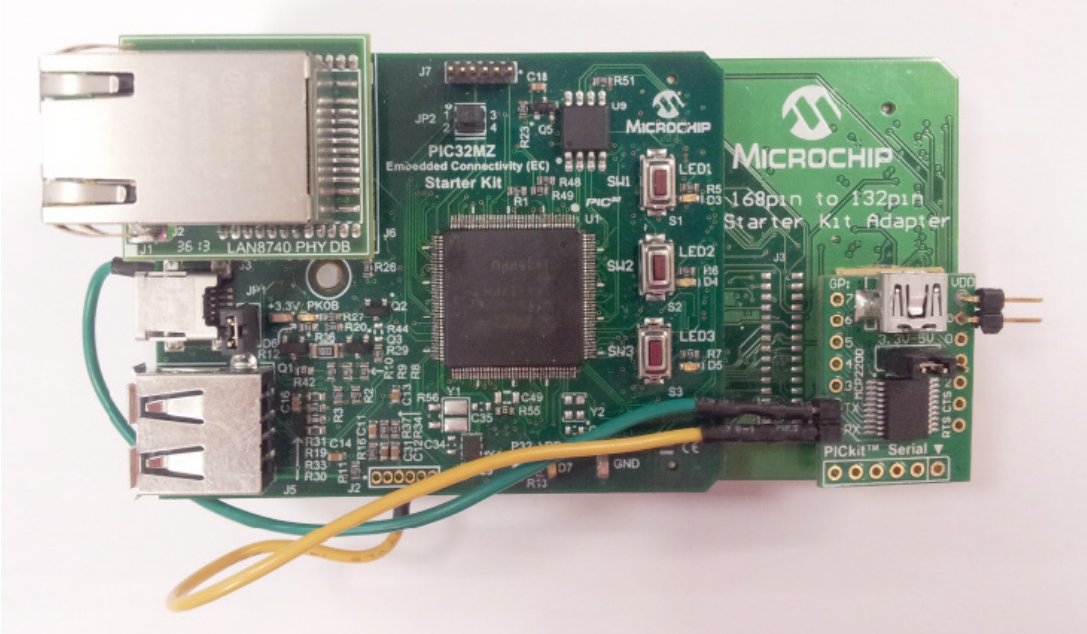
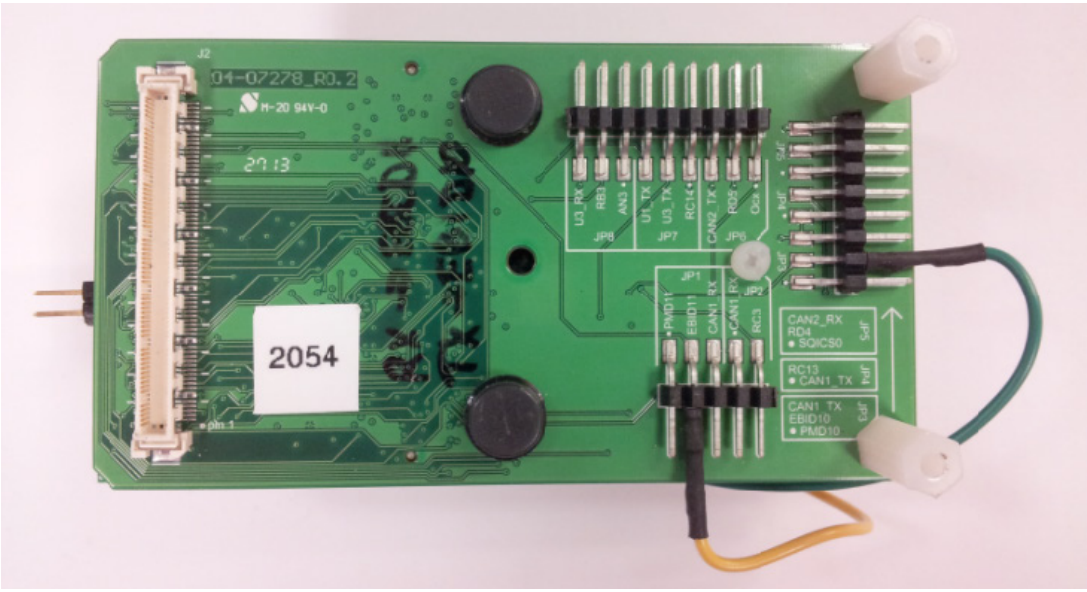


图 4—— 硬件配置 （背面）



PIC32MZ EF 入门工具包

将主机计算机连接到 PIC32MZ EF 入门工具包的 J11。

PIC32MZ 图形 （DA）入门工具包

将主机计算机连接到 PIC32MZ DA 入门工具包的 J5。

运行演示

介绍如何编译并运行自举程序演示。

说明

基于个人计算机的主机演示

使用这些配置时，请执行以下操作：

操作

自举程序的操作方式如下：

1. 选择适合于目标硬件的配置。
2. 编译并对器件编程。

3. 按住入门工具包上的 SW3，强制自举程序工作。
4. LED1 会开始闪烁，指示自举程序正在工作。如果先前烧写了某个程序，则可能需要先按住 SW3，之后再向电路板加电或复位电路板。
5. 从 AN1388 源代码归档文件中打开个人计算机主机程序。
6. 选择适当的通信路径：
   - UART—— 将波特率保留为 115,200
   - UDP—— 将 IP 地址保留为 192.168.1.11， UDP 端口保留为 6234
   - USB 设备 —— 将 VID 保留为 0x4D8， PID 保留为 0x03C
7. 单击 **Connect** （连接）来连接自举程序并获取版本。
8. 如果连接了自举程序，个人计算机主机应用程序将会指示自举程序的版本。

此时，自举程序已准备好接受新的应用程序，用于烧写到程序闪存中。提供了一个演示应用程序，它在链接器脚本中配置为只在程序闪存中工作。

设置

如下准备演示应用程序：

1. 从 `<install-dir>/apps/examples/peripheral/dma/firmware` 中打开 `dma_led_pattern.X` 项目。
2. 选择适合于目标硬件的配置。
3. 编译该程序，但**不对器件编程**。

对器件编程

演示应用程序已经编译，现在可以使用自举程序将生成的十六进制文件烧写到器件中。

要将应用程序烧写到器件中：

1. 在 AN1388 介绍的个人计算机主机应用程序中，单击 **Load Hex File** （装入十六进制文件）。
2. 浏览至 `<install-dir>/apps/examples/peripheral/dma/firmware/dma_led_pattern.X/dist/<configuration>/production/<configuration>.X.production.hex`。

选择它并单击 **Open**。

3. 单击 **Erase-Program-Verify** （擦除 - 编程 - 校验）。
4. 然后，会将程序传送给自举程序，自举程序会烧写该应用程序。
5. 当程序烧写完成时，单击 **Run Application** （运行应用程序）来启动程序。
6. 如果应用程序已被正确烧写，绿色和红色 LED 会交替闪烁。
7. 单击 **Disconnect** （断开连接）释放主机应用程序。

注： 作为一种替代方法，可以按 Bootloader Library Help （自举程序库帮助）的 Bootloader Plug-in for MPLAB X IDE （用于 MPLAB X IDE 的自举程序插件）部分所述，使用可用于 MPLAB X IDE 的自举程序插件来烧写自举程序。

基于 USB 主机的演示

使用这些配置时，请执行以下操作：

操作

自举程序的操作方式如下：

1. 选择适合于目标硬件的配置。
2. 编译并对器件编程。
3. 按住入门工具包上的 SW3，强制自举程序工作。

如果先前烧写了某个程序，则可能需要先按住 SW3，之后再向电路板加电或复位电路板。

设置

如下准备演示应用程序：

1. 从 `<install-dir>/apps/examples/peripheral/dma/firmware/` 中打开 `dma_led_pattern.X` 项目。
2. 选择适合于目标硬件的配置。
3. 编译该程序，但不对器件编程。
4. 将生成的 .hex 文件复制到将插入目标 USB 端口的闪存驱动器 （U 盘）。该文件可以在 `<install-dir>/apps/examples/peripheral/dma/firmware/dma_led_pattern.X/dist/<platform>/production/dma_led_pattern.X.production.hex` 找到。
5. 将闪存驱动器上的十六进制文件重命名为 `image.hex`。

对器件编程

演示应用程序已经编译，现在可以使用自举程序将生成的十六进制文件烧写到器件中。

要将应用程序烧写到器件中：

1. 将闪存驱动器插入入门工具包上的 A 型 USB 端口。
2. 将会从闪存驱动器中装入该程序，并烧写到器件中。
3. 当程序开始运行时，移除闪存驱动器。

LiveUpdate

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示实现了一个在线更新自举程序，该自举程序会在启动过程中查找使 PIC32MZ EC 入门工具包上的闪存分区发生交换的变更。对于该自举程序演示，可以使用与基本自举程序演示相同的示例应用程序作为编程示例的应用程序来源。

该自举程序在操作上类似于 AN1388 《PIC32 自举程序》中介绍的自举程序，并与随相关的源代码归档文件提供的个人计算机应用程序配合使用。应用笔记和归档文件可从 Microchip 网站 （www.microchip.com）下载。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了自举程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `liveUpdate.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/bootloader/LiveUpdate`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`liveUpdate.X`	`<install-dir>/apps/bootloader/LiveUpdate/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
usart_pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 UART 自举程序。 **注：** 该演示不依赖于硬件加密模块。
usart_pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 UART 自举程序。 **注：** 该演示不依赖于硬件加密模块。
usart_pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ EF 入门工具包上演示 UART 自举程序。 **注：** 该演示不依赖于硬件加密模块。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

UART 通信通过 UART1 模块（通过 PPS 连接至 RPF0 和 RPF1）来完成。U1RX 和 U1TX 引脚可以通过 PIC32MZ EC 适配器板来访问。一种访问方法是使用 MCP2200 转接模块。将模块的 TX 引脚连接到适配器背面的 JP3 的 EBID10 引脚。将模块的 RX 引脚连接到适配器背面的 JP1 的 EBID11 引脚。连接模块和入门工具包的地，确保共地。

图 3 和图 4 显示了硬件配置。

图 3—— 硬件配置 （正面）

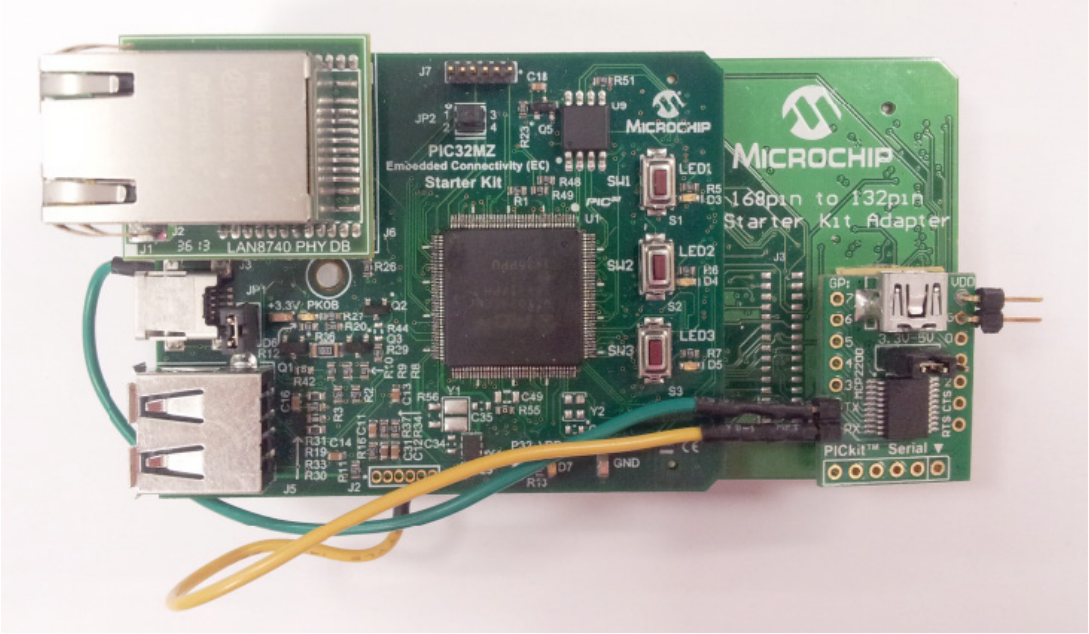
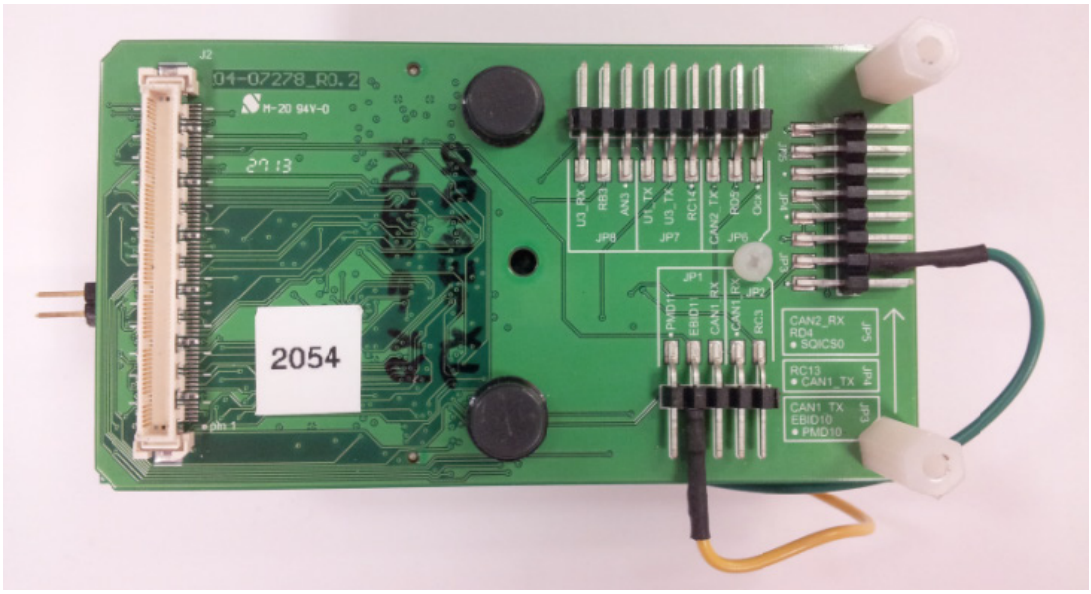


图 4—— 硬件配置 （背面）



PIC32MZ EF 入门工具包

将主机计算机连接到 PIC32MZ EF 入门工具包的 J11。

PIC32MZ 图形 （DA）入门工具包

将主机计算机连接到 PIC32MZ DA 入门工具包的 J5。

运行演示

介绍如何编译并运行自举程序演示。

说明

基于个人计算机的主机演示

使用这些配置时，请执行以下操作：

操作

自举程序的操作方式如下：

1. 选择适合于目标硬件的配置。
2. 编译并对器件编程。
3. LED1 会开始闪烁，指示程序正在工作。
4. 从 AN1388 源代码归档文件中打开个人计算机主机程序。
5. 选择适当的通信路径：
   - UART—— 将波特率保留为 115,200
   - UDP—— 将 IP 地址保留为 192.168.1.11， UDP 端口保留为 6234
   - USB 设备 —— 将 VID 保留为 0x4D8， PID 保留为 0x03C
6. 单击 **Connect** 来连接自举程序并获取版本。
7. 如果连接了自举程序，个人计算机主机应用程序将会指示自举程序的版本。

此时，自举程序已准备好接受新的应用程序，用于烧写到程序闪存中。提供了一个演示应用程序，它在链接器脚本中配置为只在程序闪存中工作。

设置

如下准备演示应用程序：

1. 从 `<install-dir>/apps/examples/peripheral/dma/firmware` 中打开 `dma_led_pattern.X` 项目。
2. 选择适合于目标硬件的配置。
3. 编译该程序，但**不对器件编程**。

对器件编程

演示应用程序已经编译，现在可以使用 LiveUpdate 将生成的十六进制文件烧写到器件中。

要将应用程序烧写到器件中：

1. 在 AN1388 中介绍的个人计算机主机应用程序中，单击 **Load Hex File**。
2. 浏览至 `<install-dir>/apps/examples/peripheral/dma/firmware/dma_led_pattern.X/dist/<configuration>/production/<configuration>.X.production.hex`。
   选择它并单击 **Open**。
3. 单击 **Erase-Program-Verify**。
4. 然后，程序会被传送给自举程序，自举程序会将该应用程序烧写到第二个闪存分区中。
5. 当程序烧写完成时，单击 **Run Application** 来启动程序。
6. 如果应用程序已被正确烧写，绿色和红色 LED 会交替闪烁。
7. 单击 **Disconnect** 释放主机应用程序。

注： 作为一种替代方法，可以按 Bootloader Library Help （自举程序库帮助）的 Bootloader Plug-in for MPLAB X IDE （用于 MPLAB X IDE 的自举程序插件）部分所述，使用可用于 MPLAB X IDE 的自举程序插件来烧写自举程序。

基于 USB 主机的演示

使用这些配置时，请执行以下操作：

操作

自举程序的操作方式如下：

1. 选择适合于目标硬件的配置。
2. 编译并对器件编程。

设置

如下准备演示应用程序：

1. 从 `<install-dir>/apps/examples/peripheral/dma/firmware/` 中打开 `dma_led_pattern.X` 项目。
2. 选择适合于目标硬件的配置。
3. 编译该程序，但不对器件编程。

4. 将生成的 .hex 文件复制到将插入目标 USB 端口的闪存驱动器。该文件可以在 `<install-dir>/apps/examples/peripheral/dma/firmware/dma_led_pattern.X/dist/<platform>/production/dma_led_pattern.X.production.hex` 找到。
5. 将闪存驱动器上的十六进制文件重命名为 `image.hex`。

对器件编程

演示应用程序已经编译，现在可以使用自举程序将生成的十六进制文件烧写到器件中。要将应用程序烧写到器件中：

1. 将闪存驱动器插入入门工具包上的 A 型 USB 端口。
2. 将会从闪存驱动器中装入该程序，并烧写到器件中。
3. 当程序开始运行时，移除闪存驱动器。

LiveUpdate_StateRetention

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示实现了一个在线更新自举程序，该自举程序会在启动过程中查找使 PIC32MZ EF 入门工具包上的闪存分区发生交换的变更。对于该自举程序演示，可以使用与基本自举程序演示相同的示例应用程序作为编程示例的应用程序来源。

该自举程序在操作上类似于 AN1388 《PIC32 自举程序》中介绍的自举程序，并与随相关的源代码归档文件提供的个人计算机应用程序配合使用。应用笔记和归档文件可从 Microchip 网站 （www.microchip.com）下载。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了自举程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `LiveUpdate_StateRetention.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/bootloader/LiveUpdate_StateRetention`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`LiveUpdate_StateRetention.X`	`<install-dir>/apps/bootloader/LiveUpdate_StateRetention/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk_meb2_v1	pic32mz_ef_sk+meb2	在 PIC32MZ EF 入门工具包上通过 UART 自举程序演示在线更新。
pic32mz_ef_sk_meb2_v2	pic32mz_ef_sk+meb2	对 v1 配置稍加修改，在 PIC32MZ EF 入门工具包上通过 UART 自举程序演示在线更新。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包

将主机计算机连接到 PIC32MZ EF 入门工具包的 J11。

运行演示

介绍如何编译并运行在线更新状态保持演示。

说明

基于个人计算机的主机演示

使用这些配置时，请执行以下操作：

操作

自举程序的操作方式如下：

1. 选择适合于目标硬件的配置。
2. 编译并对器件编程。
3. LED1 会开始闪烁，指示程序正在工作。
4. 从 AN1388 源代码归档文件中打开个人计算机主机程序。
5. 选择适当的 UART 通信路径，将波特率保留为 115,200
6. 单击 **Connect** 来连接自举程序并获取版本。
7. 如果连接了自举程序，个人计算机主机应用程序将会指示自举程序的版本。

此时，自举程序已准备好接受新的应用程序，用于烧写到程序闪存中。提供了一个演示应用程序，它在链接器脚本中配置为只在程序闪存中工作。

设置

要运行演示，两个配置（v1 和 v2）都需要进行编译。选择要编译的配置之一，并烧写到入门工具包中，但另一个配置只进行编译。记下只进行编译的那一个配置，因为演示的剩余部分将需要它。

对器件编程

演示应用程序已经编译，现在可以使用 LiveUpdate 将生成的十六进制文件烧写到器件中。

要将应用程序烧写到器件中：

1. 在 AN1388 中介绍的个人计算机主机应用程序中，单击 **Load Hex File**。
2. 浏览至
   `<install-dir>/apps/bootloader/LiveUpdate_StateRetention/firmware/LiveUpdate_StateRetention.X/dist/<configuration>/production/LiveUpdate_StateRetention.X.production.hex`。
   选择它并单击 **Open**。
3. 单击 **Erase-Program-Verify**。
4. 然后，程序会被传送给自举程序，自举程序会将该应用程序烧写到第二个闪存分区中。
5. 当程序烧写完成时，单击 **Run Application** 来启动程序。
6. 如果应用程序已被正确烧写，绿色和红色 LED 会交替闪烁。
7. 单击 **Disconnect** 释放主机应用程序。

注： 作为一种替代方法，可以按 Bootloader Library Help （自举程序库帮助）的 Bootloader Plug-in for MPLAB X IDE （用于 MPLAB X IDE 的自举程序插件）部分所述，使用可用于 MPLAB X IDE 的自举程序插件来烧写自举程序。

加密演示

本主题介绍加密演示。

简介

加密库演示应用程序帮助。

说明

该分发包包含三个与加密相关的固件项目，用于演示 MPLAB Harmony 加密库的功能。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的加密库演示应用程序。

encrypt_decrypt

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示通过运用几种加密功能（包括 MD5、TDES、DES、AES、RSA、ECC 和随机数发生）来验证软件或硬件是否正确执行。在演示运行时，入门工具包上的黄色 LED 将点亮，指示正在进行处理。如果所有功能都执行成功，入门工具包上的绿色 LED 将点亮。

在测试硬件加密时，必须使用带加密引擎的入门工具包（DM320006-C）。软件加密可以在 PIC32MX795F512L 或任意版本的 PIC32MZ 器件上执行。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了加密演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `encrypt_decrypt.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/crypto/encrypt_decrypt`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`encrypt_decrypt.X`	`<install-dir>/apps/crypto/encrypt_decrypt/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上使用软件库来演示加密、解密、哈希值计算和随机数发生。
pic32mz_ec_sk_hw	pic32mz_ec_s	使用带加密功能的 PIC32MZ 嵌入式连接（Embedded Connectivity，EC）入门工具包上的硬件加密模块来演示加密、解密、哈希值计算和随机数发生。
pic32mz_ef_sk_hw	pic32mz_ef_sk	使用带加密功能的 PIC32MZ 嵌入式连接与浮点单元（Embedded Connectivity with Floating Point Unit，EF）入门工具包上的硬件加密模块来演示加密、解密、哈希值计算和随机数发生。
pic32mz_da_sk_hw	pic32mz_da_sk	使用带加密功能的 PIC32MZ DA 入门工具包上的硬件加密模块来演示加密、解密、哈希值计算和随机数发生。
pic32mz_ec_sk_sw	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接（EC）入门工具包上使用软件库来演示加密、解密、哈希值计算和随机数发生。
pic32mz_ef_sk_sw	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上使用软件库来演示加密、解密、哈希值计算和随机数发生。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行加密演示。

说明

该演示会运用各种加密、解密、哈希值计算和随机数功能。

1. 首先编译并对目标器件编程。在编译时，选择对应于所用硬件的配置。
2. 观察入门工具包上的 LED 的状态。在演示执行时，黄色 LED 会点亮。如果所有功能都成功通过，绿色 LED 会点亮。如果发生错误，红色 LED 会点亮。

large_hash

本部分介绍受支持的演示板，说明了如何配置硬件（如需要）以及如何运行演示。

说明

该应用程序演示如何对大数据块执行哈希值计算。在此例中，演示会对由字母“a”构成的大小为 512 * 1024 的块执行 MD5、SHA-1、SHA-256、 SHA-384 和 SHA-512 哈希值计算。

在 PIC32MZ 器件上，具有足够的闪存，链接器脚本配置为从物理地址 0x9D08_0000 开始创建大小为 512 * 1024 的块。

应用程序通过两种方式运行哈希值计算：

- 在 PIC32MZ 器件上，它的第一种运行方法是在一个函数调用中传递整个 512 * 1024 的块。
- 第二种方式是唯一一种在 PIC32MX 器件上运行的方式，它会将分配在堆栈中的由字母 “a” 构成的大小为 1024 的块传递给引擎，此操作重复执行 512 次。

在执行哈希值计算之后，应用程序通过系统控制台输出哈希值计算的结果，以及它执行每种形式所用的时间。然后，它会将每种算法生成的哈希值与已知值进行比较。如果所有测试都通过，绿色 LED 会点亮，并通过系统控制台发出一条消息。如果有任何测试失败，红色 LED 会点亮，并通过系统控制台发出相应的消息。

在测试硬件加密时，必须使用配有加密引擎的 PIC32MZ EC 入门工具包 （DM320006-C）。软件加密可以在任意版本的 PIC32MX 或 PIC32MZ 器件上执行。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了加密演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `large_hash.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/crypto/large_hash`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`large_hash.X`	`<install-dir>/apps/crypto/large_hash/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上使用软件库来演示大数据块的哈希值计算。
pic32mz_ec_sk_hw	pic32mz_ec_sk	使用带加密功能的 PIC32MZ 嵌入式连接 （EC）入门工具包上的硬件加密模块来演示大数据块的哈希值计算。
pic32mz_ef_sk_hw	pic32mz_ef_sk	使用带加密功能的 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上的硬件加密模块来演示大数据块的哈希值计算。
pic32mz_da_sk_hw	pic32mz_da_sk	使用带加密功能的 PIC32MZ 图形 （DA）入门工具包上的硬件加密模块来演示大数据块的哈希值计算。
pic32mz_ec_sk_sw	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上使用软件库来演示大数据块的哈希值计算。
pic32mz_ef_sk_sw	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上使用软件库来演示大数据块的哈希值计算。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

UART 通信通过 UART2 模块来完成。 U2RX 和 U2TX 引脚可以通过入门工具包 I/O 扩展板来访问。一种访问方法是使用 MCP2200 转接模块。将模块的 TX 引脚连接到连接器 J11 上的引脚 46。将模块的 RX 引脚连接到连接器 J11 上的引脚 48。将 GND 引脚连接在一起，以确保共地。

图 1 和图 2 显示了硬件配置和跳线连接的近距离特写。

图 1—— 硬件配置

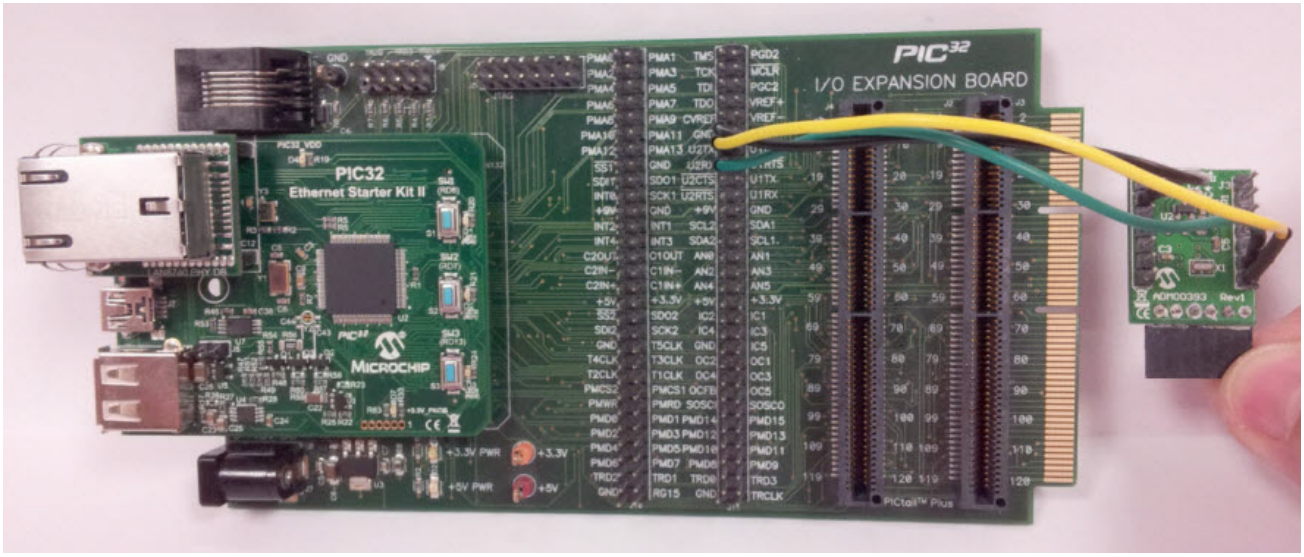
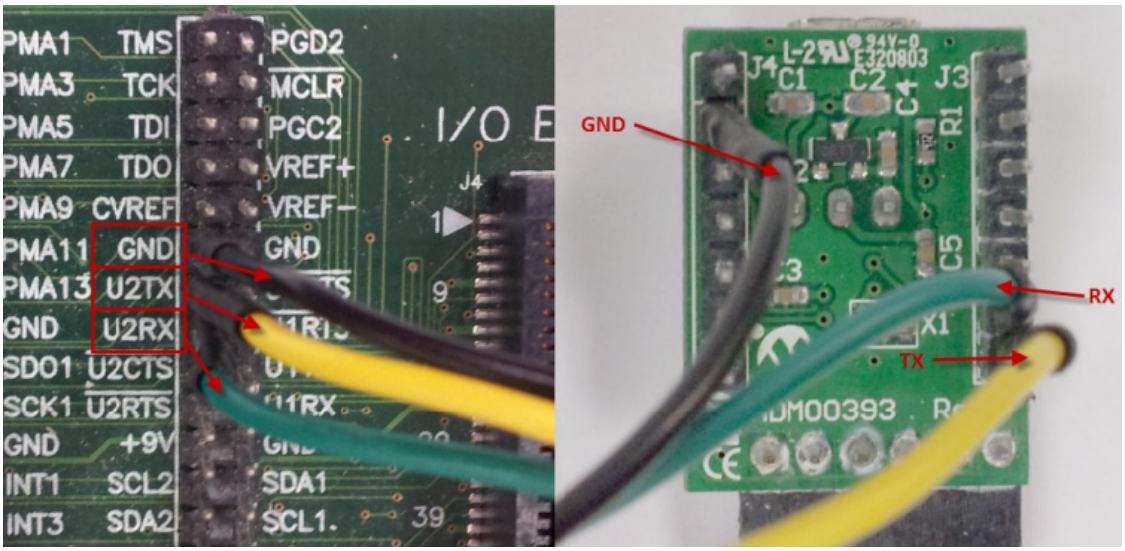


图 2—— 跳线连接



PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ 嵌入式连接 （EC）入门工具包

UART 通信通过 UART1 模块（通过 PPS 连接至 RPF0 和 RPF1）来完成。U1RX 和 U1TX 引脚可以通过 PIC32MZ EC 适配器板来访问。一种访问方法是使用 MCP2200 转接模块。将模块的 TX 引脚连接到适配器背面的 JP3 的 EBID10 引脚。将模块的 RX 引脚连接到适配器背面的 JP1 的 EBID11 引脚。连接模块和入门工具包的地，确保共地。

图 3 和图 4 显示了硬件配置。

图 3—— 硬件配置 （正面）

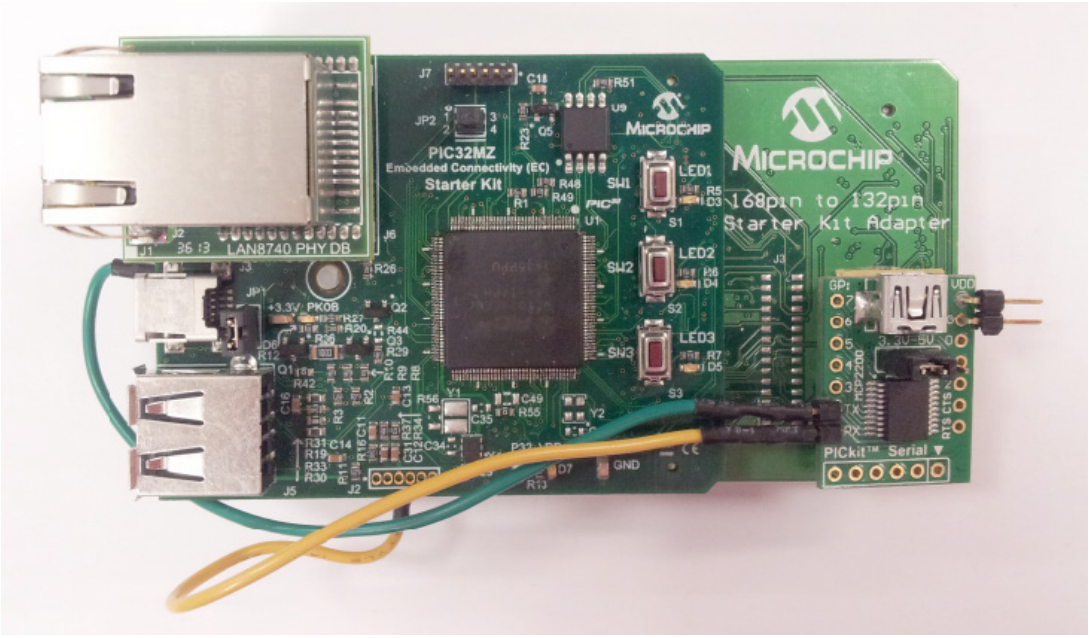
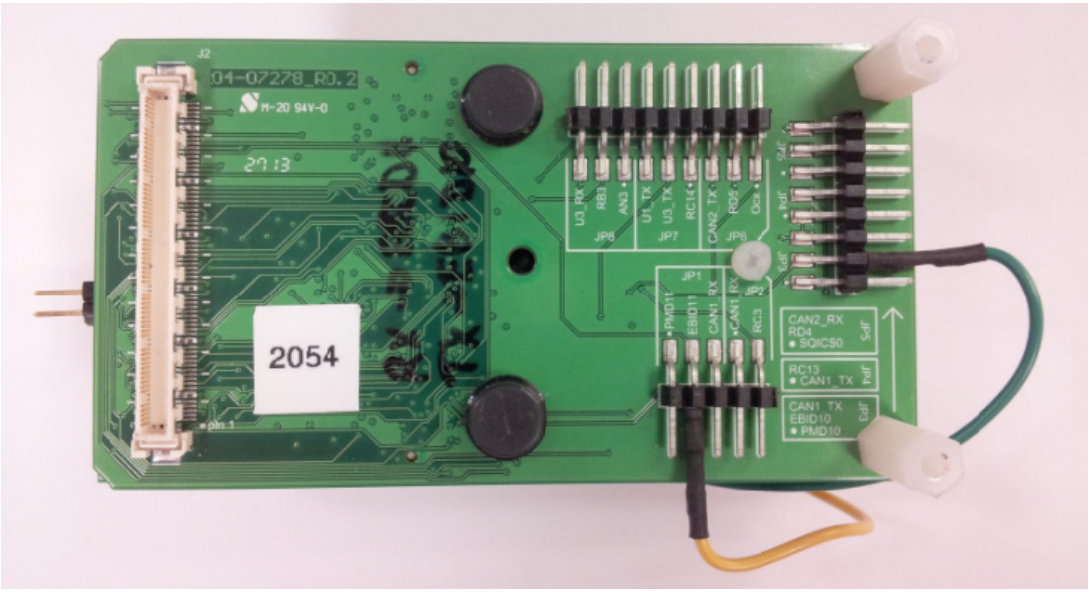


图 4—— 硬件配置 （背面）



运行演示

介绍如何编译并运行加密演示。

说明

该演示会对大数据块运用哈希函数。

1. 首先编译并对目标器件编程。在编译时，选择适合于硬件的配置。
2. 打开串行终端程序（如 PuTTY 或 TeraTerm），并将其连接到 MCP2200 转接模块的串行端口。串行配置为 115200 波特率、8 个数据位、无奇偶校验位和 1 个停止位。
3. 观察入门工具包上的 LED 的状态。在演示执行时，黄色 LED 会点亮。如果所有功能都成功通过，绿色 LED 会点亮。如果发生错误，红色 LED 会点亮。
4. 观察程序在串行终端程序中的输出。它会报告哈希值计算的结果，以及执行所用的周期数。所用的实际周期数取决于所用的硬件和硬件引擎可用的缓冲区大小。以下示例显示了使用配有加密引擎的 PIC32MZ EC 入门工具包时的输出：

```
Starting the test.

MD5 from Flash: 30C2557E8302A5BEB290C71520D87F42 took 481405 clock cycles
MD5 from feed: 30C2557E8302A5BEB290C71520D87F42 took 804934 clock cycles
SHA from Flash: F7FEC128D7FCD59222BA37368D3B7210D4C7B6EF took 481151 clock cycles
SHA from feed: F7FEC128D7FCD59222BA37368D3B7210D4C7B6EF took 804901 clock cycles
SHA256 from Flash: 85A84A75886E8A526DBEC4E16E3375FAA307B4AEAD79C9ED3264C0477A6F6EBA took 702033
clock cycles
SHA256 from feed: 85A84A75886E8A526DBEC4E16E3375FAA307B4AEAD79C9ED3264C0477A6F6EBA took 806480
clock cycles
SHA384 from Flash:
A550561A6330048EFE826A97E5FED843FA1CE646A9BF546CCB433C2FCB0E54821C4C945EED9A592B5BF43157E212F277

took 45452328 clock cycles
SHA384 from feed:
A550561A6330048EFE826A97E5FED843FA1CE646A9BF546CCB433C2FCB0E54821C4C945EED9A592B5BF43157E212F277
took 45391039 clock cycles
SHA512 from Flash:
7F49157FB359B39EA6DA934DC9A10709FEDF8846D139D0E637A3C0FC833B6F42703858DBACEE28F4489B5E95FAB5E565
5A25F838B0DC7BF3C84C7CC0264F6A4F
took 45807606 clock cycles
SHA512 from feed:
7F49157FB359B39EA6DA934DC9A10709FEDF8846D139D0E637A3C0FC833B6F42703858DBACEE28F4489B5E95FAB5E565
5A25F838B0DC7BF3C84C7CC0264F6A4F
took 45775518 clock cycles
All tests passed.
```

驱动程序演示

本部分介绍驱动程序演示。

I2C 驱动程序演示

本主题介绍 I2C 驱动程序演示。

简介

本帮助文件包含关于 MPLAB Harmony I2C 驱动程序应用程序演示 （包含在 MPLAB Harmony 库分发包中）的说明和关联信息。

说明

该应用程序演示 MPLAB Harmony I2C 驱动程序的功能。本部分介绍硬件要求，以及使用 Microchip 开发工具编译并运行演示项目的过程。

提供了一个演示应用程序：

- i2c_rtcc—— 在该演示中，I2C 外设的一个实例充当主器件，向 / 从两个外部从器件发送 / 接收数据。从器件是外部 MCP7049N 实时时钟日历 （Real-Time Clock Calendar， RTCC）器件。

要了解关于 MPLAB Harmony I2C 驱动程序、配置驱动程序和 I2C 驱动程序提供的 API 的更多信息，请参见 I2C 驱动程序库帮助文档。

演示

本主题介绍如何运行本发行版中包含的 I2C 驱动程序演示应用程序。

i2c_rtcc

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 I2C RTCC 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `i2c_rtcc.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/i2c/i2c_rtcc`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`i2c_rtcc.X`	`<install-dir>/apps/driver/i2c/i2c_rtcc/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	该配置的目的是在中断模式和动态操作下演示 I2C 主模式传输设置。所用的硬件是连接到 Explorer 16 开发板的 PIC32MX795F512L PIM。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置的目的是在中断模式和动态操作下演示 I2C 主模式传输设置。所用的硬件是连接到 MEB II 的 PIC32MZ EC 入门工具包。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置的目的是在中断模式和动态操作下演示 I2C 主模式传输设置。所用的硬件是连接到 MEB II 的 PIC32MZ EF 入门工具包。
pic32mz_da_sk_meb2_wvga	pic32mz_da_sk+meb2+wvga	该配置的目的是在中断模式和动态操作下演示 I2C 主模式传输设置。所用的硬件是连接到 MEB II 的 PIC32MZ 图形 （DA）入门工具包。

配置硬件

介绍如何配置支持的硬件。

说明

注： i2c_rtcc 演示在 Microchip MCP7949N RTCC 器件上进行了测试。该器件的地址为 0xDE。

Explorer 16 开发板与 PIC32MX795F512L PIM

- 在将 PIC32MX795F512L PIM 连接到 Explorer 16 开发板之前，请确保处理器选择开关 （S2）处于 PIM 位置
- 短接 Explorer 16 开发板上的 JP2，以使能 LED

如果使用入门工具包 I/O 扩展板，则进行以下连接：

- I/O 扩展板 J11 引脚 36 （SDA2）和 J11 引脚 38 （SCL2）应通过 2.2k 欧姆的电阻上拉至 3.3V

如果使用外部 RTCC，则进行以下连接：

- 通过跳线将 I/O 扩展板 J11 引脚 36 （SDA2）和 J11 引脚 38 （SCL2）连接到外部 I2C 器件相应的信号线
- 如果连接了外部 I2C 器件，请确保主器件和从器件共用一个公共地。

如果使用 PICtail™ Plus 子板，则进行以下连接：

- PICtail Plus 子板引脚 RA2 （SCL2）和 RA3 （SDA2）应通过 2.2k 欧姆的电阻上拉至 3.3V

如果使用外部 RTCC，则进行以下连接：

- 通过跳线将 PICtail Plus 子板引脚 RA2 （SCL2）和引脚 RA3 （SDA2）连接到外部 I2C 器件相应的 SCL 和 SDA 线
- 如果连接了外部 I2C 器件，请确保主器件和从器件共用一个公共地。

连接到 MEB II 的 PIC32MZ EC 入门工具包、 PIC32MZ EF 入门工具包或 PIC32MZ 图形 （DA）入门工具包

PIC32MZ EC/EF 入门工具包上的跳线 JP2 应根据所使用的调试器 / 编程器进行如下连接：

- 如果使用 PKOB，则应短接引脚 1 和 3 以及引脚 2 和 4
- 如果使用 MPLAB REAL ICE™ 或 MPLAB ICD 3，则应将引脚 1 和 3 以及引脚 2 和 4 保留为开路

与 I2C 有关的连接如下：

- 将 MEB II J2 引脚 3 （SCL2）连接到外部 I2C 器件相应的 SCL 线
- 将 MEB II J2 引脚 5 （SDA2）连接到外部 I2C 器件相应的 SDA 线
- 如果连接了外部 I2C 器件，请确保主器件和从器件共用一个公共地。

运行演示

介绍如何编译并运行 I2C RTCC 演示。

说明

1. 该演示说明如何配置并使用 I2C 驱动程序 API 来支持中断模式下的 I2C 缓冲操作。在该演示中，I2C 配置为单个实例和单个客户端。

成功编译所选配置的演示应用程序之后，可以将固件编程到目标器件中。

要在调试模式下运行演示，请执行以下步骤：

1. 基于目标硬件，从 MPLAB X IDE 的 Project Properties （项目属性）中选择相应的配置。
2. 在 MPLAB X IDE 中，从 Hardware Tool （硬件工具）菜单中选择您的器件编程器。
3. 选择 *Debug （调试）* > *Debug Main Project （调试主项目）* 或在工具栏中单击 **Debug Main Project**。
4. 编译 MPLAB X IDE 项目中的选定配置，并对演示板编程。

I2C 驱动程序会将 I2C 外设的 I2C2 实例配置为主模式。SDA 和 SCL 线按配置硬件介绍的方式连接到 Microchip MCP7940N RTCC 器件。主器件会写入 RTCC 器件的 SRAM 存储器中的连续存储单元。然后，主器件会从同一页读回其内容。

可以通过检查缓冲区变量的内容来确定该操作的结果。

下表列出了预期的结果。

测试用例	缓冲区的内容
I2C2 （主器件）	RXbuffer_4[] = "3RTCCSLAVE" （从 RTCC 器件接收到的数据）

NVM 驱动程序演示

本主题介绍 NVM 驱动程序演示。

简介

本帮助文件包含关于 MPLAB Harmony NVM 驱动程序库应用程序演示 （包含在 MPLAB Harmony 库分发包中）的说明和关联信息。

说明

该应用程序演示 MPLAB Harmony NVM 驱动程序库的功能。本部分介绍硬件要求，以及在 Microchip 开发工具中编译并运行演示项目的过程。在该演示应用程序中，使用 NVM 驱动程序来访问 PIC32MX 和 PIC32MZ 器件的内部闪存，以执行写操作和读操作，并通过 LED 指示结果。

要了解关于 MPLAB Harmony NVM 驱动程序、配置 NVM 驱动程序和 NVM 驱动程序提供的 API 的更多信息，请参见 MPLAB Harmony NVM 驱动程序库文档。

演示

本主题介绍如何运行本发行版中包含的 NVM 驱动程序演示应用程序。

nvm_read_write

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 NVM 读 / 写演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `nvm_read_write.X project` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/nvm/nvm_read_write`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`nvm_read_write.X`	`<install-dir>/apps/driver/nvm/nvm_read_write/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	该配置的目的是使用 PIC32 USB 入门工具包 II 和动态 NVM 驱动程序实现在 PIC32MX 系列器件上执行演示。
pic32mx_usb_sk2_sta	pic32mx_usb_sk2	该配置的目的是使用 PIC32 USB 入门工具包 II 和静态 NVM 驱动程序实现在 PIC32MX 系列器件上执行演示。
pic32mz_ec_sk	pic32mz_ec_sk	该配置的目的是使用 PIC32MZ 嵌入式连接 （EC）入门工具包和动态 NVM 驱动程序实现在 PIC32MZ EC 系列器件上执行演示。
pic32mz_ec_sk_sta	pic32mz_ec_sk	该配置的目的是使用 PIC32MZ 嵌入式连接 （EC）入门工具包和静态 NVM 驱动程序实现在 PIC32MZ EC 系列器件上执行演示。
pic32mz_ef_sk	pic32mz_ef_sk	该配置的目的是使用 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包和动态 NVM 驱动程序实现在 PIC32MZ EF 系列器件上执行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何运行 NVM 驱动程序演示。

说明

这是一个简单的演示，说明如何配置并使用 NVM 驱动程序 API 来实现和访问 PIC32MX 和 PIC32MZ 器件的片内闪存。

如何运行该演示应用程序

成功编译演示应用程序之后，就可以在目标器件中烧写固件。

要在调试模式下运行演示，请执行以下步骤：

1. 基于目标硬件，从 MPLAB X IDE 的 Project Properties 中选择相应的配置。
2. 在 MPLAB X IDE 中，从 Hardware Tool 菜单中选择您的器件编程器。
3. 选择 *Debug > Debug Main Project* 或在工具栏中单击 **Debug Main Project**。
4. 编译 MPLAB X IDE 项目中的选定配置，并对演示板编程。

演示的执行状态 （通过 / 失败）通过演示板上的 LED 指示，如下表所示。

演示板	成功指示	失败指示
PIC32 USB 入门工具包 II	绿色 LED	红色 LED
PIC32MZ EC 入门工具包 PIC32MZ EF 入门工具包	绿色 LED	红色 LED

该演示应用程序使用 32 KB 的 NVM 存储器区域，从地址 DRV_NVM_MEDIA_START_ADDRESS 开始。

该应用程序会执行以下操作：

- 擦除全部的 32 KB 存储器，并通过读回数据来验证擦除操作
- 通过对写操作排队以在一页中顺序执行写操作。读回并验证数据。
- 重复步骤 1，擦除先前操作的所有数据
- 对分布在可用存储区域上的地址执行随机写操作。该操作演示驱动程序层的写操作排队。此外，它还演示使用驱动程序事件处理程序来跟踪已排队操作的完成情况。读回并验证数据。
- 重复步骤 1，擦除先前操作的所有数据
- 执行擦除写操作。该操作演示擦除写功能的用法

SPI 驱动程序演示

本主题介绍 SPI 驱动程序演示。

简介

本帮助文件包含关于 MPLAB Harmony SPI 驱动程序应用程序演示 （包含在 MPLAB Harmony 库分发包中）的说明和关联信息。

说明

该应用程序演示 MPLAB Harmony SPI 驱动程序的功能。本部分介绍硬件要求，以及使用 Microchip 开发工具编译并运行演示项目的过程。

提供了 3 个演示应用程序：

- serial_eeprom—— 在该演示应用程序中，使用 SPI 驱动程序来访问 Explorer 16 开发板中的外部 EEPROM，执行写操作和读操作，并通过 LED 指示结果
- spi_loopback—— 在该演示应用程序中，使用 SPI 驱动程序在同一器件上的 SPI 主模块和从模块之间传输数据，并通过 LED 指示结果
- spi_multislave—— 在该演示应用程序中，使用 SPI 驱动程序通过从选择（Slave Select，SS）功能在同一器件上的单个主模块和两个从模块之间传输数据，并通过 LED 指示结果

要了解关于 MPLAB Harmony SPI 驱动程序、配置 SPI 驱动程序和 SPI 驱动程序提供的 API 的更多信息，请参见 SPI 驱动程序库文档。

演示

本主题介绍如何运行本发行版中包含的 SPI 驱动程序演示应用程序。

serial_eeprom

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

在该演示应用程序中，使用 SPI 驱动程序来访问 Explorer 16 开发板中的外部 EEPROM，执行写操作和读操作，并通过 LED 指示结果。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了串行 EEPROM 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `serial_eeprom.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/spi/serial_eeprom`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`serial_eeprom.X`	`<install-dir>/apps/driver/spi/serial_eeprom/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx360_pim_e16	pic32mx795_pim+e16	该配置的目的是使用连接到Explorer 16开发板的PIC32MX360F512L PIM执行配置为中断模式和动态操作的演示。
pic32mx795_pim_e16	pic32mx795_pim+e16	该配置的目的是使用连接到Explorer 16开发板的PIC32MX795F512L PIM执行配置为中断模式和动态操作的演示。
pic32mx795_pim_e16_sta	pic32mx795_pim+e16	该配置的目的是使用连接到Explorer 16开发板的PIC32MX795F512L PIM执行配置为中断模式和静态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

Explorer 16 开发板与 PIC32MX795F512L PIM

- 在将 PIC32MX795F512L PIM 连接到 Explorer 16 开发板之前，请确保处理器选择开关 （S2）处于 PIM 位置
- 短接 Explorer 16 开发板上的 JP2，以使能 LED

Explorer 16 开发板与 PIC32MX360F512L PIM

- 在将 PIC32MX360F512L PIM 连接到 Explorer 16 开发板之前，请确保处理器选择开关 （S2）处于 PIM 位置
- 短接 Explorer 16 开发板上的 JP2，以使能 LED

运行演示

本部分演示如何运行 SPI 驱动程序串行 EEPROM 演示。

说明

该演示说明如何配置并使用 SPI 驱动程序 API 来实现和访问 Explorer 16 开发板的板载 EEPROM。

如何运行该演示应用程序

成功编译演示应用程序之后，就可以将固件烧写到目标器件中。

要在调试模式下运行演示，请执行以下步骤：

1. 在 MPLAB X IDE 中，从 *Project Properties > Hardware Tools* 中选择器件编程器。
2. 选择 *Debug > Debug Main Project* 或在工具栏中单击 **Debug Main Project**。

成功对器件编程之后，可以观察到 Explorer 16 开发板上的 LED 点亮。这说明演示项目运行成功。编程结果可以通过其他 LED 来了解。如果 LED D9 点亮，则表示 EEPROM 写 / 读功能失败。如果 LED D10 点亮，则表示 EEPROM 写 / 读功能通过。

spi_loopback

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

在该演示应用程序中，使用 SPI 驱动程序在同一器件上的 SPI 主模块和从模块之间传输数据，并通过 LED 指示结果。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SPI 环回演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `spi_loopback.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/spi/spi_loopback`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`spi_loopback.X`	`<install-dir>/apps/driver/spi/spi_loopback/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	配置为中断模式和动态操作，在 PIC32 USB 入门工具包 II 上演示 SPI 环回。
pic32mx_usb_sk2_int_sta	pic32mx_usb_sk2	配置为中断模式和静态操作，在 PIC32 USB 入门工具包 II 上演示 SPI 环回。
pic32mx_usb_sk2_poll_dyn	pic32mx_usb_sk2	配置为查询模式和动态操作，在 PIC32 USB 入门工具包 II 上演示 SPI 环回。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	配置为中断模式和动态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 SPI 环回。

pic32mz_ec_sk_int_dyn_dma	pic32mz_ec_sk	配置为中断模式和动态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上使用 DMA 演示 SPI 环回。
pic32mz_ec_sk_int_sta	pic32mz_ec_sk	配置为中断模式和静态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 SPI 环回。
pic32mz_ec_sk_poll_dyn	pic32mz_ec_sk	配置为查询模式和动态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 SPI 环回。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	配置为中断模式和动态操作，在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 SPI 环回。
pic32mz_ef_sk_poll_dyn	pic32mz_ef_sk	配置为查询模式和动态操作，在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 SPI 环回。
pic32mz_da_sk_int_dyn	pic32mz_da_sk	配置为中断模式和动态操作，在 PIC32MZ 图形（DA）入门工具包上演示 SPI 环回。
pic32mz_da_sk_poll_dyn	pic32mz_da_sk	配置为查询模式和动态操作，在 PIC32MZ 图形（DA）入门工具包上演示 SPI 环回。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包

该演示需要以下硬件：

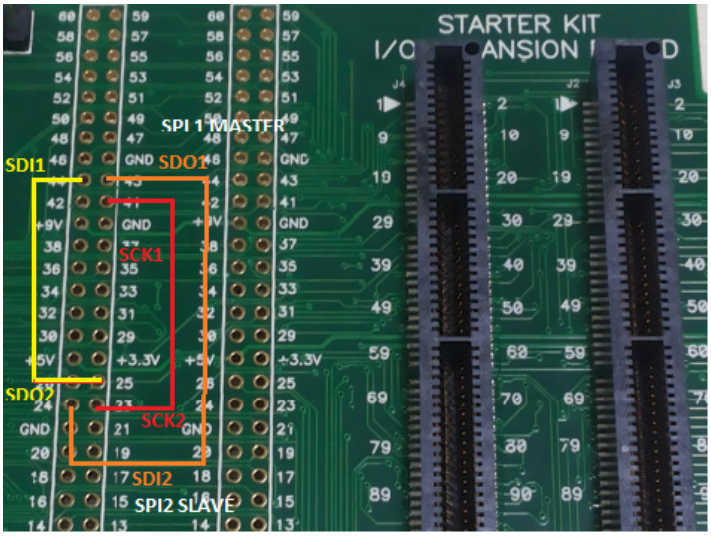
- PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包或 PIC32MZ DA 入门工具包
- PIC32MZ EC 适配器板
- 入门工具包 I/O 扩展板

根据所使用的入门工具包，该演示中将使用 SPI1 和 SPI2 模块或 SPI1 和 SPI3 模块。 PIC32MZ 器件支持外设引脚选择 （Peripheral Pin Select， PPS）功能。必须使用 PPS 来配置该器件上的 SPI 模块的 SPI 引脚。

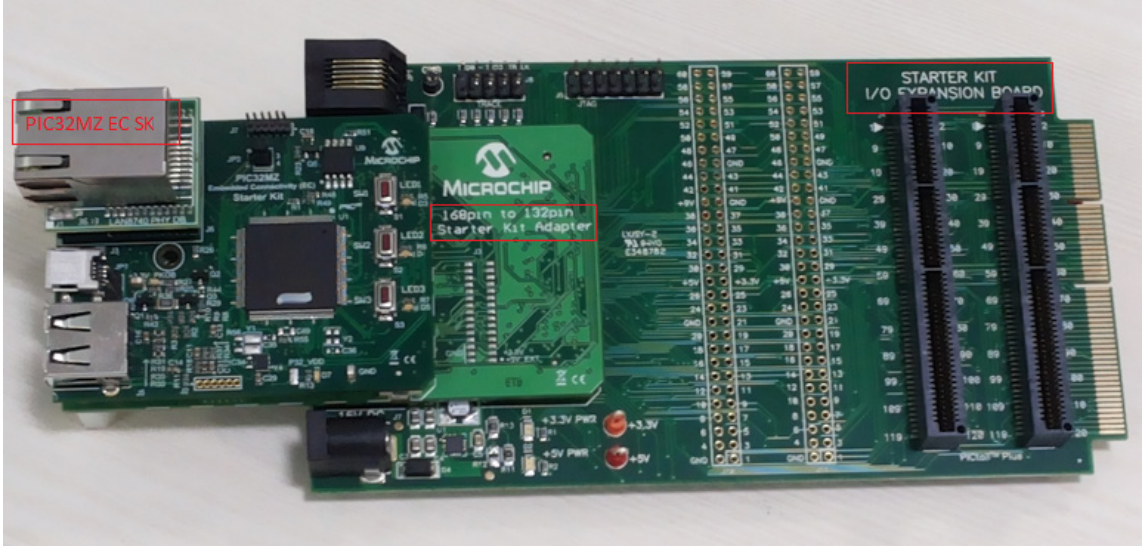
注： 要了解关于 PIC32MZ PPS 功能和引脚配置的更多信息，请参见具体器件数据手册的 **“I/O 端口”**章节中的**第 12.3 节“外设引脚选择 （PPS）”**。

PIC32MZ EC 入门工具包和 PIC32MZ EF 入门工具包配置：

1. （使用连线）连接入门工具包 I/O 扩展板上的引脚，如下图所示。该硬接线连接会将 PIC32MZ EC 入门工具包上的 PIC32MZ2048ECH144 器件或 PIC32MZ EF 入门工具包上的 PIC32MZ2048EFM144 器件的 SPI1 线连接到 SPI2 线。

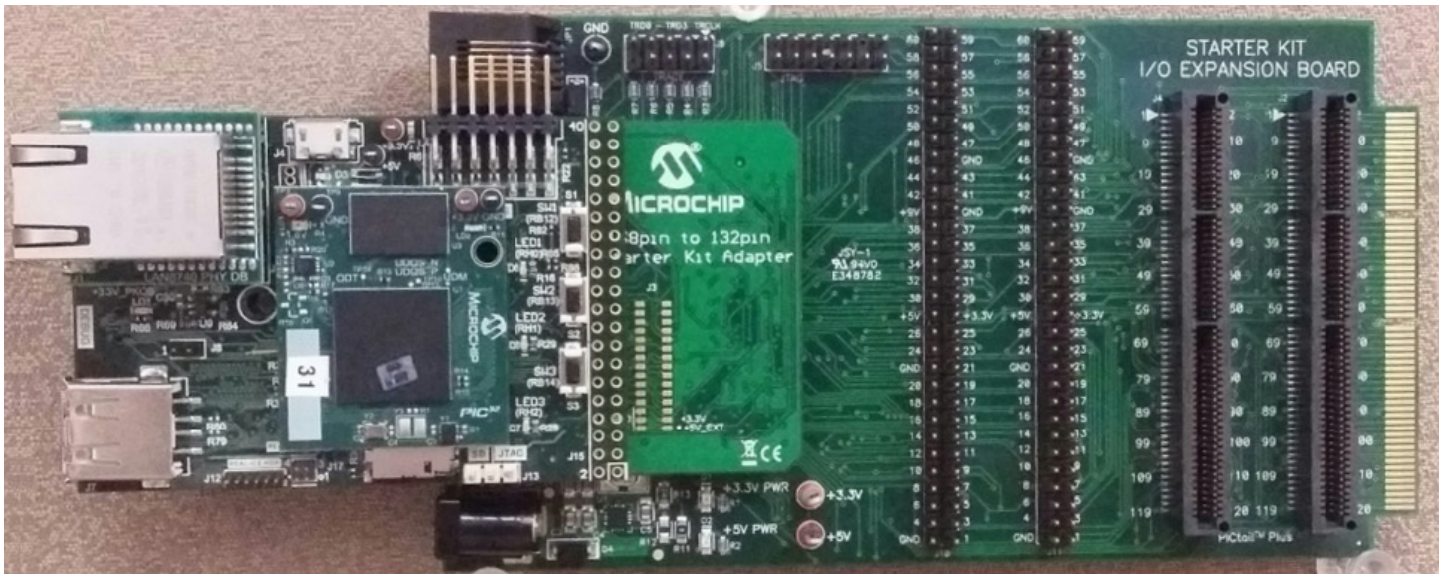


2. 接下来，连接所需的入门工具包、PIC32MZ EC 适配器板和入门工具包 I/O 扩展板，如下图所示（该示例中显示的是 PIC32MZ EC 入门工具包）。



PIC32MZ DA 入门工具包配置：

1. （使用连线）连接入门工具包 I/O 扩展板上的引脚，如下图所示。该硬接线连接会将 PIC32MZ DA 入门工具包上的 PIC32MZ2048DAB288 器件的 SPI1 线连接到 SPI3 线。
2. 接下来，连接所需的入门工具包、PIC32MZ DA 适配器板和入门工具包 I/O 扩展板，如下图所示（该示例中显示的是 PIC32MZ DA 入门工具包）。



对于该演示， PIC32MZ 端口引脚被映射为具有 SPI 功能。 PPS 映射和端口引脚模式配置为数字模式 （仅对于模拟引脚）源代码在 `system_init.c` 文件的 `SYS_Initialize` 函数中。

PIC32MZ EC 和 PIC32MZ EF 引脚映射

下表说明了 PIC32MZ EC 和 PIC32MZ EF 器件的 SPI1 和 SPI2 引脚如何进行映射并通过入门工具包 I/O 扩展板相互连接。

SPI 模块	SPI 线	PIC32MZ2048ECH144 或 PIC32MZ2048EFM144 器件引脚编号	模拟引脚	PIC32MZ2048ECH144 或 PIC32MZ2048EFM144 端口引脚名称 / 功能	I/O 扩展板 J10 连接器上的引脚编号	对于该演示，将该引脚连接到：
SPI1	SCK1	109	无	SCK1	41	SCK2
SPI1	SDI1	69	AN32	RPD14	44	SDO2
SPI1	SDO1	98	无	RPD10	43	SDI2
SPI1	/SS	未使用	未使用	未使用	未使用	未使用

SPI2	SCK2	14	AN14	SCK2	23	SCK1
SPI2	SDI2	121	无	RPD7	24	SDO1
SPI2	SDO2	25	AN45	RPB5	25	SDI1
SPI2	/SS	未使用	未使用	未使用	未使用	未使用

PIC32MZ DA 引脚映射

下表说明了 PIC32MZ DA 器件的 SPI1 和 SPI3 引脚如何进行映射并通过入门工具包 I/O 扩展板相互连接。

SPI 模块	SPI 线	PIC32MZ2064DAB288 器件引脚编号	模拟引脚	PIC32MZ2064DAB288 端口引脚名称 / 功能	I/O 扩展板 J10 连接器上的引脚编号	I/O 扩展板 J11 连接器上的引脚编号	对于该演示，将该引脚连接到：
SPI1	SCK1	M04	无	SCK1	41	N/A	SCK3
SPI1	SDI1	E18	AN5	RPD14	44	N/A	SDO3
SPI1	SDO1	J18	AN22	RPD10	N/A	52	SDI3
SPI1	/SS	未使用	未使用	未使用	未使用	未使用	未使用
SPI3	SCK3	J17	AN29	SCK3	N/A	21	SCK1
SPI3	SDI3	A12	AN12	RPC1	N/A	54	SDO1
SPI3	SDO3	A14	AN9	RPB1	37	N/A	SDI1
SPI3	/SS	未使用	未使用	未使用	未使用	未使用	未使用

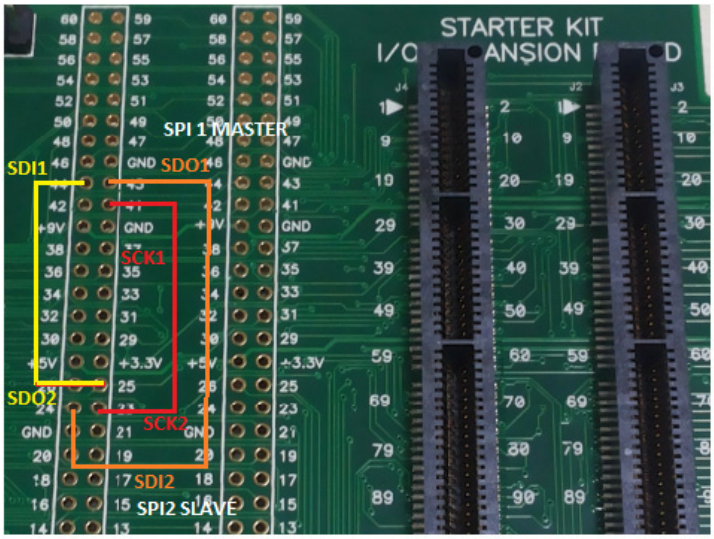
PIC32 USB 入门工具包 II

该演示需要以下硬件：

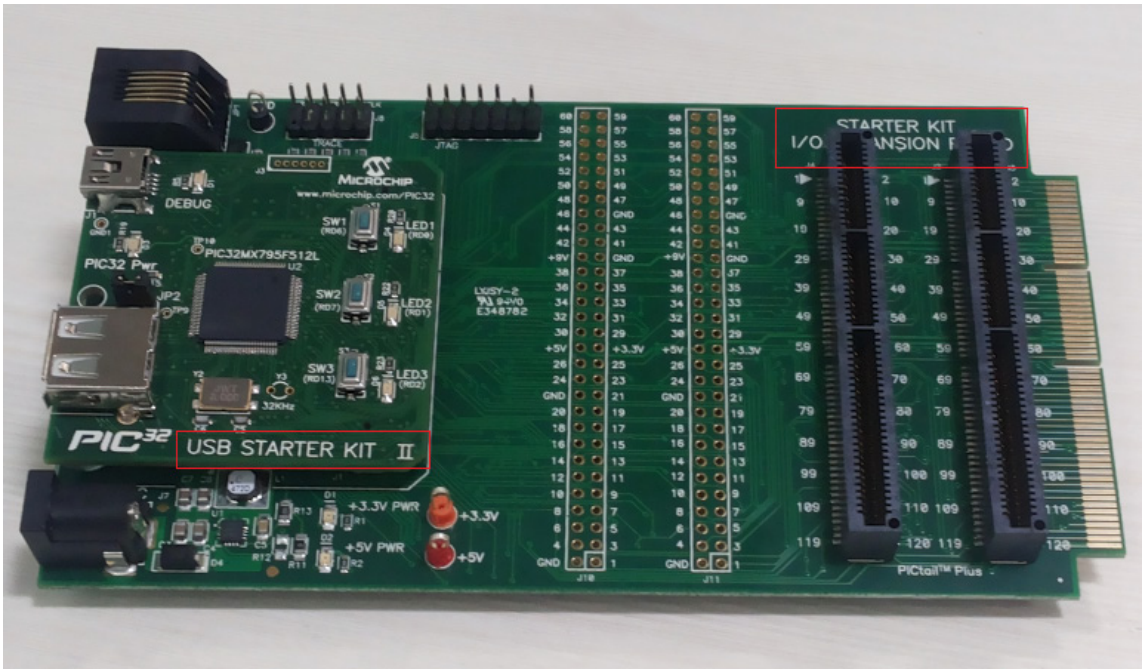
- PIC32 USB 入门工具包 II
- 入门工具包 I/O 扩展板

在该演示应用程序中，使用的是 SPI1 和 SPI2 模块。

1. （使用连线）连接入门工具包 I/O 扩展板上的引脚，如下图所示。该硬接线连接会将 PIC32 USB 入门工具包 II 上的 PIC32MX795F512L 器件的 SPI1 线通过入门工具包 I/O 扩展板连接到 SPI2 线。



2. 接下来，连接 PIC32 USB 入门工具包 II 和入门工具包 I/O 扩展板，如下图所示。



对于 PIC32MX795F512L 器件， SPI 模块具有专用 I/O 引脚。

下表说明了 SPI1 和 SPI2 引脚如何进行映射并通过入门工具包 I/O 扩展板相互连接。

SPI 模块	SPI 线	PIC32MX795F512L 器件引脚编号	PIC32MX795F512L 端口引脚名称 / 功能	I/O 扩展板 J10 连接器上的引脚编号	对于该演示，将该引脚连接到：
SPI1	SCK1	70	SCK1	41	SCK2
SPI1	SDI1	9	SDI1	44	SDO2
SPI1	SDO1	72	SDO1	43	SDI2
SPI1	/SS	未使用	未使用	未使用	未使用
SPI2	SCK2	10	SCK2	23	SCK1
SPI2	SDI2	11	SDI2	24	SDO1
SPI2	SDO2	12	SDO2	25	SDI1
SPI2	/SS	未使用	未使用	未使用	未使用

运行演示

本部分演示如何运行 SPI 驱动程序 SPI 环回演示。

说明

该演示说明如何配置并使用 SPI 驱动程序 API 来在中断模式和查询模式下支持面向多个驱动程序客户端的多个 SPI 硬件实例。该演示说明了 SPI 驱动程序的 “多实例多客户端” 功能。

成功编译演示应用程序之后，就可以将固件烧写到目标器件中。

要在调试模式下运行演示，请执行以下步骤：

1. 基于目标硬件，从 MPLAB X IDE 的 Project Properties 中选择相应的配置。
2. 在 MPLAB X IDE 中，从 Hardware Tool 菜单中选择您的器件编程器。
3. 选择 *Debug* > *Debug Main Project* 或在工具栏中单击 **Debug Main Project**。
4. 编译 MPLAB X IDE 项目中的选定配置，并对演示板编程。

SPI 驱动程序会将同一 PIC32 器件上的 SPI1 配置为主模块，将 SPI2 配置为从模块。SPI1 和 SPI2 线按配置硬件介绍的方式相互连接。主模块会向从模块发送一个数据块 （64 字节）。数据通过 SPI 在同一 PIC32 器件上发送和接收。数据会被环回给发送方。

在执行程序时， SPI 主模块 （SPI1）将使用应用程序中为 SPI1 定义的设置通过 SPI 通道发送一个字符串 / 数据序列。

从 SPI1 发送数据之后，驱动程序将从 SPI2 读取接收到的所有数据。如果接收到数据，程序将验证所接收的数据是否为主模块发送的数据，然后进入空闲模式。

在空闲模式下，固件既不发送也不接收任何数据。如果应用程序进入空闲模式，入门工具包的 LED2 （黄色）将点亮。

如果发生任何错误，或如果发送的数据与接收到的数据不相同，入门工具包的 LED1 （红色）将点亮，指示演示失败。

如果发送的数据与接收到的数据完全相同，入门工具包的 LED3 （绿色）将点亮，指示演示成功。

spi_multislave

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

在该演示应用程序中，使用 SPI 驱动程序来演示 SPI 协议的单主模块和多从模块功能，并使用 SPI 驱动程序的 `DRV_SPI_ClientConfigure` 函数在从模块之间切换。

在该演示中会进行两次数据传输，主模块会向每个从模块发送一个数据块。

从模块使用从选择 （/SSx）引脚选择，这意味着当一个从模块的 /SSx 引脚为低电平有效时，只有该从模块可以接收数据。因此，当一个从模块接收数据时，另一个从模块通过使该从模块的 /SSx 引脚变为高电平来保持空闲。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SPI 多从模块演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `spi_multislave.X` 项目，然后选择所需的配置。
以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/spi/spi_multislave`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`spi_multislave.X`	`<install-dir>/apps/driver/spi/spi_multislave/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	配置为中断模式和动态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 SPI 多从模块功能。
pic32mz_ec_sk_int_sta	pic32mz_ec_sk	配置为中断模式和静态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 SPI 多从模块功能。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	配置为中断模式和动态操作，在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 SPI 多从模块功能。

配置硬件

介绍如何配置支持的硬件。

说明

该演示需要以下硬件：

- PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包
- PIC32MZ EC 适配器板
- 入门工具包 I/O 扩展板

在该演示中，分别使用 SPI1、 SPI2 和 SPI3 模块作为主模块、从模块 1 和从模块 2。此外，还使用两个 GPIO 引脚来控制从模块选择（/SSx）引脚。

PIC32MZ 器件支持外设引脚选择 （PPS）功能。必须使用 PPS 来配置该器件上的 SPI 模块的 SPI 引脚。

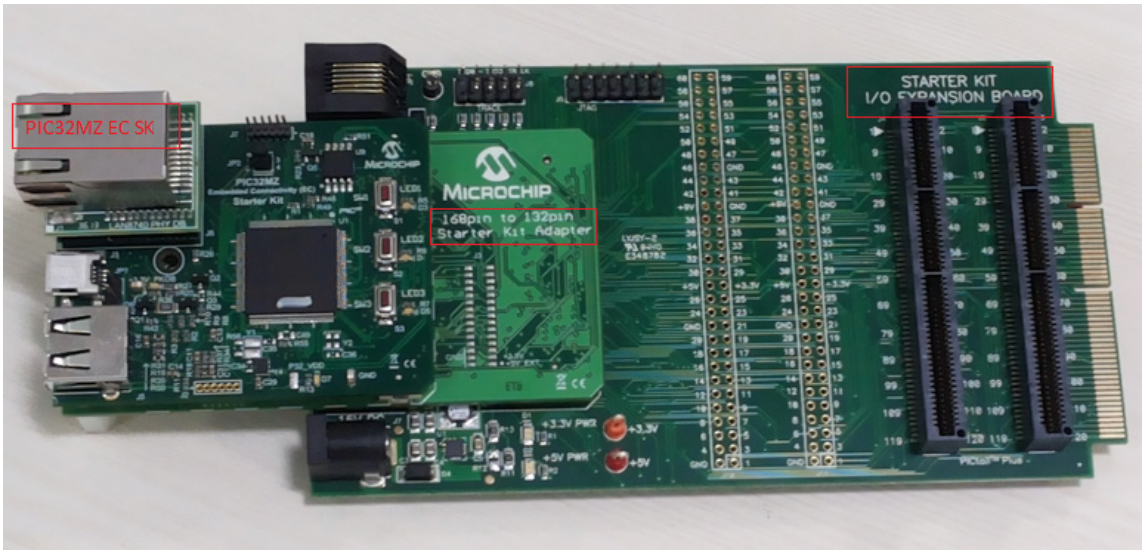
注： 要了解关于 PIC32MZ PPS 功能和引脚配置的更多信息，请参见具体器件数据手册的 **“I/O 端口”**章节中的**第 12.3 节“外设引脚选择 （PPS）”**。

PIC32MZ EC 入门工具包和 PIC32MZ EF 入门工具包配置：

1. 短接入门工具包 I/O 扩展板的 J10 和 J11 连接器上的引脚，如下：

- 引脚 41 （J10）、引脚 23 （J10）和引脚 48 （J11）：SCK1、 SCK2 和 SCK3
- 引脚 43 （J10）、引脚 24 （J10）和引脚 52 （J11）：SDO1、 SDI2 和 SDI3
- 引脚 44 （J10）、引脚 25 （J10）和引脚 37 （J11）：SDI1、 SDO2 和 SDO3
- 引脚 34 （J10）和引脚 26 （J10）：RH10 和 SS2
- 引脚 33 （J10）和引脚 46 （J11）：RH15 和 SS3

2. 接下来，连接所需的入门工具包、PIC32MZ EC 适配器板和入门工具包 I/O 扩展板，如下图所示（该示例中显示的是 PIC32MZ EC 入门工具包）。



PIC32MZ EC 和 PIC32MZ EF 引脚映射：

下表说明了 PIC32MZ EC 和 PIC32MZ EF 器件的 SPI1、 SPI2 和 SPI3 引脚如何进行映射并通过入门工具包 I/O 扩展板相互连接。

Module	Function	Device Pin #	Port Pin Name	Starter Kit I/O Expansion Board Pin #	Demonstration Setup Connection
SPI1	SCK1	109	SCK1	41 (J10)	SCK2, SCK3
	SDI1	69	RPD14	44 (J10)	SDO2, SDO3
	SDO1	98	RPD10	43 (J10)	SDI2, SDI3
	/SS1	Not Used	Not Used	Not Used	Not Used
SPI2	SCK2	14	SCK2	23 (J10)	SCK1
	SDI2	121	RPD7	24 (J10)	SDO1
	SDO2	25	RPB5	25 (J10)	SDI1
	/SS2	104	RPD0	26 (J10)	RH10
SPI3	SCK3	61	SCK3	48 (J11)	SCK1
	SDI3	49	RPB10	52 (J11)	SDO1
	SDO3	96	RPA15	37 (J11)	SDI1
	/SS3	62	RPB15	46 (J11)	RH15
GPIO (for Slave Select)	Output	83	RH10	34 (J10)	SS2
	Output	103	RH15	33 (J10)	SS3

运行演示

本部分演示如何运行 SPI 驱动程序 SPI 多从模块演示。

说明

成功编译演示应用程序之后，就可以将固件烧写到目标器件中。

要在调试模式下运行演示，请执行以下步骤：

1. 基于目标硬件，从 MPLAB X IDE 的 Project Properties 中选择相应的配置。
2. 在 MPLAB X IDE 中，从 Hardware Tool 菜单中选择您的器件编程器。
3. 选择 *Debug > Debug Main Project* 或在工具栏中单击 **Debug Main Project**。

成功对器件编程之后，可以观察到入门工具包上的 LED 点亮。这说明演示项目运行成功。编程结果可以通过 LED 来了解。

SPI 驱动程序会将同一 PIC32 器件上的 SPI1 配置为主模块，将 SPI2 和 SPI3 配置为从模块。 SPI1、SPI2 和 SPI3 线按配置硬件介绍的方式相互连接。

在第一次传输中，主模块会向从模块 1 发送一个数据块 （编号 0 至 63，共 64 字节），然后在第二次传输中，再向从模块 2 发送一个数据块 （编号 64 至 1，共 64 字节）。

在每次传输中，相应的从模块使用 /SS2 和 /SS3 引脚进行选择，它们分别由 GPIO 引脚 RH10 和 RH15 驱动。

下表说明了 SPI 从模块在应用程序不同阶段的状态：

Application State	RH10 State (Connected to /SS2)	RH15 State (Connected to /SS3)	Slave 1	Slave 2
Initialization	High	High	Inactive	Inactive
Transfer 1	Low	High	Active	Inactive
Transfer 2	High	Low	Inactive	Active
Idle	High	High	Inactive	Inactive

在两次传输都完成后，会使用发送的数据验证两个从模块接收到的数据：

1. 如果 SPI2 接收到的数据与第一次传输中发送的数据匹配， LED2 （黄色）会点亮
2. 如果 SPI3 接收到的数据与第二次传输中发送的数据匹配， LED3 （绿色）会点亮
3. 如果两次传输都未完成或任何一个从模块的数据不匹配， LED1 （红色）会点亮

SPI 闪存驱动程序演示

本主题介绍 SST25VF020B SPI 闪存驱动程序演示。

简介

本帮助文件包含关于 MPLAB Harmony SPI 闪存驱动程序应用程序演示 （包含在 MPLAB Harmony 库分发包中）的说明和关联信息。

说明

该应用程序演示 MPLAB Harmony SPI 闪存驱动程序的功能。本部分介绍硬件要求，以及使用 Microchip 开发工具编译并运行演示项目的过程。

演示

本主题介绍如何运行本发行版中包含的 SPI 闪存驱动程序演示应用程序。

sst25vf020b

该演示使用 SST25VF020B SPI 闪存驱动程序通过 SPI 来擦除、写入和读取板载 SST25VF020B 闪存，并验证操作是否正确执行。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SPI 闪存驱动程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `sst25vf020b.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/spi_flash/sst25vf020b`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`sst25vf020b.X`	`<install-dir>/apps/driver/spi_flash/sst25vf020b/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
bt_audio_dk_int_dyn	bt_audio_dk	该配置在 PIC32 蓝牙音频开发工具包上运行。SPI 驱动程序和 SST25VF20B SPI 闪存驱动程序配置为中断模式和动态操作。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 蓝牙音频开发工具包

- 确保开关 S1 设置到 PIC32_MCLR

运行演示

本部分介绍如何运行 SPI 闪存驱动程序演示。

说明

要运行该演示：

1. 首先编译并对目标器件编程。在编译时，选择适合于硬件的配置。
2. 观察开发工具包上的 LED 的状态。如果 LED 8 和 LED 9 点亮，则表示演示在正常工作。如果 LED 5 或 LED 6 点亮，则表示演示不在正常工作。

USART 驱动程序演示

本主题介绍 USART 驱动程序演示。

简介

本部分提供关于 MPLAB Harmony USART 驱动程序演示应用程序 （包含在 MPLAB Harmony 库分发包中）的说明和信息。

说明

该应用程序演示如何使用 MPLAB Harmony USART 驱动程序。本部分介绍硬件要求，以及在 Microchip 开发工具中编译并执行演示项目的过程。提供了两个演示：

- usart_echo—— 在该演示应用程序中，USART 驱动程序会在开始时发送数据字符串，然后接受接收到的任何字符，并回送数据。错误或成功状态通过 LED 指示。
- usart_loopback—— 在该演示应用程序中，使用了 USART 驱动程序 “多实例多客户端” 功能。该应用程序会使用同一 PIC32 器件的两个 USART 硬件实例。该应用程序会在 USART1 上发送数据块 （64 字节），在 USART2 上接收它，然后从 USART2 发送回 USART1。 USART1 会接收回先前发送的所有数据。数据会被环回到 USART1。

要了解关于 MPLAB Harmony USART 驱动程序、配置 USART 驱动程序和 USART 驱动程序提供的 API 的更多信息，请参见 USART 驱动程序库文档。

演示

本主题介绍如何运行本发行版中包含的 USART 驱动程序演示应用程序。

usart_echo

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USART 回送演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `usart_echo.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/usart/usart_echo`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`usart_echo.X`	`<install-dir>/apps/driver/usart/usart_echo/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16_int_dyn	pic32mx795_pim+e16	该演示在 PIC32MX795F512L PIM 和 Explorer 16 开发板上运行，并配置为中断模式和动态操作。
pic32mx795_pim_e16_int_sta	pic32mx795_pim+e16	该演示在 PIC32MX795F512L PIM 和 Explorer 16 开发板上运行，并配置为中断模式和静态操作。
pic32mx795_pim_e16_poll_dyn	pic32mx795_pim+e16	该演示在 PIC32MX795F512L PIM 和 Explorer 16 开发板上运行，并配置为查询模式和动态操作。
pic32mx795_pim_e16_poll_sta	pic32mx795_pim+e16	该演示在 PIC32MX795F512L PIM 和 Explorer 16 开发板上运行，并配置为查询模式和静态操作。

配置硬件

本部分介绍如何配置支持的硬件。

说明

Explorer 16 开发板

以下是执行该演示所需的硬件配置设置。

1. 在将 PIC32MX795F512L PIM 连接到 Explorer 16 开发板之前，请确保处理器选择开关 （S2）处于 PIM 位置。
2. 短接 Explorer 16 开发板上的 JP2，以使能 LED。

PIC32MX795F512L PIM

不需要硬件相关配置或跳线设置更改。

运行演示

本部分演示如何运行 USART 驱动程序演示。

说明

成功编译所需配置的演示应用程序之后，将固件烧写到目标器件中。

在执行时，应用程序会使用在应用程序中定义的设置通过 USART 发送以下字符串。

```
**************************************************
Welcome to Microchip USART Driver Demo Application.
Press any character, the character will be echoed back.
Press 'ESC' key to exit the Demo Application.
**************************************************
```

发送该文本之后，固件会读取通过 USART 接收到的任何字符，并且它会传送回相同的数据。如果接收到的字符为“ESC”（0x1B），程序会进入空闲模式。在空闲模式下，固件既不发送也不接收任何数据。如果应用程序进入空闲模式， Explorer 16 开发板的 LED 5 将点亮。如果发生任何错误， Explorer 16 开发板的 LED 9 将点亮。

usart_loopback

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USART 环回演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `usart_loopback.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/driver/usart/usart_loopback`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`usart_loopback.X`	`<install-dir>/apps/driver/usart/usart_loopback/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	配置为中断模式和动态操作，在 PIC32 USB 入门工具包 II 上演示 USART 环回。
pic32mx_usb_sk2_int_sta	pic32mx_usb_sk2	配置为中断模式和静态操作，在 PIC32 USB 入门工具包 II 上演示 USART 环回。
pic32mx_usb_sk2_poll_dyn	pic32mx_usb_sk2	配置为查询模式和动态操作，在 PIC32 USB 入门工具包 II 上演示 USART 环回。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	配置为中断模式和动态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 USART 环回。
pic32mz_ec_sk_int_sta	pic32mz_ec_sk	配置为中断模式和静态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 USART 环回。
pic32mz_ec_sk_poll_dyn	pic32mz_ec_sk	配置为查询模式和动态操作，在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 USART 环回。
pic32mz_ec_sk_int_dyn	pic32mz_ef_sk	配置为中断模式和动态操作，在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 USART 环回。
pic32mz_ef_sk_poll_dyn	pic32mz_ef_sk	配置为查询模式和动态操作，在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 USART 环回。
pic32mz_da_sk_int_dyn	pic32mz_da_sk	配置为中断模式和动态操作，在PIC32MZ图形（DA）入门工具包上演示USART环回。
pic32mz_da_sk_poll_dyn	pic32mz_da_sk	配置为查询模式和动态操作，在PIC32MZ图形（DA）入门工具包上演示USART环回。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包或 PIC32MZ DA 入门工具包

注： 考虑到篇幅的原因，仅显示了 PIC32MZ EC 入门工具包。但是，以下配置信息适用于两种入门工具包。

该演示需要以下硬件：

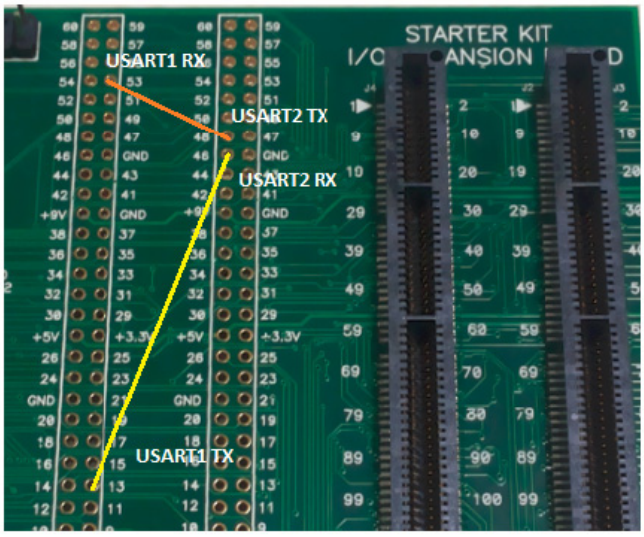
- PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包或 PIC32MZ DA 入门工具包
- PIC32MZ EC 适配器板
- 入门工具包 I/O 扩展板

PIC32MZ EC 和 PIC32MZ EF 入门工具包

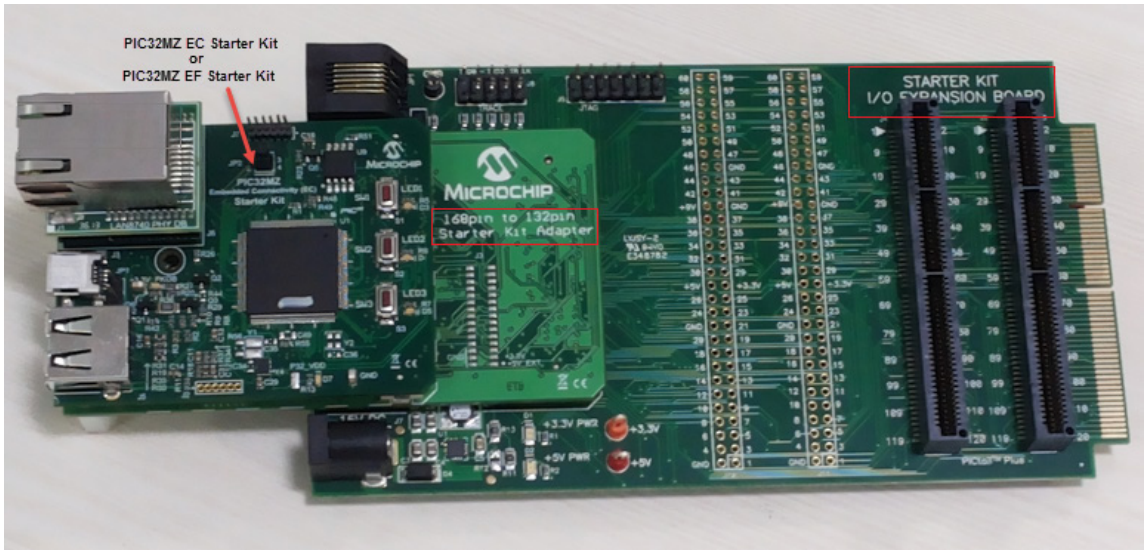
在该演示应用程序中，使用的是 USART1 和 USART2 模块。PIC32MZ 器件支持外设引脚选择（PPS）功能。需要使用 PPS 来配置该器件上的 USART 模块的 USART 引脚。

注： 要了解关于 PIC32MZ PPS 功能和引脚配置的更多信息，请参见具体器件数据手册的 **“I/O 端口”**章节中的**第 12.3 节“外设引脚选择（PPS）”**。这些文档可从 Microchip 网站（www.microchip.com）下载。

1. （使用连线）连接入门工具包 I/O 扩展板上的引脚，如下图所示。该硬接线连接会将 PIC32MZ EC 入门工具包上的 PIC32MZ2048ECH144 器件或 PIC32MZ EF 入门工具包上的 PIC32MZ2048EFM144 器件的 USART1 线连接到 USART2 线。

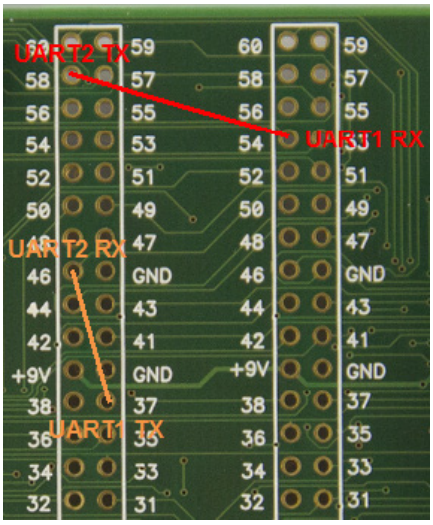


2. 接下来，连接 PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包、PIC32MZ EC 适配器板和入门工具包 I/O 扩展板，如下图所示。



PIC32MZ DA 入门工具包

1. （使用连线）连接入门工具包 I/O 扩展板上的引脚，如下图所示。该硬接线连接会将 PIC32MZ DA 入门工具包上的 PIC32MZ2064DAB288 器件的 USART1 线连接到 USART2 线。



2. 对于该演示， PIC32MZ 端口引脚被映射为具有 USART 功能。 PPS 映射和端口引脚模式配置为数字模式 （仅对于模拟引脚）源代码在 `system_init.c` 文件的 `SYS_Initialize` 函数中。

3. PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包

下表说明了 USART1 和 USART2 引脚如何进行映射并通过入门工具包 I/O 扩展板相互连接。

USART 模块	USART 线	PIC32MZ2048ECH144 或 PIC32MZ2048EFM 器件引脚编号	模拟引脚	PIC32MZ2048ECH144 或 PIC32MZ2048EFM 端口引脚名称 / 功能	I/O 扩展板 J10 连接器上的引脚编号	I/O 扩展板 J11 连接器上的引脚编号	对于该演示，将该引脚连接到：
USART1	TX	13	AN19	PMRD	13	N/A	USART2 RX
USART1	RX	48	AN49	PMA7	53	N/A	USART2 TX
USART2	TX	61	AN9	RPB14	N/A	48	USART1 RX
USART2	RX	62	AN10	RPB15	N/A	46	USART1 TX

PIC32MZ DA 入门工具包

下表说明了 USART1 和 USART2 引脚如何进行映射并通过入门工具包 I/O 扩展板相互连接。

USART 模块	USART 线	PIC32MZ2064DAB288 器件引脚编号	模拟引脚	PIC32MZ2064DAB288 端口引脚名称 / 功能	I/O 扩展板 J10 连接器上的引脚编号	I/O 扩展板 J11 连接器上的引脚编号	对于该演示，将该引脚连接到：
USART1	TX	A14	AN9	RPB1	37	N/A	USART2 RX
USART1	RX	A12	AN12	RPC1	54	N/A	USART2 TX
USART2	TX	B09	AN23	RPG9	N/A	58	USART1 RX
USART2	RX	B10	AN26	RPE09	N/A	46	USART1 TX

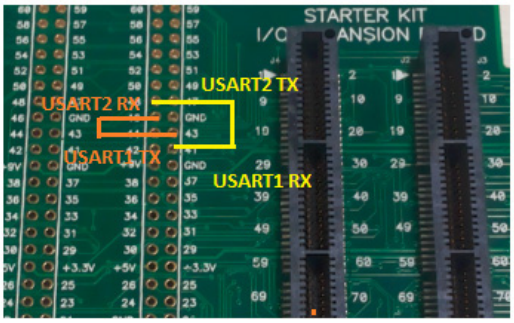
PIC32 USB 入门工具包 II

该演示需要以下硬件：

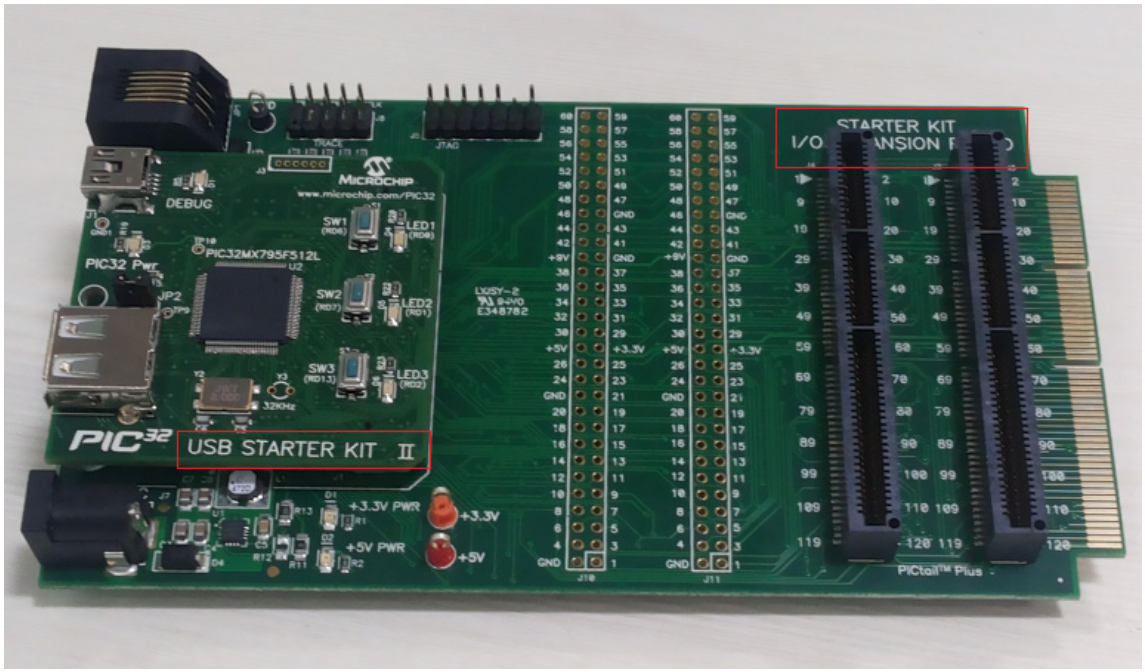
- PIC32 USB 入门工具包 II
- 入门工具包 I/O 扩展板

在该演示应用程序中，使用的是 USART1 和 USART2 模块。

1. （使用连线）连接入门工具包 I/O 扩展板上的引脚，如下图所示。该硬接线连接会将 PIC32 USB 入门工具包 II 上的 PIC32MX795F512L 器件的 USART1 线通过入门工具包 I/O 扩展板连接到 USART2 线。



2. 接下来，连接 PIC32 USB 入门工具包 II 和入门工具包 I/O 扩展板，如下图所示。



对于 PIC32MX795F512L 器件， USART 模块具有专用 I/O 引脚。

下表说明了 USART1 和 USART2 引脚如何进行映射并通过入门工具包 I/O 扩展板相互连接。

USART 模块	USART 线	PIC32MX795F512L 器件引脚编号	PIC32MX795F512L 端口引脚名称 / 功能	I/O 扩展板 J11 连接器上的引脚编号	对于该演示，将该引脚连接到：
USART1	TX	53	U1TX	43	U2RX
USART1	RX	52	U1RX	41	U2TX
USART2	TX	50	U2TX	48	U1RX
USART2	RX	49	U2RX	46	U1TX

运行演示

本部分演示如何运行 USART 驱动程序 USART 环回演示。

说明

该演示说明如何配置并使用 USART 驱动程序 API 来在中断模式和查询模式下支持面向多个驱动程序客户端的多个 USART 硬件实例。该演示说明了 USART 驱动程序的 “多实例多客户端” 功能。

成功编译演示应用程序之后，就可以将固件烧写到目标器件中。

要在调试模式下运行演示，请执行以下步骤：

1. 基于目标硬件，从 MPLAB X IDE 的 Project Properties 中选择相应的配置。

2. 在 MPLAB X IDE 中，从 Hardware Tool 菜单中选择您的器件编程器。
3. 选择 *Debug* > *Debug Main Project* 或在工具栏中单击 **Debug Main Project**。
4. 编译 MPLAB X IDE 项目中的选定配置，并对演示板编程。

USART 驱动程序会按 `system_config.h` 文件中定义的方式配置同一 PIC32 器件上的 USART1 和 USART2 模块。USART1 和 USART2 线按配置硬件介绍的方式相互连接。USART1 会向 USART2 发送一个数据块（64 字节）。USART2 会向 USART1 发送回相同的数据。数据通过另一个 USART 在同一 USART 模块上发送和接收。数据会被环回给发送方。

在执行程序时， USART1 将使用应用程序中为 USART1 定义的设置通过 USART1 发送通道发送一系列字符串 / 数据。

从 USART1 发送数据之后，驱动程序将从 USART2 读取接收到的所有数据。如果接收到数据，则向 USART1 发送回相同的数据。程序将验证所接收的数据是否为所发送的数据，然后进入空闲模式。

在空闲模式下，固件既不发送也不接收任何数据。如果应用程序进入空闲模式，入门工具包的 LED2 （黄色）将点亮。

如果发生任何错误，或如果发送的数据与接收到的数据不相同，入门工具包的 LED1 （红色）将点亮，指示演示失败。

如果在 UART1 上发送的数据与接收到的数据相同，入门工具包的 LED3 （绿色）将点亮，指示演示成功。

示例

提供关于 MPLAB Harmony 示例应用程序的信息。 MPLAB Harmony 示例应用程序提供了一些用途单一的简单应用程序项目，用于说明如何使用选定的 MPLAB Harmony 库。它们是在整个 MPLAB Harmony 文档中提供的源代码的有用示例。

my_first_app

介绍 `my_first_app` 示例。

说明

MPLAB Harmony `my_first_app` 示例应用程序提供了 Creating Your First Project （创建您的第一个项目）教程中介绍的解决方案。

外设库示例

本主题介绍外设库示例。

简介

MPLAB 外设库 （PLIB）的示例应用程序提供了一些非常简单的单一用途示例，用于说明如何使用 MPLAB Harmony 外设库。

说明

外设库 （PLIB）是随 MPLAB Harmony 提供的最低级库。它们提供 Microchip 单片机上可用外设的功能抽象，以便隐藏不同器件之间可能存在的寄存器细节差异。但是，它们不会在调用之间维护任何状态数据 （至少不会维护未存储在硬件寄存器中的任何状态数据），也不会对外设的资源提供任何保护。因此，调用某个外设的 PLIB 的所有代码都必须负责处理该外设的所有权，并负责管理该外设的行为。

因此， PLIB 通常仅由 MPLAB Harmony 器件驱动程序或系统服务使用。但是，有时需要直接从应用程序中与 PLIB 进行交互。因此，提供了一些简单的示例，说明如何使用外设库的接口。这些示例在 MPLAB Harmony 安装的以下位置提供：`<install-dir>/apps/examples/peripheral`。

PIC32MX 器件示例是针对 Explorer 16 开发板和 PIC32MX795F512L PIM 编写的。 PIC32MZ 器件示例是针对 PIC32MZ 嵌入式连接（EC）入门工具包编写的。这些示例经过测试，可以在 Explorer 16 开发板、PIC32 USB 入门工具包 II 和 PIC32MZ 嵌入式连接（EC）入门工具包上工作，每个项目相应的电路板配置可以在 MPLAB X IDE 中选择。

ADC 外设库示例

本主题介绍 ADC 外设库示例。

简介

ADC 外设库演示应用程序帮助

说明

该分发包包含一个与 ADC 相关的固件项目，用于演示 MPLAB Harmony ADC 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 ADC 外设库演示应用程序。

adc_pot

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例使用 PIC32MX795F512L PIM 来读取 Explorer 16 开发板上的电位器 R6。读取结果将显示在 LED 上。调节电位器时， LED 显示将向上或向下 “滑动”。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ADC 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adc_pot.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adc/adc_pot`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`adc_pot.X`	`<install-dir>/apps/examples/peripheral/adc/adc_pot/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	演示使用 ADC 读取 PIC32MX795F512L PIM 和 Explorer 16 开发板上的电位器 R6。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM、 Explorer 16 开发板和入门工具包 I/O 扩展板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 ADC 演示。

说明

该演示通过 ADC 读取电位器的设置。

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的可用配置。
2. 以任一方向旋转电位器 R6。检查 Explorer 16 开发板上的 LED 的状态。顺时针方向旋转电位器时，点亮的 LED 会增多。逆时针方向旋转电位器时，点亮的 LED 会减少。

adc_pot_dma

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例使用 ADC 读取 Explorer 16 开发板 （PIC32MX795F512L PIM）上的电位器 R6，并使用 DMA 在 RAM 中存储数据。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ADC 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adc_pot_dma.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adc/adc_pot_dma`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`adc_pot_dma.X`	`<install-dir>/apps/examples/peripheral/adc/adc_pot_dma/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	演示 ADC 使用 DMA 通道读取 PIC32MX795F512L PIM 和 Explorer 16 开发板上的电位器 R6。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM、 Explorer 16 开发板和入门工具包 I/O 扩展板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 ADC 演示。

说明

该演示通过 ADC 读取电位器的设置。

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的可用配置。
2. 以任一方向旋转电位器 R6。检查 Explorer 16 开发板上的 LED 的状态。顺时针方向旋转电位器时，点亮的 LED 会增多。逆时针方向旋转电位器时，点亮的 LED 会减少。

12 位高速 SAR ADC （ADCHS）外设库示例

本主题介绍 12 位高速 SAR ADC （ADCHS）外设库示例。

简介

12 位高速 SAR ADC （ADCHS）外设库演示应用程序帮助

说明

该分发包包含一个与 ADC 相关的固件项目，用于演示 MPLAB Harmony ADCHS 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference （中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 ADCHS 外设库演示应用程序。

adchs_3ch_dma

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例演示如何使用系统 DMA 来自动将 3 个 ADC 通道的转换数据存储到 3 个独立缓冲区中。每个通道设置为使用 2 Msps 的转换速率进行采样，并使用单个定时器触发信号来同步所有 3 个通道的转换。平均转换数据会显示在终端仿真器 （如 RealTerm）上。

该应用程序用于采用入门工具包 I/O 扩展板与适配器板的 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包，或用于 PIC32MZ 图形（DA）入门工具包。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ADCHS 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adchs_3ch_dma.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adchs/adchs_3ch_dma`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`adchs_3ch_dma.X`	`<install-dir>/apps/examples/peripheral/adchs/adchs_3ch_dma/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk	pic32mz_ef_sk	演示在 1 类模拟输入（电位器）使用入门工具包 I/O 扩展板和 PIC32MZ EC 入门工具包适配器板连接到 PIC32MZ EF 入门工具包时，如何操作 3 个系统 DMA 通道来存储该模拟输入的转换数据。
pic32mz_da_sk	pic32mz_da_sk	演示在 1 类模拟输入连接到 PIC32MZ DA 入门工具包时，如何操作 3 个系统 DMA 通道来存储该模拟输入的转换数据。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包

1. 将 PIC32MZ EF 入门工具包连接到 PIC32MZ EC 入门工具包适配器板。
2. 在入门工具包 I/O 扩展板上连接两个板的组合 （如步骤 1 中所述）。
3. 连接一个模拟信号或具有合适值 （4.7K 或 10K）的电位器，两端分别连接到 3.3V 和 J11 的 GND 引脚；电位器的抽头连接到入门工具包 I/O 扩展板上的 J11 引脚 34 和 33 （AN0 和 AN1）和 J10 引脚 32 （AN2）。
4. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ EF 入门工具包上的 mini-USB 连接器 J11。
5. 使用 9V 电源为入门工具包 I/O 扩展板供电。

PIC32MZ 图形 （DA）入门工具包

1. 连接 3 个模拟信号或具有合适值 （4.7K 或 10K）的 3 个电位器，两端分别连接到 3.3V 和 J11 的 GND 引脚；电位器的抽头连接到 PIC32MZ EF 入门工具包的 J15 引脚 16、 36 和 13 （AN2、 AN3 和 AN4）。
2. 通过使用 mini-USB 电缆从 PC 连接到 PIC32MZ EF 入门工具包上的 mini-USB 连接器 J19 （DEGUG），为入门工具包供电并将固件下载到入门工具包中。
3. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ EF 入门工具包上的 mini-USB 连接器 J11。

运行演示

介绍如何编译并运行 ADCHS 外设库演示。

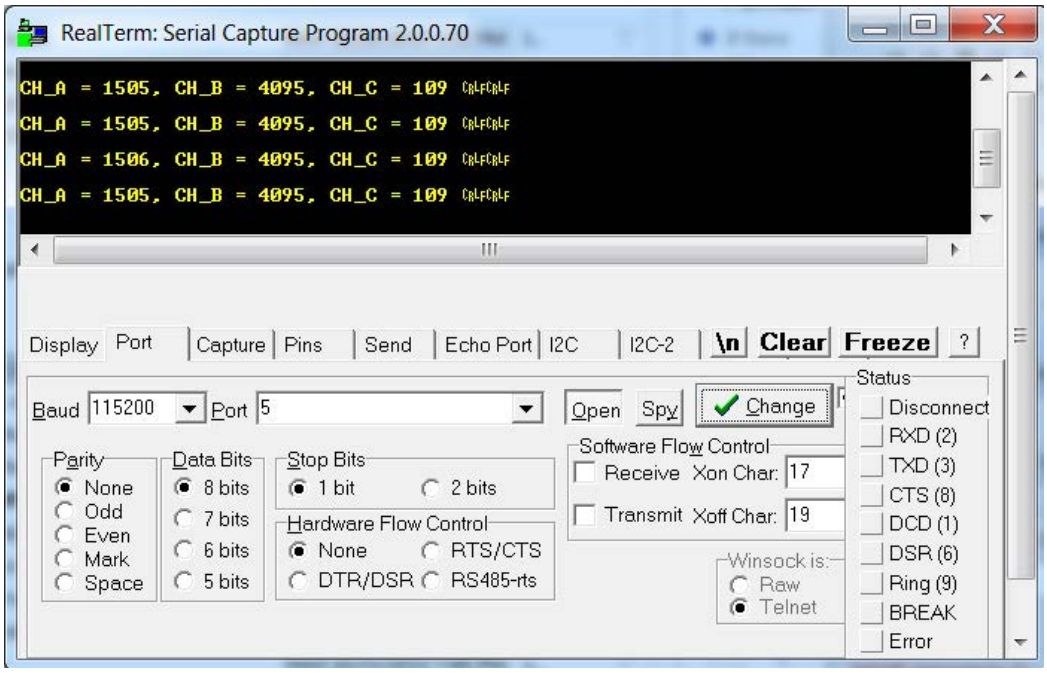
说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adchs_3ch_dma.X` 项目，然后选择所需的配置。

执行以下操作来运行演示：

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的可用配置。
2. 启动终端仿真器 （RealTerm/Tera Term 等）并选择适当的 COM 端口，并将串行端口设置为 115200-N-1。
3. 启动演示。控制台窗口中将显示以下消息：Result: 3171。
4. 该示例演示如何使用系统 DMA 来自动将 3 个 ADC 通道的转换数据存储到 3 个独立缓冲区中。每个通道设置为使用 2 Msps 的转换速率进行采样，并使用单个定时器触发信号来同步所有 3 个通道的转换。平均转换数据会显示在终端仿真器 （如 RealTerm）上。

下图显示了正在工作的终端仿真器窗口。



adchs_oversample

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例演示如何读取使用入门工具包 I/O 扩展板与 PIC32MZ EC 入门工具包适配器板连接到 PIC32MZ DA 入门工具包或 PIC32MZ EF 入门工具包的电位器，执行转换数据过采样，并在终端仿真器 （如 RealTerm）上显示更高分辨率的 ADC 转换读数。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ADCHS 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adchs_oversample.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adchs/adchs_oversample`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`adchs_oversample.X`	`<install-dir>/apps/examples/peripheral/adchs/adchs_oversample/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk	pic32mz_ef_sk	演示在 1 类模拟输入 （电位器）使用入门工具包 I/O 扩展板和 PIC32MZ EC 入门工具包适配器板连接到 PIC32MZ EF 入门工具包对其进行转换时，如何在过采样模式下执行数字滤波器操作。
pic32mz_da_sk	pic32mz_da_sk	演示在 1 类模拟输入 （电位器）连接到 PIC32MZ DA 入门工具包对其进行转换时，如何在过采样模式下执行数字滤波器操作。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包

1. 将 PIC32MZ EF 入门工具包连接到 PIC32MZ EC 入门工具包适配器板。
2. 在入门工具包 I/O 扩展板上连接两个板的组合 （如步骤 1 中所述）。
3. 连接具有合适值 （4.7K 或 10K）的电位器，两端分别连接到 3.3V 和 J11 的 GND 引脚；电位器的抽头连接到入门工具包 I/O 扩展板上的 J11 引脚 34。
4. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ EF 入门工具包上的 mini-USB 连接器 J11。
5. 使用 9V 电源为入门工具包 I/O 扩展板供电。

PIC32MZ 图形 （DA）入门工具包

1. 连接具有合适值 （4.7K 或 10K）的电位器，两端分别连接到 3.3V 和 J11 的 GND 引脚；电位器的抽头连接到 PIC32MZ DA 入门工具包上的 J15 引脚 36。
2. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ DA 入门工具包上的 mini-USB 连接器 J11。
3. 使用 mini-USB 电缆从 PC 连接到 DEBUG mini-USB 连接器 J19，以便为入门工具包供电。

运行演示

介绍如何编译并运行 ADCHS 外设库演示。

说明

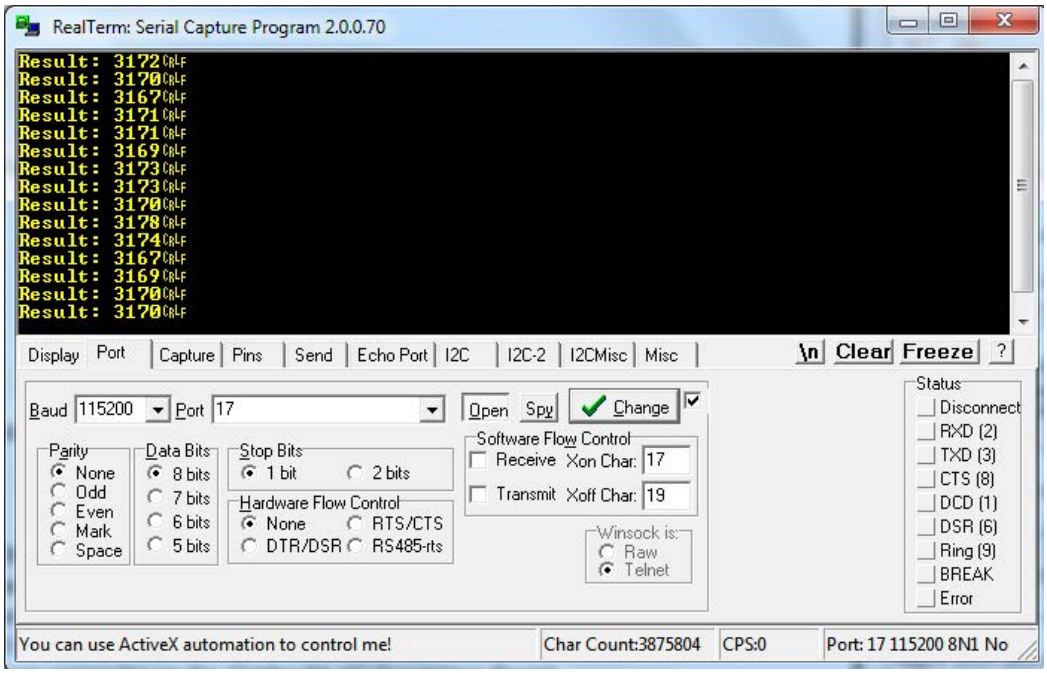
该演示使用数字滤波器对电位器的转换数据进行过采样。执行以下操作来运行演示：

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的可用配置。
2. 启动终端仿真器 （RealTerm/Tera Term 等）并选择适当的 COM 端口，并将串行端口设置为 115200-N-1。
3. 启动演示。控制台窗口中将显示以下消息：Result: 3171。
4. 旋转电位器时，控制台上的 “Result:”会显示高分辨率 ADC 转换值。
5. 更高分辨率的数据可以通过这种方式验证：先运行 adchs_pot 演示并观察 12 位转换数据；之后，电位器设置保持不变，运行 adchs_oversample 演示，可以观察并验证更高分辨率的 15 位转换数据。理论值计算如下：
   - 12 位分辨率的最大计数：4096
   - 15 位分辨率的最大计数：32768
   - ADC 参考输入电压：3.3 V

假设模拟输入为 0.319 V：

- 12 位分辨率 ADC 读数 （通过 adchs_pot 获得）：((4096/3.3) * 0.319) = 396
- 15 位分辨率 ADC 读数 （通过 adchs_oversample 过采样）：((32768/3.3) * 0.319) = 3167

下图显示了正在工作的终端仿真器窗口。



adchs_pot

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例演示如何读取使用入门工具包 I/O 扩展板与 PIC32MZ EC 入门工具包适配器板连接到 PIC32MZ DA 入门工具包或 PIC32MZ EF 入门工具包的电位器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ADCHS 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adchs_pot.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adchs/adchs_pot`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
adchs_pot.X	<install-dir>/apps/examples/peripheral/adchs/adchs_pot/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk	pic32mz_ef_sk	演示在电位器使用入门工具包 I/O 扩展板和 PIC32MZ EC 入门工具包适配器板连接到 PIC32MZ EF 入门工具包时，如何通过 ADC 读取电位器。
pic32mz_da_sk	pic32mz_da_sk	演示在电位器连接到 PIC32MZ DA 入门工具包时，如何通过 ADC 读取电位器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包

1. 将 PIC32MZ EF 入门工具包连接到 PIC32MZ EC 入门工具包适配器板。
2. 在入门工具包 I/O 扩展板上连接两个板的组合 （如步骤 1 中所述）。
3. 连接具有合适值 （4.7K 或 10K）的电位器，两端分别连接到 3.3V 和 J11 的 GND 引脚；电位器的抽头连接到入门工具包 I/O 扩展板上的 J11 引脚 34。
4. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ EF 入门工具包上的 mini-USB 连接器 J11。
5. 使用 9V 电源为入门工具包 I/O 扩展板供电。

PIC32MZ 图形 （DA）入门工具包

1. 连接具有合适值 （4.7K 或 10K）的电位器，两端分别连接到 3.3V 和 J11 的 GND 引脚；电位器的抽头连接到 PIC32MZ DA 入门工具包上的 J15 引脚 36。
2. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ DA 入门工具包上的 mini-USB 连接器 J11。
3. 使用 mini-USB 电缆从 PC 连接到 DEBUG mini-USB 连接器 J19，以便为入门工具包供电。

运行演示

介绍如何编译并运行 ADCHS 外设库演示。

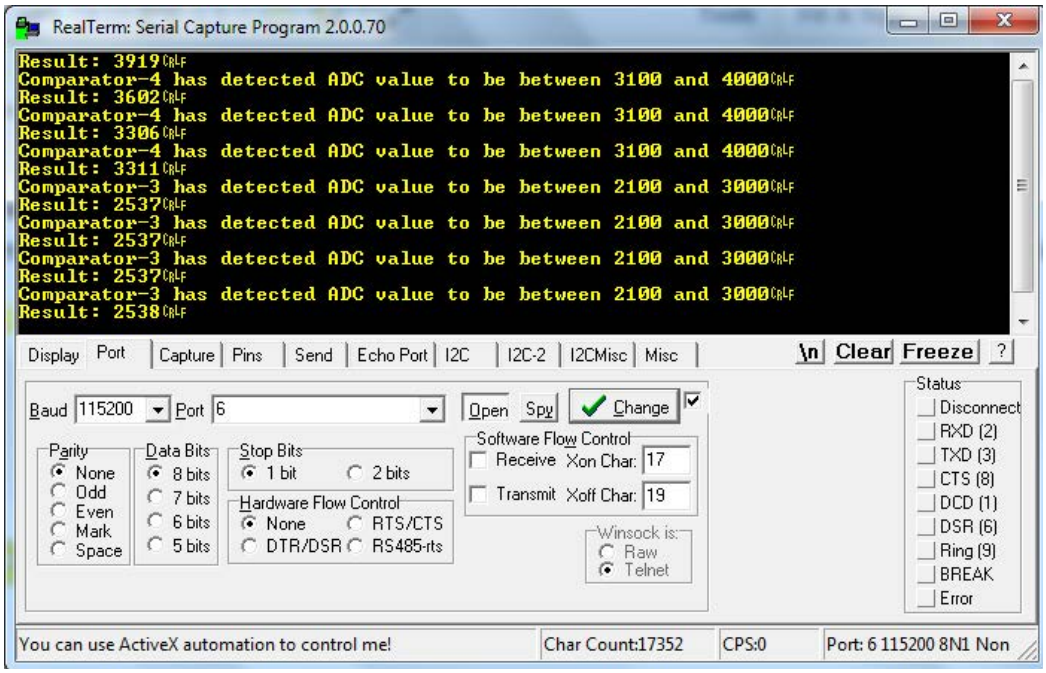
说明

该演示通过 ADC 读取电位器的设置。

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的可用配置。
2. 启动终端仿真器 （RealTerm/Tera Term 等）并选择适当的 COM 端口，并将串行端口设置为 115200-N-1。
3. 启动演示。控制台窗口中将显示以下消息：
   - Comparator 1 has detected the ADC value to be between 0 and 1000
   - Result: 960
4. 旋转电位器时，控制台上的“Result:”会显示 ADC 转换值。有 4 个数字比较器用于处理 ADC 转换值并产生事件。数字比较器事件的产生条件如下：
   - 数字比较器 1 在 ADC 转换值介于 0 至 1000 之间时产生事件。入门工具包上的所有 LED 保持熄灭。

- 数字比较器 2 在 ADC 转换值介于 1100 至 2000 之间时产生事件。此外，入门工具包上的 LED D1 点亮。
- 数字比较器 3 在 ADC 转换值介于 2100 至 3000 之间时产生事件。此外，入门工具包上的 LED D1 和 D2 点亮。
- 数字比较器 4 在 ADC 转换值介于 3100 至 4000 之间时产生事件。此外，入门工具包上的所有 LED 点亮。

5. 下图显示了正在工作的终端仿真器窗口。



adchs_sensor

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该外设库示例演示在通道扫描模式下 （通过 Timer3 匹配事件触发）转换 3 个 2 类和 1 个 3 类模拟输入。在 MEB II 板上，所选的 3 个 2 类模拟输入连接到加速度计的 3 个轴，3 类模拟输入连接到温度传感器。演示应用程序代码使用加速度计 3 个轴的转换值，将它们转换为 3 个轴的倾斜度，并在串行端口终端上显示结果（以弧度为单位）。此外，来自温度传感器的转换数据会按比例转换为摄氏度，并显示在串行端口终端上。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ADCHS 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adchs_sensor.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adchs/adchs_sensor`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`adchs_sensor.X`	`<install-dir>/apps/examples/peripheral/adchs/adchs_sensor/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	演示在将模拟输入连接到 MEB II 上的 PIC32MZ EF 入门工具包时，通过定时器触发的扫描转换功能，转换 3 个 2 类模拟输入（加速度计）和 1 个 3 类模拟输入（温度传感器）。
pic32mz_da_sk_meb2	pic32mz_da_sk+meb2	演示在将模拟输入连接到 MEB II 上的 PIC32MZ DA 入门工具包时，通过定时器触发的扫描转换功能，转换 3 个 2 类模拟输入（加速度计）和 1 个 3 类模拟输入（温度传感器）。

配置硬件

介绍如何配置支持的硬件。

说明

连接到 MEB II 的 PIC32MZ EF 入门工具包

1. 将 PIC32MZ EF 入门工具包安装到 MEB II 上。
2. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ EF 入门工具包上的 mini-USB 连接器 J11。
3. 使用 MEB II 上的 9V 电源或通过 SK 上的调试连接器 （J3）为入门工具包 I/O 扩展板供电。

注： 加速度计 y 和 z 轴连接到 PIC32MZ EF 入门工具包上的 AN7 （RB12）和 AN8 （RB13）。这两个引脚也用于开关输入 SW1 和 SW2。在应用程序初始化期间，端口会被配置为模拟端口。在运行该演示时，请小心不要按下 SW1 或 SW2。否则，转换后的 ADC 值将不正确。

连接到 MEB II 的 PIC32MZ DA 入门工具包

1. 将 PIC32MZ DA 入门工具包安装到 MEB II 上。
2. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ DA 入门工具包上的 mini-USB 连接器 J11。
3. 使用 MEB II 上的 9V 电源或通过 SK 上的调试连接器 （J3）为入门工具包 I/O 扩展板供电。

运行演示

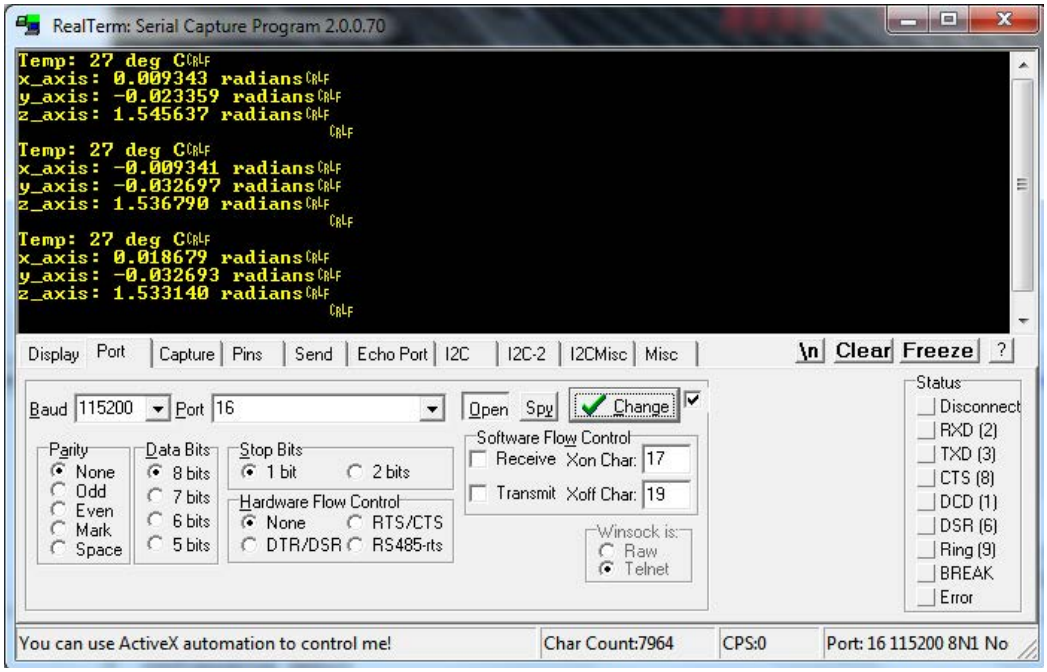
介绍如何编译并运行 ADCHS 外设库演示。

说明

该演示会读取加速度计和温度传感器。执行以下操作来运行演示：

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的可用配置。
2. 启动终端仿真器 （RealTerm/Tera Term 等）并选择适当的 COM 端口，并将串行端口设置为 115200-N-1。
3. 启动演示。控制台窗口中将显示以下消息：
   - Temp: 27 deg C
   - x-axis: 0.001 radians
   - y-axis: 0.01 radians
   - z-axis: 1.52 radians
4. 当 MEB II 板倾斜时，在控制台窗口上显示的每个轴的角度会发生变化。
5. 如果 （通过触摸）加热温度传感器 “U8”，则控制台窗口上显示的温度会发生变化。

下图显示了正在工作的终端仿真器窗口。



adchs_touchsense

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该外设库示例使用 PIC32MZ EF 入门工具包和多媒体扩展板 II （MEB II）上的触摸板 （B2）来演示 ADCHS 外设库的电容分压器（Capacitive Voltage Divider， CVD）功能和模拟通道扫描功能。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ADCHS 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adchs_touchsense.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adchs/adchs_touchsense`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`adchs_touchsense.X`	`<install-dir>/apps/examples/peripheral/adchs/adchs_touchsense/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	演示在模拟输入连接到 MEB II 上的 PIC32MZ EF 入门工具包时，ADCHS 的 CVD 功能，同时在扫描模式下转换 3 类模拟输入 （AN28）。

pic32mz_da_sk_meb2	pic32mz_da_sk+meb2	演示在模拟输入连接到 MEB II 上的 PIC32MZ DA 入门工具包时，ADCHS 的 CVD 功能，同时在扫描模式下转换 3 类模拟输入 （AN28）。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包和 MEB II

1. 将 PIC32MZ EF 入门工具包安装到 MEB II 上。
2. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ EF 入门工具包上的 mini-USB 连接器 J11。
3. 使用 MEB II 上的 9V 电源或通过 PIC32MZ EF 入门工具包上的调试连接器 （J3）为入门工具包 I/O 扩展板供电。

PIC32MZ DA 入门工具包和 MEB II

1. 将 PIC32MZ DA 入门工具包安装到 MEB II 上。
2. 使用 mini-USB 电缆从 PC 连接到 PIC32MZ DA 入门工具包上的 mini-USB 连接器 J11。
3. 使用 MEB II 上的 9V 电源或通过 PIC32MZ DA 入门工具包上的调试连接器 （J3）为入门工具包 I/O 扩展板供电。

运行演示

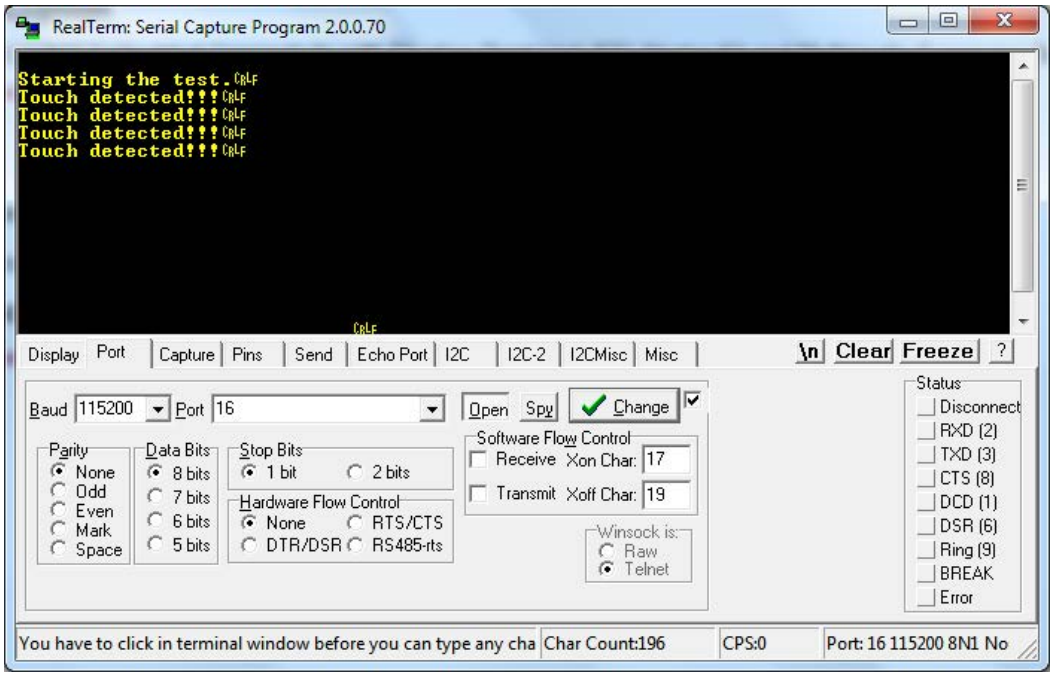
介绍如何编译并运行 ADCHS 外设库演示。

说明

该示例演示 ADCHS 的 CVD 功能，并检测 MEB II 上的 B2 触摸板的触摸事件。

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的可用配置。
2. 启动终端仿真器 （RealTerm/Tera Term 等）并选择适当的 COM 端口，并将串行端口设置为 115200-N-1。
3. 启动演示。
4. 触摸 MEB II 上的 B2 触摸板。B2 触摸板位于 MEB II 的 LCD 一侧，不是位于 MEB II 的入门工具包一侧。在被触摸时，MEB II 上的 LED D4 会点亮，串行终端程序上会显示“Touch Detected!!!”（检测到触摸 !!!）。该消息每秒重复一次，直到 B2 不再被触摸为止。
5. B2 不再被触摸时， LED D4 会熄灭，并在经过短暂的延时后， LED D3 点亮。

下图显示了正在工作的终端仿真器窗口。



流水线 ADC （ADCP）外设库示例

本主题介绍流水线模数转换器 （Pipelined Analog-to-Digital Converter， ADCP）外设库示例。

简介

流水线 ADC 外设库演示应用程序帮助

说明

该分发包包含一个与 ADC 相关的固件项目，用于演示 MPLAB Harmony 流水线 ADC 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的流水线 ADC 外设库演示应用程序。

adcp_cal

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例说明在使用流水线 ADC 进行正常操作之前如何配置它。此外，它还说明了所需的通道勘误项设置、过采样，以及使用 DMA 传输数据，以达到最佳效果。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了流水线 ADC 校准演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `adcp_cal.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/adcp/adcp_cal`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`adcp_cal.X`	`<install-dir>/apps/examples/peripheral/adcp/adcp_cal/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_pim_e16	pic32mz_ec_pim+e16	该演示使用连接到 Explorer 16 开发板的 PIC32MZ2048ECH100 接插模块（PIM），说明了在使用流水线 ADC 前需要执行的软件校准步骤。该校准通过对与 ADC 连接的内部参考电压（Internal Voltage Reference，IVREF）进行采样来计算直流失调。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ2048ECH100 PIM、 Explorer 16 开发板和入门工具包 I/O 扩展板

软件配置为使用分别与 PIM 的引脚 33 和 32 连接的外部 VREF+ 和 VREF-。这可以通过入门工具包 I/O 扩展板或 PICtail™ 原型设计子板来实现。

VREF+ 引脚电压需要在施加主 VDD 电源之时或之后施加。软件设置为接受 3.3V 作为 VREF+ 电压。软件设置为接受 0.0V 作为 VREF- 电压，从而可以将该引脚连接到 GND 或 AVSS。

如果对 VREF+ 或 VREF- 使用其他电压，则需要在软件中 （在 `adcp.h` 中）更新相应的设置，才能正确进行计算。

可以使用 Explorer 16 开发板上的电位器 R6 来驱动演示中的输入之一。在插入 PICtail 插槽的 PICtail 原型设计子板的 RB5 和 RG8 之间接上跳线。该子板也可以用于设置 VREF+ 和 VREF- 连接。

VREF+ 可以通过将 PICtail 原型设计子板上的 RB9 连接到 +3.3V 来设置。VREF- 可以通过连接 PICtail 原型设计子板上的 RB8 来设置。

运行演示

介绍如何编译并运行流水线 ADC 演示。

说明

该演示说明如何根据勘误项和数据手册说明使流水线 ADC 模块在 PIC32MZ 器件上工作。

1. 连接并使用调试器对器件编程。
2. 在 `app.c` 中第 234 行附近设置断点，该行内容为：
   `appData.state = APP_STATE_NORMALIZE_DATA;`
3. 在 `app.c` 中第 241 行附近设置断点，该行内容为：
   `appData.state = APP_STATE_DISPLAY_DATA;`
4. 运行程序。
5. 当程序到达第一个断点时，可以使用调试器变量窗口观察 `appData.ADC_Data1` 中的数据。
6. 从断点处继续运行程序。
7. 当程序到达第二个断点时，可以使用调试器变量窗口观察 `appData.ADC_Data1` 中的数据。这些数据现在已被归一化，意味着值已被转换为 8 位结果。它将作为应用程序使用的数据。
8. 继续运行程序来采集一组新的数据，根据需要重复步骤 5 至 7。如果您已连接跳线，使电位器 R6 可以工作，则旋转电位器并观察结果的变化。
9. 与电位器连接的通道上的读数大小也会反映在 Explorer 16 开发板的 LED 上。

注： Timer3 用于自动触发 ADC 扫描。如果要使用的通道数量发生变化，需要相应地调整 Timer3 周期。

执行以下操作进行调整：

1. 打开 Microchip Harmony 配置器。
2. 装入 adcp_cal 项目的配置。
3. 浏览至 *Harmony Framework Configuration （Harmony 框架配置）* > *Drivers （驱动程序）* > *Timer （定时器）* > *Use Timer Driver? （使用定时器驱动程序？）* 并选择 *TMR Driver Instance 0 （TMR 驱动程序实例 0）*。
4. 根据以下公式更改定时器周期值：
   - 定时器周期 `= 721 * APP_NUM_ANX_PINS`（其中， `APP_NUM_ANX_PINS` 在 `adcp_config.h` 中定义，它表示要采样的引脚数）
5. 再次生成并运行演示

BMX 外设库示例

本主题介绍 BMX 外设库示例。

简介

BMX 外设库演示应用程序帮助

说明

该分发包包含一个与 BMX 相关的固件项目，用于演示 MPLAB Harmony BMX 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 BMX 外设库演示应用程序。

mem_partition

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例会在 PIC32MX 器件中设置一个存储器分区，用于内核和用户段。如果存储器设置正确， Explorer 16 开发板上的 LED 会点亮。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 BMX 外设库 BMX 处理程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `mem_partition.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/bmx/mem_partition`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`mem_partition.X`	`<install-dir>/apps/examples/peripheral/bmx/mem_partition/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板上演示 BMX 存储器分区演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 BMX 演示。

说明

该演示会通过 BMX 外设库设置内核和用户存储器分区。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 观察 Explorer 16 开发板上的 LED 的状态。如果存储器分区正确， Explorer 16 开发板上的 LED 会点亮。

CAN 外设库示例

本主题介绍 CAN 演示。

简介

CAN 库演示应用程序帮助。

说明

该分发包包含一个与 CAN 相关的固件项目，用于演示 MPLAB Harmony CAN 库的功能。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 CAN 库演示应用程序。

echo_send

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示说明如何使用 CAN 外设库和器件上的 CAN 外设通过 CAN 总线发送报文。该演示使用报文过滤来回送发送到其特定地址的用户数据。该演示不使用 CAN 驱动程序或 CAN 协议栈。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 CAN 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `echo_send.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/can/echo_send`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
echo_send.X	<install-dir>/apps/examples/peripheral/can/echo_send/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_125_sk	pic32mx_125_sk	演示如何在 PIC32MX1/2/5 入门工具包上通过 CAN 总线发送报文。
pic32mz_ec_sk_io_ebrd_can_pictail	pic32mz_ec_sk	使用连接到 PIC32MZ EC 适配器板和入门工具包 I/O 扩展板的 CAN/LIN PICtail Plus 子板和 PIC32MZ EC 入门工具包，演示如何通过 CAN 总线发送报文。
pic32mz_ef_sk_io_ebrd_can_pictail	pic32mz_ef_sk	使用连接到 PIC32MZ EC 适配器板和入门工具包 I/O 扩展板的 CAN/LIN PICtail Plus 子板和 PIC32MZ EF 入门工具包，演示如何通过 CAN 总线发送报文。
pic32mz_da_sk_io_ebrd_can_pictail	pic32mz_da_sk	使用连接到 PIC32MZ EC 适配器板和入门工具包 I/O 扩展板的 CAN/LIN PICtail Plus 子板和 PIC32MZ DA 入门工具包，演示如何通过 CAN 总线发送报文。
pic32mx_usb_sk2	pic32mx_usb_sk2	使用组合的 CAN/LIN PICtail Plus 子板和 PIC32 USB 入门工具包 II，演示如何通过 CAN 总线发送报文。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX1/2/5 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

连接到 PIC32MZ EC 适配器板的 PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包

在 PIC32MZ EC 适配器板上，短接 JP1 的引脚 1 和 2，并短接 JP3 的引脚 1 和 2

连接到 PIC32MZ EC 适配器板和入门工具包 I/O 扩展板的 PIC32MZ DA 入门工具包

为使该配置可以工作，在 PIC32MZ EC 适配器板上， JP1 和 JP3 应开路。

在入门工具包 I/O 扩展板上配置以下跳线：

- 将 J11 引脚 12 设置到 J11 引脚 7
- 将 J11 引脚 9 设置到 J11 引脚 8

CAN/LIN PICtail Plus 子板

- 对于 CAN1，将 J4 设置到 1，将 J2 设置到 2-3
- 对于 CAN2，通过将 J15 和 J16 设置到 1-2 来使用 RE5 和 RE4

运行演示

介绍如何编译并运行 CAN 演示。

说明

运行该演示需要以下工具：

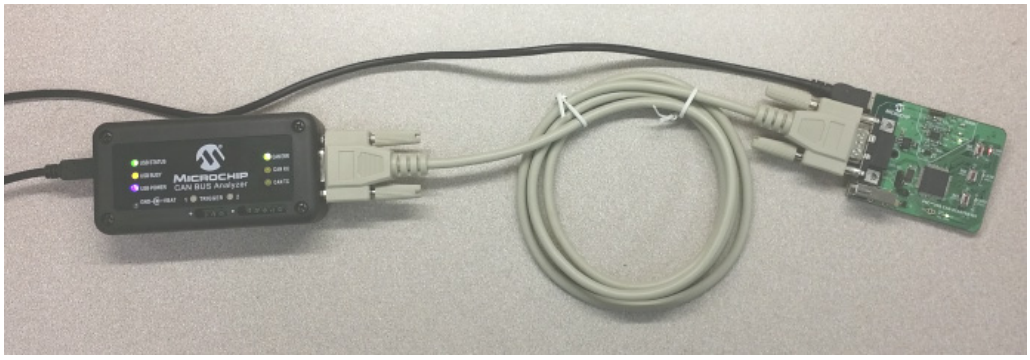
- MPLAB X IDE
- MPLAB XC32 C/C++ 编译器

注： 请参见具体版本的发行说明。

- 以下工具之一：
  - PIC32MX1/2/5 入门工具包
  - CAN/LIN PICtail Plus 子板和 PIC32 USB 入门工具包 II 组合
  - PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包或 PIC32MZ DA 入门工具包
- 一个 Microchip CAN 总线分析器 （APGDT002）或等效工具
- 一条 DB9 电缆
- 两条 USB A 转 B 迷你电缆
- 个人计算机

硬件示例

下图说明了 PIC32MX1/2/5 入门工具包的硬件设置。

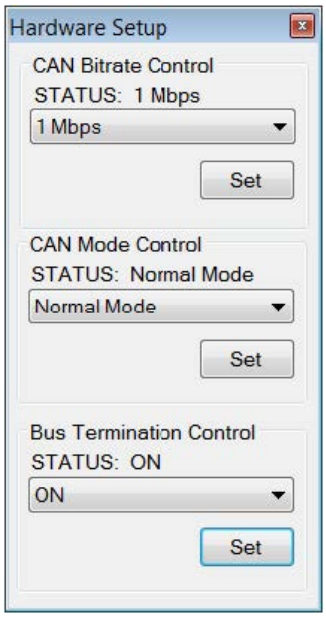


关于 CAN 总线分析器的更多操作指示信息，请参见 *“CAN BUS Analyzer User's Guide ”*（DS50001848）。

关于 CAN 总线的更多信息，请访问：http://www.can-cia.org/

执行以下操作来运行演示：

1. 将 DB9 电缆的一端连接到入门工具包，另一端连接到 CAN 总线分析器。
2. 将 USB 电缆连接到入门工具包上的 J3 和 CAN 总线分析器。
3. 打开 CAN 总线分析器软件。
4. 将分析器配置为使用在 MHC 内选择的波特率 （典型值为 1 Mbps），正常操作，端接电阻：ON （开启）。



5. 启动 MPLAB X IDE 并打开项目 echo_send.X。
6. 对入门工具包编程。
7. 在 CAN 总线分析器软件中打开 Rolling Trace （滚动跟踪）窗口。
8. 按下入门工具包上的按钮，以便在跟踪窗口中查看输出。
9. 打开 Transmit （发送）窗口。
10. 要获得来自入门工具包的响应，必须使用以下地址：ID = 0x201。在地址后面，输入 DLC，即数据长度代码 （要跟随的字节数）。例如，如果输入了 0x201, DLC, Byte1, Byte2, Byte3, Byte4，入门工具包的响应将为：0x102, DLC, Byte1, Byte2, Byte3, Byte4。

演示输出

下图显示了来自 CAN 总线分析器的 Rolling Trace 发送 / 接收输出。

Rolling Trace

TRACI	ID	DL(	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	TIME STAMP (sec)	TIME DELTA (sec)
RX	0x102	4	0x44	0x33	0x22	0x11					759.7169	696.243
TX	0x201	4	0x11	0x22	0x33	0x44					63.4742	993.755

按钮按下映射

SW1 按下 ——0x12C, 2, 0x0B, 0x16。

下图显示了 Rolling Trace 输出消息。

File View Tools Setup Help

Rolling Trace

TRACI	ID	DL(	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	TIME STAMP (sec)	TIME DELTA (sec)
RX	0x12C	2	0x0B	0x16							876.5379	0.018
RX	0x55	4	0x11	0x22	0x33	0x44					873.5240	4.791
RX	0x100	4	0xDE	0xAD	0xBE	0xEF					868.6979	108.981

比较器外设库示例

本主题介绍比较器外设库示例。

简介

比较器外设库演示应用程序帮助

说明

该分发包包含一个与比较器相关的固件项目，用于演示 MPLAB Harmony 比较器外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的比较器外设库演示应用程序。

simple_comparator

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例会设置 PIC32MX 器件上的比较器，将反相输入上的电位器输入与同相输入上的参考电压进行比较。根据比较结果，Explorer 16 开发板上的一半 LED 将会点亮。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了比较器外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `simple_comparator.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/cmp/simple_comparator`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`simple_comparator.X`	`<install-dir>/apps/examples/peripheral/cmp/simple_comparator/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板上运行的简单比较器演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行比较器演示。

说明

该演示会将模拟比较器设置为在反相和同相输入的相对值发生变化时触发中断。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 观察 Explorer 16 开发板上的 LED 的状态。根据反相和同相输入之间的比较结果，一半 LED （D3-D6 或 D7-D10）会点亮。

CVREF 外设库示例

本主题介绍 CVREF 外设库示例。

简介

CVREF 外设库演示应用程序帮助。

说明

该分发包包含一个与 CVREF 相关的固件项目，用于演示 MPLAB Harmony CVREF 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 CVREF 外设库演示应用程序。

triangle_wave

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例会设置 PIC32 器件中的 CVREF 分压器，产生可以在示波器上观察到的三角波形。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 CVREF 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `triangle_wave.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/cvref/triangle_wave`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`triangle_wave.X`	`<install-dir>/apps/examples/peripheral/cvref/triangle_wave/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板上运行的 CVREF 三角波形演示。
pic32mz_ec_pim_e16	pic32mz_ec_pim+e16	在 PIC32MZ2048ECH100 PIM 和 Explorer 16 开发板上运行的 CVREF 三角波形演示。
pic32mz_ef_pim_e16	pic32mz_ef_pim+e16	在 PIC32MZ2048EFH100 PIM 和 Explorer 16 开发板上运行的 CVREF 三角波形演示。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ DA 入门工具包上运行的 CVREF 三角波形演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板
将示波器探针连接到 RB10，即 CVREFOUT 引脚。

PIC32MZ2048ECH100 PIM 和 Explorer 16 开发板
将示波器探针连接到 RB10，即 CVREFOUT 引脚。

PIC32MZ2048EFH100 PIM 和 Explorer 16 开发板
将示波器探针连接到 RB10，即 CVREFOUT 引脚。

PIC32MZ DA 入门工具包和入门工具包 I/O 扩展板
将示波器探针连接到 RB10，即 CVREFOUT 引脚。

运行演示

介绍如何编译并运行 CVREF 演示。

说明

该演示会通过改变 CVREF 引脚上的电压输出来产生可以使用示波器观察的三角波形。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 使用示波器观察 CVREFOUT 引脚的输出。将观察到不同电压大小的三角波形。

DMA 外设库示例

本主题介绍 DMA 外设库示例。

简介

DMA 外设库演示应用程序帮助

说明

该分发包包含一个与 DMA 相关的固件项目，用于演示 MPLAB Harmony DMA 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 DMA 外设库演示应用程序。

dma_led_pattern

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会使用由 Timer1 中断触发的 DMA 传输 （通过运用 DMA 外设库）在 LED 上显示存储在闪存中的模式。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 DMA 外设库 DMA 处理程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `dma_led_pattern.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/dma/dma_led_pattern`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`dma_led_pattern.X`	`<install-dir>/apps/examples/peripheral/dma/dma_led_pattern/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
bootloader_pic32mx_eth_sk	pic32mx_eth_sk	该配置由自举程序演示使用和编程。更多信息，请参见这些演示。
bootloader_pic32mz_ec_sk	pic32mz_ec_sk	该配置由自举程序演示使用和编程。更多信息，请参见这些演示。
bootloader_pic32mz_da_sk	pic32mz_da_sk	该配置由自举程序演示使用和编程。更多信息，请参见这些演示。
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板组合上演示 DMA LED 模式。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 DMA LED 模式。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 DMA LED 模式。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示 DMA LED 模式。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MX795F512L PIM 和 Explorer 16 开发板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 DMA 演示。

说明

该演示会基于 DMA 报警触发中断。

1. 首先编译并对目标器件编程。在编译时，选择对应于所用硬件的配置。
2. 在成功执行时观察 LED 上的模式。该模式会重复且连续地使 LED 不断闪烁。如果发生了故障， LED 将停止切换。

EBI 外设库示例

本主题介绍 EBI 外设库示例。

简介

外部总线接口 （External Bus Interface， EBI）外设库演示应用程序帮助

说明

该分发包包含一个 EBI SRAM 读 / 写演示项目，用于演示 EBI 的功能，以及它存储和访问它所连接器件上的数据的能力。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 EBI 外设库演示应用程序。

sram_read_write

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

用于 EBI 外设库 SRAM 读 / 写演示。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `sram_read_write.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/ebi/sram_read_write`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`sram_read_write.X`	`<install-dir>/apps/examples/peripheral/ebi/sram_read_write`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	在 PIC32MZ EC 入门工具包和 MEB II 上演示 SRAM 读 / 写。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	在 PIC32MZ EF 入门工具包和 MEB II 上演示 SRAM 读 / 写。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 EBI 基本演示。

说明

请使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 将 PIC32MZ EC 入门工具包连接到 MEB II。
3. 将 USB 电缆连接到 PIC32MZ EC 入门工具包上的 mini-B 调试器端口，另一端连接到计算机。
4. 编译、下载演示项目并在目标板上运行。

该演示中的状态机有 4 种状态：APP_STATE_WRITE、 APP_STATE_READBACK、 APP_STATE_DONE 和 APP_STATE_FAIL。

该演示会初始化 EBI 硬件模块。配置硬件后，该演示会通过一个前行地址向整个存储器阵列中写入数据。在该演示完成数据写入后，它会回读整个存储器阵列并使用期望数据来检查它。

如果检查通过，该演示会转到 APP_STATE_DONE 状态并保持该状态，然后通过点亮 PIC32MZ EC 入门工具包的 LED3（绿色 LED）指示演示成功。

如果检查失败，该演示会转到 APP_STATE_FAIL 状态并保持该状态，然后通过点亮 PIC32MZ EC 入门工具包的 LED1 （红色 LED）来指示失败。

I2C 外设库示例

本主题介绍 I2C 外设库示例。

简介

I2C 外设库演示应用程序帮助

说明

该分发包包含两个与 I2C 相关的固件项目，用于演示 MPLAB Harmony I2C 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 I2C 外设库演示应用程序。

i2c_interrupt

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 I2C 外设库示例演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `i2c_interrupt.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/i2c/i2c_interrupt`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`i2c_interrupt.X`	`<install-dir>/apps/examples/peripheral/i2c/i2c_interrupt/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	该配置的目的是在中断模式和动态操作下演示 I2C 主模式传输设置。所用的硬件是连接到 Explorer 16 开发板的 PIC32MX795F512L PIM。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置的目的是在中断模式和动态操作下演示 I2C 主模式传输设置。所用的硬件是连接到 MEB II 的 PIC32MZ EC 入门工具包。

pic32mz_da_sk_meb2_wvga	pic32mz_da_sk+meb2+wvga	该配置的目的是在中断模式和动态操作下演示 I2C 主模式传输设置。所用的硬件是连接到 MEB II 的 PIC32MZ 图形 （DA）入门工具包。

配置硬件

介绍如何配置支持的硬件。

说明

注： i2c_interrupt 演示在 Microchip MCP7949N RTCC 器件上进行了测试。该器件的地址为 0xDE。

Explorer 16 开发板与 PIC32MX795F512L PIM

- 在将 PIC32MX795F512L PIM 连接到 Explorer 16 开发板之前，请确保处理器选择开关 （S2）处于 PIM 位置
- 短接 Explorer 16 开发板上的 JP2，以使能 LED

如果使用入门工具包 I/O 扩展板，则进行以下连接：

- I/O 扩展板 J11 引脚 36 （SDA2）和 J11 引脚 38 （SCL2）应通过 2.2k 欧姆的电阻上拉至 3.3V

如果使用外部 RTCC，则进行以下连接：

- 通过跳线将 I/O 扩展板 J11 引脚 36 （SDA2）和 J11 引脚 38 （SCL2）连接到外部 I2C 器件相应的信号线
- 如果连接了外部 I2C 器件，请确保主器件和从器件共用一个公共地。

如果使用 PICtail Plus 子板，则进行以下连接：

- PICtail Plus 子板引脚 RA2 （SCL2）和 RA3 （SDA2）应通过 2.2k 欧姆的电阻上拉至 3.3V

如果使用外部 RTCC，则进行以下连接：

- 通过跳线将 PICtail Plus 子板引脚 RA2 （SCL2）和引脚 RA3 （SDA2）连接到外部 I2C 器件相应的 SCL 和 SDA 线
- 如果连接了外部 I2C 器件，请确保主器件和从器件共用一个公共地。

连接到 MEB II 的 PIC32MZ EC 入门工具包、 PIC32MZ EF 入门工具包和 PIC32MZ 图形 （DA）入门工具包

PIC32MZ EC/EF 入门工具包上的跳线 JP2 应根据所使用的调试器 / 编程器进行如下连接：

- 如果使用 PKOB，则应短接引脚 1 和 3 以及引脚 2 和 4
- 如果使用 MPLAB REAL ICE 或 MPLAB ICD 3，则应将引脚 1 和 3 以及引脚 2 和 4 保留为开路

与 I2C 有关的连接如下：

- 将 MEB II J2 引脚 3 （SCL2）连接到外部 I2C 器件相应的 SCL 线
- 将 MEB II J2 引脚 5 （SDA2）连接到外部 I2C 器件相应的 SDA 线
- 如果连接了外部 I2C 器件，请确保主器件和从器件共用一个公共地。

运行演示

介绍如何编译并运行 I2C 演示。

说明

1. 该演示说明如何配置并使用 I2C 驱动程序 API 来支持中断模式下的 I2C 缓冲操作。在该演示中，I2C 配置为单个实例和单个客户端。

成功编译所选配置的演示应用程序之后，可以将固件编程到目标器件中。

要在调试模式下运行演示，请执行以下步骤：

1. 基于目标硬件，从 MPLAB X IDE 的 Project Properties 中选择相应的配置。
2. 在 MPLAB X IDE 中，从 Hardware Tool 菜单中选择您的器件编程器。
3. 选择 *Debug* > *Debug Main Project* 或在工具栏中单击 **Debug Main Project**。
4. 编译 MPLAB X IDE 项目中的选定配置，并对演示板编程。

I2C 驱动程序会将 I2C 外设的 I2C2 实例配置为主模式。 SDA 和 SCL 线按配置硬件介绍的方式连接到 Microchip MCP7940N RTCC 器件。主器件会写入 RTCC 器件的 SRAM 存储器中的连续存储单元。然后，主器件会从同一页读回其内容。

可以通过检查缓冲区变量的内容来确定该操作的结果。

下表列出了预期的结果。

测试用例	缓冲区的内容
I2C2 （主器件）	RXbuffer_6[] = "3RTCCSLAVE" （从 RTCC 器件接收到的数据）

输入捕捉外设库示例

本主题介绍输入捕捉外设库示例。

简介

输入捕捉外设库演示应用程序帮助

说明

该分发包包含一个与输入捕捉相关的固件项目，用于演示 MPLAB Harmony 输入捕捉外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的输入捕捉外设库演示应用程序。

ic_basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示基于输入捕捉 1 引脚 （RD8）的上升沿来运用定时器捕捉。定时器捕捉值存储在 “CaptureTime” 变量中。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了输入捕捉外设库输入捕捉处理程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 ic_basic.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/examples/peripheral/ic/ic_basic。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
ic_basic.X	<install-dir>/apps/examples/peripheral/ic/ic_basic/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 和 I/O 扩展板组合上演示输入捕捉外设库。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II 和入门工具包 I/O 扩展板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 IC 演示。

说明

该演示会触发中断。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 在成功进行输入捕捉后， LED3 将点亮。

NVM 外设库示例

本主题介绍 NVM 外设库示例。

简介

NVM 外设库演示应用程序帮助

说明

该分发包包含一个与 NVM 相关的固件项目，用于演示 MPLAB Harmony NVM 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 NVM 外设库演示应用程序。

flash_modify

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会执行以下操作：

- 擦除程序闪存中的一页
- 向程序闪存中写入一行数据
- 读取并验证写入的数据
- 通过板载 LED 指示操作成功或失败

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 NVM 外设库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `flash_modify.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/nvm/flash_modify`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`flash_modify.X`	`<install-dir>/apps/examples/peripheral/nvm/flash_modify/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板组合上演示 NVM 外设库。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 NVM 外设库。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 NVM 外设库。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示 NVM 外设库。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

- 开关 S2 应设置到 PIM
- 跳线 J2 应就位

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

移除跳线 JP1。

PIC32MZ DA 入门工具包

移除跳线 JP1。

运行演示

介绍如何编译并运行 NVM 演示。

说明

该演示通过 NVM 外设库来执行内部闪存擦除操作、写操作和读操作。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 在成功执行演示时，相应硬件上的 LED 会点亮，指示成功或失败。关于不同配置的 LED 状态，请参见下表。

项目配置	成功状态	失败状态
pic32mx795_pim_e16	LED4 点亮	LED3 点亮
pic32mz_ec_sk pic32mz_ef_sk pic32mz_da_sk	LED3 点亮	LED1 点亮

输出比较外设库示例

本主题介绍输出比较外设库示例。

简介

输出比较外设库演示应用程序帮助。

说明

该分发包包含一个与输出比较相关的固件项目，用于演示 MPLAB Harmony 输出比较外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的输出比较外设库演示应用程序。

oc_pwm

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会在输出比较 1 输出引脚上生成占空比为 25% 的 40 kHz PWM。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了输出比较外设库输出比较处理程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `oc_pwm.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/oc/oc_pwm`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
oc_pwm.X	<install-dir>/apps/examples/peripheral/oc/oc_pwm

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示输出比较 PWM。
pic32mz_ec_sk	pic32mz_ec_sk+meb2	在 PIC32MZ EC 入门工具包和 MEB II 组合上演示输出比较 PWM。
pic32mz_ef_sk	pic32mz_ef_sk+meb2	在 PIC32MZ EF 入门工具包和 MEB II 组合上演示输出比较 PWM。
pic32mz_da_sk	pic32mz_da_sk+meb2	在 PIC32MZ DA 入门工具包和 MEB II 组合上演示输出比较 PWM。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II 和入门工具包 I/O 扩展板

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 OC 演示。

说明

该演示会展示在输出比较引脚上生成的 PWM。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 对于相应的配置，用户应使用示波器在以下引脚上看到占空比为 25% 的 40 kHz 信号：
   - pic32mx_usb_sk2：I/O 扩展板上的输出比较 1 输出引脚
   - pic32mz_ec_sk 或 pic32mz_ef_sk：MEB II 的 PICtail 连接器上的输出比较 5 输出引脚 （引脚 22）

振荡器外设库示例

本主题介绍振荡器外设库示例。

简介

振荡器外设库演示应用程序帮助

说明

该分发包包含一个与振荡器相关的固件项目，用于演示 MPLAB Harmony 振荡器外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的振荡器外设库演示应用程序。

osc_config

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例演示如何在运行时期间更改系统时钟源和 PLL 值。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了振荡器外设库的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `osc_config.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/osc/osc_config`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`osc_config.X`	`<install-dir>/apps/examples/peripheral/osc/osc_config/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示 Timer3 中断。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 Timer3 中断。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上演示 Timer3 中断。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示 Timer3 中断。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行振荡器演示。

说明

该演示会在运行时针对不同设置配置振荡器。

1. 首先编译并对目标器件编程。在编译时，选择所用硬件的相应配置。
2. 如果时钟设置在运行时期间发生更改， LED3 会点亮。

PMP 外设库示例

本主题介绍 PMP 外设库示例。

简介

PMP 外设库演示应用程序帮助

说明

该分发包包含一个与 PMP 相关的固件项目，用于演示 MPLAB Harmony PMP 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 PMP 外设库演示应用程序。

pmp_lcd

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例会初始化 Explorer 16 开发板上的 LCD，并使用 PMP 外设库向它发送信息字符串。

编译应用程序

用于 PMP 外设库 PMP 报警演示。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `pmp_lcd.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/pmp/pmp_lcd`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`pmp_lcd.X`	`<install-dir>/apps/examples/peripheral/pmp/pmp_lcd/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 Explorer 16 开发板和 PIC32MX795F512L PIM 上演示 PMP LCD 配置。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 PMP 演示。

说明

该演示使用 PMP 外设库函数来说明 PMP/ 计数器功能。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 检查 Explorer 16 开发板的 LCD。在运行时，显示屏上应显示消息 “Hello! Testing...”（您好 ! 测试中...）。

端口外设库示例

本主题介绍端口外设库示例。

简介

端口外设库演示应用程序帮助。

说明

该分发包包含两个与端口相关的固件项目，用于演示 MPLAB Harmony 端口外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的端口外设库演示应用程序。

blinky_leds

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例会使用选定频率来使 LED 发生闪烁。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了端口外设库的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `blinky_leds.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/ports/blinky_leds`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`blinky_leds.X`	`<install-dir>/apps/examples/peripheral/ports/blinky_leds/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示端口外设库。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示端口外设库。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示端口外设库。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示端口外设库。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行端口演示。

说明

该演示说明如何运用端口外设库功能。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 观察 LED3 是否闪烁。
3. 更改 `app.h` 中 APP_LED_BLINK_DELAY 的值，然后编译并运行代码。观察 LED 闪烁频率是否发生了变化。

cn_interrupt

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例使用与外设库和特定硬件关联的电平变化通知中断来控制 LED 翻转。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了端口外设库的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cn_interrupt.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/ports/cn_interrupt`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cn_interrupt.X`	`<install-dir>/apps/examples/peripheral/ports/cn_interrupt/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示端口外设库。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示端口外设库。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示端口外设库。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示端口外设库。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行电平变化通知中断演示。

说明

该演示使用电平变化通知中断来控制 LED 翻转。板载开关和消抖逻辑用于电平变化通知中断。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 检查 LED3 的状态，它应该发生闪烁。
3. 如果按下入门工具包上的开关 1 一次， LED3 会停止闪烁。
4. 如果再次按下开关 1， LED3 会恢复闪烁。
5. 可以多次重复步骤 3 和 4。

电源外设库示例

本主题介绍电源外设库示例。

简介

电源外设库演示应用程序帮助。

说明

该分发包包含两个与电源相关的固件项目，用于演示 MPLAB Harmony 电源外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的电源外设库演示应用程序。

sleep_mode

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示使用看门狗定时器从休眠模式中唤醒器件，然后基于状态翻转 LED。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了电源外设库休眠模式演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `sleep_mode.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/power/sleep_mode`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`sleep_mode.X`	`<install-dir>/apps/examples/peripheral/power/sleep_mode/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示休眠模式。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示休眠模式。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示休眠模式。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行电源演示。

说明

该演示会将器件置为休眠模式，然后使用看门狗定时器唤醒它。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 检查 LED 的状态。 LED1 会点亮几秒钟，然后熄灭，然后 LED3 会开始翻转。

deep_sleep_mode

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会将器件置为深度休眠模式，并使用深度休眠看门狗定时器（Deep Sleep Watchdog Timer， DSWDT）来唤醒器件。该示例还演示了使用深度休眠通用寄存器 （Deep Sleep General Purpose Register， DSGPR）来保存对于应用程序非常关键的内容。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了电源外设库深度休眠模式演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `deep_sleep_mode.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/power/deep_sleep_mode`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`deep_sleep_mode.X`	`<install-dir>/apps/examples/peripheral/power/deep_sleep_mode/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）系列入门工具包上演示深度休眠模式。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行电源演示。

说明

该演示会将器件置为深度休眠模式，然后使用深度休眠看门狗定时器 （DSWDT）来唤醒它。

1. 编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 检查LED的状态。分别将LED3、LED2和LED1视为二进制的BIT2、BIT1和BIT0。观察二进制计数从000递增到111，这会不断重复。
3. 在任意计数值下按下开关 2，会将器件置为深度休眠模式。观察计数停止，器件将处于深度休眠模式，直到 DSWDT 发生超时 （约 5 秒）。

4. 5 秒之后，器件会从深度休眠中唤醒，并恢复计数。

基本操作

该应用程序会在 LED 上显示二进制计数。当按下开关 2 时，应用程序会将计数值存储到深度休眠通用寄存器（DSGPR0）中，并将器件置为深度休眠模式。

在发生 DSWDT 超时事件时，器件会唤醒，并且应用程序会检查器件先前是否处于深度休眠模式。如果是，则读取 DSGPR0 中的计数值，并且器件从该值开始恢复 LED 计数。

复位外设库示例

本主题介绍复位外设库示例。

简介

复位外设库演示应用程序帮助

说明

该分发包包含一个与复位相关的固件项目，用于演示 MPLAB Harmony 复位外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的复位外设库演示应用程序。

reset_handler

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例会将每个复位标志分配到一个 I/O 端口引脚，然后对各个复位标志执行检查。如果标志置 1，相应的 LED 会点亮。然后，可以通过按下电路板上的开关清零所有标志。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了复位外设库复位处理程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `reset_handler.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/reset/reset_handler`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`reset_handler.X`	`<install-dir>/apps/examples/peripheral/reset/reset_handler/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板上演示复位外设库。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示复位外设库。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示复位外设库。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ DA 入门工具包上演示复位外设库。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

为使 LED 工作，应连接跳线 JP2 （短接）。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行复位演示。

说明

该演示会确定复位的原因并相应地点亮 LED。

首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。

PIC32MX795F512L PIM 和 Explorer 16 开发板：

通过检查 LED 状态来观察复位的不同原因，如下：

1. 按下开关 S6 来清零所有复位标志，这会使所有 LED 熄灭。
2. 按下电路板上的 MCLR 开关，以产生复位。 LED D9 应点亮。
3. 拔下电源并再次插上，以产生上电复位 （Power-on Reset， POR）。 LED D5 应点亮。
4. 再次按下 MCLR 开关。 LED D5 和 D9 应都点亮。
5. 再次按下开关 S6 来清除所有复位，这会使所有 LED 熄灭。
6. 按下开关 S4 来使能看门狗定时器。几秒钟后， LED 10 应点亮，指示发生了 WDT 复位。

可以从头开始重复这些步骤。

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包或 PIC32MZ DA 入门工具包：

1. 按下开关 S3 来清零所有复位标志，这会使所有 LED 熄灭。
2. 按下开关 S1 来使能看门狗定时器。几秒钟后， LED 1 应点亮，指示发生了 WDT 复位。

注： POR 会清零除 BOR 和 POR 标志之外的所有其他复位标志。

RTCC 外设库示例

本主题介绍 RTCC 外设库示例。

简介

RTCC 外设库演示应用程序帮助

说明

该分发包包含一个与 RTCC 相关的固件项目，用于演示 MPLAB Harmony RTCC 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 RTCC 外设库演示应用程序。

rtcc_alarm

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会设置 RTCC 模块，使它作为闹钟工作，在每天上午 6:00 启动。时间设置为 5:59:55，所以在运行代码后，闹钟会触发 5 秒。

编译应用程序

用于 RTCC 外设库 RTCC 闹钟演示。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `rtcc_alarm.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/rtcc/rtcc_alarm`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`rtcc_alarm.X`	`<install-dir>/apps/examples/peripheral/rtcc/rtcc_alarm/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板组合上演示 RTCC 闹钟。

pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 RTCC 闹钟。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上演示 RTCC 闹钟。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示 RTCC 闹钟。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MX795F512L PIM 和 Explorer 16 开发板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 RTCC 闹钟演示。

说明

该演示会基于 RTCC 闹钟触发中断。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 检查 LED 的状态 （Explorer 16 开发板上的 D3 或入门工具包上的 LED3）。 LED 点亮状态指示成功， LED 熄灭状态指示失败。

SPI 外设库示例

本主题介绍 SPI 外设库示例。

简介

SPI 外设库演示应用程序帮助

说明

该分发包包含两个与 SPI 相关的固件项目，用于演示 MPLAB Harmony SPI 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 SPI 外设库演示应用程序。

spi_loopback

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会对数据传输进行环回。该示例假定 SPI SDO （输出）已连接到 SDI （输入）。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 PIO 模式下的 SPI 外设库闪存读操作演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `spi_loopback.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/spi/spi_loopback`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`spi_loopback.X`	`<install-dir>/apps/examples/peripheral/spi/spi_loopback/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 和入门工具包 I/O 扩展板组合上演示 SPI 环回。
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM、Explorer 16 开发板和入门工具包 I/O 扩展板组合上演示 SPI 环回。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接（EC）入门工具包和入门工具包 I/O 扩展板组合上演示 SPI 环回。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包和入门工具包 I/O 扩展板组合上演示 SPI 环回。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形（DA）入门工具包和入门工具包 I/O 扩展板组合上演示 SPI 环回。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM、 Explorer 16 开发板和入门工具包 I/O 扩展板

- （使用连线）连接入门工具包 I/O 扩展板上的 SDI1 和 SDO1 引脚
- 将入门工具包 I/O 扩展板连接到 Explorer 16 开发板，并将 PIM 装到开发板上

PIC32MZ EC 入门工具包和入门工具包 I/O 扩展板

- （使用连线）连接入门工具包 I/O 扩展板上的 SDI2 引脚 （J11 引脚 32）和 SDO2 引脚 （J10 引脚 35）
- 将 PIC32MZ EC 入门工具包与适配器板连接到入门工具包 I/O 扩展板

PIC32MZ EF 入门工具包和入门工具包 I/O 扩展板

- （使用连线）连接入门工具包 I/O 扩展板上的 SDI2 引脚 （J11 引脚 32）和 SDO2 引脚 （J10 引脚 35）
- 将 PIC32MZ EF 入门工具包与适配器板连接到入门工具包 I/O 扩展板

PIC32MZ DA 入门工具包和入门工具包 I/O 扩展板

- （使用连线）连接入门工具包 I/O 扩展板上的 SDI1 引脚 （J11 引脚 52）和 SDO1 引脚 （J10 引脚 44）
- 将 PIC32MZ DA 入门工具包与适配器板连接到入门工具包 I/O 扩展板

PIC32 USB 入门工具包 II 和入门工具包 I/O 扩展板

- （使用连线）连接入门工具包 I/O 扩展板上的 SDI1 和 SDO1 引脚
- 将入门工具包 I/O 扩展板连接到 PIC32 USB 入门工具包 II

运行演示

介绍如何编译并运行 SPI PIO 模式传输演示。

说明

该演示会在 SPI 模块 （使用 PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包时为 SPI1 和 SPI2，使用 PIC32MZ DA 入门工具包时为 SPI1 和 SPI3）上进行数据环回。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 检查 LED 的状态：
   - Explorer 16 开发板上的 LED D4 和 D5
   - PIC32 USB 入门工具包 II、PIC32MZ EC 入门工具包、PIC32MZ EF 入门工具包或 PIC32MZ DA 入门工具包上的 LED3 和 LED2。
3. 如果演示成功， LED D5 或 LED3 会点亮。如果演示失败， LED D4 或 LED2 会点亮。

SQI 外设库示例

本主题介绍 SQI 外设库示例。

简介

SQI 外设库演示应用程序帮助

说明

该分发包包含两个与 SQI 相关的固件项目，用于演示 MPLAB Harmony SQI 外设库的功能。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 SQI 外设库演示应用程序。

flash_read_dma_mode

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示应用程序会在 DMA 模式下运用 SQI 模块，其操作是读取 PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包上的 SQI 闪存（SST26VF032/SST26VF032B）器件中的某个页。该演示会使用 PIO 模式来写入闪存。以下步骤按照运行时期间的调用说明了其功能：

1. 将 SQI 设置为以 25 MHz 运行。（`system_init.c`:: SYS_Initialize， `app.c`:: APP_Initialize）
2. 设定时钟频率之后，演示会为写读事务准备所连接的串行闪存器件 （`app.c`:: APP_Tasks:: APP_STATE_INIT_FLASH:: SQI_Flash_Setup）。
3. 读取所连接闪存器件的器件 ID （`app.c`:: APP_Tasks:: APP_STATE_FLASH_ID_READ:: SQI_FlashID_Read）。

4. 在 PIO 模式下写入闪存中的某个页 （app.c:: APP_Tasks:: APP_STATE_WRITE_FLASH:: SQI_PIO_PageWrite(FLASH_PAGE_ADDR)）。
5. 使用 DMA 模式读取闪存中该页的内容，并将它与写入的数据进行比较，以确保操作已成功 （app.c:: APP_Tasks:: APP_STATE_READ_FLASH_DMA_MODE:: SQI_DMA_Read(FLASH_PAGE_ADDR)）。
6. 使用入门工具包上的 LED3 （绿色 LED）指示演示成功 （app.c:: APP_Tasks:: APP_STATE_DONE:: BSP_SwitchONLED(LED_GRN)）。
7. LED3 熄灭表示演示失败 （停滞在其中一种状态）。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 DMA 模式下的 SQI 外设库闪存读操作演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 flash_read_dma_mode.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/examples/peripheral/sqi/flash_read_dma_mode。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
flash_read_dma_mode.X	<install-dir>/apps/examples/peripheral/sqi/flash_read_dma_mode/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 SQI DMA 模式传输。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 SQI DMA 模式传输。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ DA 入门工具包上演示 SQI DMA 模式传输。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SQI DMA 模式传输演示。

说明

该演示会对 SQI 闪存器件进行页写和页读。

1. 首先编译并对目标器件编程。在编译时，选择所用入门工具包的相应配置。
2. 检查所用入门工具包上的 LED3 的状态，以确定演示的状态 （LED 点亮表示成功， LED 熄灭表示失败）。

flash_read_pio_mode

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示应用程序会在 PIO 模式下运用 SQI 模块，其操作是使用 SQI 外设库读取和写入 PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包上的 SQI 闪存 （SST26VF032/SST26F032B）器件中的某个页。

以下步骤按照运行时期间的调用说明了其功能：

1. 将 SQI 设置为以 25 MHz 运行。（`system_init.c`:: SYS_Initialize， `app.c`:: APP_Initialize）。
2. 设定时钟频率之后，演示会为写读事务准备所连接的串行闪存器件 （`app.c`:: APP_Tasks:: APP_STATE_INIT_FLASH:: SQI_Flash_Setup）。
3. 读取所连接闪存器件的器件 ID （`app.c`:: APP_Tasks:: APP_STATE_FLASH_ID_READ:: SQI_FlashID_Read）
4. 在 PIO 模式下写入闪存中的某个页 （`app.c`:: APP_Tasks:: APP_STATE_WRITE_FLASH:: SQI_PIO_PageWrite(FLASH_PAGE_ADDR)）。
5. 读取闪存中该页的内容，并将它与写入的数据进行比较，以确保操作已成功 （`app.c`:: APP_Tasks:: APP_STATE_READ_FLASH_PIO_MODE:: SQI_PIO_Read(FLASH_PAGE_ADDR)）。
6. 使用入门工具包上的 LED3 （绿色 LED）指示演示成功 （app.c:: APP_Tasks:: APP_STATE_DONE:: BSP_SwitchONLED(LED_GRN)）。
7. LED3 熄灭表示演示失败 （停滞在其中一种状态）。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 PIO 模式下的 SQI 外设库闪存读操作演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `flash_read_pio_mode.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/sqi/flash_read_pio_mode`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`flash_read_pio_mode.X`	`<install-dir>/apps/examples/peripheral/sqi/flash_read_pio_mode/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 SQI PIO 模式传输。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 SQI PIO 模式传输。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ DA 入门工具包上演示 SQI PIO 模式传输。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SQI PIO 模式传输演示。

说明

该演示会对 SQI 闪存器件进行页写和页读。

1. 首先编译并对目标器件编程。在编译时，选择所用入门工具包的相应配置。
2. 检查所用入门工具包上的 LED3 的状态，以确定演示的状态 （LED3 点亮表示成功， LED3 熄灭表示失败）。

flash_read_xip_mode

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示应用程序会在 XIP 模式下运用 SQI 模块，其操作是读取 PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包上的 SQI 闪存（SST26VF032/SST26VF032B）器件中的某个页。该演示会使用 PIO 模式来写入闪存。

以下步骤按照运行时期间的调用说明了其功能：

1. 将 SQI 设置为以 25 MHz 运行。（system_init.c:: SYS_Initialize， app.c:: APP_Initialize）
2. 设定时钟频率之后，演示会为写读事务准备所连接的串行闪存器件 （app.c:: APP_Tasks:: APP_STATE_INIT_FLASH:: SQI_Flash_Setup）。
3. 读取所连接闪存器件的器件 ID （app.c:: APP_Tasks:: APP_STATE_FLASH_ID_READ:: SQI_FlashID_Read）。
4. 在 PIO 模式下写入闪存中的某个页 （app.c:: APP_Tasks:: APP_STATE_WRITE_FLASH:: SQI_PIO_PageWrite(FLASH_PAGE_ADDR)）。
5. 使用 DMA 模式读取闪存中该页的内容，并将它与写入的数据进行比较，以确保操作已成功 （app.c:: APP_Tasks:: APP_STATE_READ_FLASH_DMA_MODE:: SQI_XIP_Read(FLASH_PAGE_ADDR)）。
6. 使用入门工具包上的 LED3 （绿色 LED）指示演示成功 （app.c:: APP_Tasks:: APP_STATE_DONE:: BSP_SwitchONLED(LED_GRN)）。

7. LED3 熄灭表示演示失败 （停滞在其中一种状态）。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 XIP 模式下的 SQI 外设库闪存读操作演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `flash_read_xip_mode.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/sqi/flash_read_pio_mode`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`flash_read_xip_mode.X`	`<install-dir>/apps/examples/peripheral/sqi/flash_read_xip_mode/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 SQI XIP 模式传输。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ DA 入门工具包上演示 SQI XIP 模式传输。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SQI XIP 模式传输演示。

说明

该演示会对 SQI 闪存器件进行页写和页读。

1. 首先编译并对目标器件编程。在编译时，选择所用入门工具包的相应配置。
2. 检查入门工具包上的 LED3 的状态，以确定演示的状态 （LED3 点亮表示成功， LED3 熄灭表示失败）。

定时器外设库示例

本主题介绍 TMR 外设库示例。

简介

定时器外设库演示应用程序帮助

说明

该分发包包含一个与定时器相关的固件项目，用于演示 MPLAB Harmony 定时器外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的定时器外设库演示应用程序。

timer3_interrupt

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示使用 32 位模式下的 Timer3 来基于超时 （1 秒）产生中断。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TMR 外设库 TMR 报警演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `timer3_interrupt.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/tmr/timer3_interrupt`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`timer3_interrupt.X`	`<install-dir>/apps/examples/peripheral/tmr/timer3_interrupt/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板组合上演示 Timer3 中断。
pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示 Timer3 中断。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 Timer3 中断。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上演示 Timer3 中断。

pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示 Timer3 中断。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

为使 LED 工作，应连接跳线 JP2 （短接）。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 TMR 演示。

说明

该演示使用 TMR 外设库函数来说明定时器 / 计数器功能。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 检查 LED 的状态。 LED3 点亮状态指示成功， LED3 熄灭状态指示失败。

USART 外设库示例

本主题介绍 USART 外设库示例。

简介

USART （USART）外设库演示应用程序帮助

说明

该分发包包含一个与 USART 相关的固件项目，用于演示 MPLAB Harmony USART 外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 USART 外设库演示应用程序。

uart_basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示在 UART 模式下使用 USART2 向控制台发送字符，并在控制台上回显接收到的字符，同时点亮 LED。

编译应用程序

用于 USART 外设库 USART 报警演示。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `uart_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/usart/uart_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`uart_basic.X`	`<install-dir>/apps/examples/peripheral/usart/uart_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板组合上演示 USART 超时功能。
pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示 USART 外设库。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 USART 外设库。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上演示 USART 外设库。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示 USART 外设库。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

在个人计算机和 Explorer 16 开发板 P1 （UART）端口之间连接一条标准串行电缆或 USB 转 RS-232 适配器电缆。

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包

可以使用可选的 MCP2200 转接模块 （ADM00393）（它是一个 USB 转 UART 串行转换器适配器板）来运行该演示。如果使用该转接模块，则还需要其他接口。可接受的接口是入门工具包 I/O 扩展板与 PIC32MZ EC 适配器板（AC320006）。通过跳线从 J10 插头的引脚 58 和 23 分别连接到转接模块的 RX 和 TX。此外，连接 GND。

PIC32MZ DA 入门工具包

将一条 USB 电缆连接到 PIC32MZ DA 入门工具包上的 J5 mini-B 连接器。将该 USB 电缆连接到运行终端仿真程序的计算机。

PIC32 USB 入门工具包 II

可以使用可选的 MCP2200 转接模块（它是一个 USB 转 UART 串行转换器适配器板）来运行该演示。如果使用该转接模块，则还需要其他接口。可接受的接口是入门工具包 I/O 扩展板。通过跳线从 J11 插头的引脚 48 和 46 分别连接到转接模块的 RX 和 TX。此外，必须短接 GND。

运行演示

介绍如何编译并运行 USART 演示。

说明

该演示使用 USART 外设库函数来说明 USART 功能。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 启动控制台客户端 （操作系统本机客户端 /Tera Term 等）并将串行端口设置为 9600-N-1。
3. 启动演示。控制台窗口中将显示以下消息：
   *** UART Interrupt-driven Application Example ***
   *** Type some characters and observe the LED turn ON ***
4. 正如消息中所示，可以注意到控制台窗口中回显所输入的字符，并且 LED 点亮。
5. LED（Explorer 16 开发板上的 D3 或 PIC32 USB 入门工具包 II、PIC32MZ EC 入门工具包、PIC32MZ EF 入门工具包和 PIC32MZ DA 入门工具包上的 LED3）点亮表示在进行中断处理。

WDT 外设库示例

本主题介绍 WDT 外设库示例。

简介

看门狗定时器 （Watchdog Timer， WDT）外设库演示应用程序帮助

说明

该分发包包含一个与 WDT 相关的固件项目，用于演示 MPLAB Harmony 看门狗定时器外设库的功能。

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 WDT 外设库演示应用程序。

wdt_timeout

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示通过使用 WDT 外设库来运用看门狗超时功能。

编译应用程序

用于 WDT 外设库 WDT 报警演示。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wdt_timeout.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/peripheral/wdt/wdt_timeout`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wdt_timeout.X`	`<install-dir>/apps/examples/peripheral/wdt/wdt_timeout/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	在PIC32MX795F512L PIM和Explorer 16开发板组合上演示WDT超时功能。
pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示 WDT 超时功能。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ 嵌入式连接 （EC）入门工具包上演示 WDT 超时功能。
pic32mz_ef_sk	pic32mz_ef_sk	在PIC32MZ嵌入式连接与浮点单元（EF）入门工具包上演示WDT超时功能。
pic32mz_da_sk	pic32mz_da_sk	在 PIC32MZ 图形 （DA）入门工具包上演示 WDT 超时功能。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MX795F512L PIM 和 Explorer 16 开发板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 WDT 超时演示。

说明

该演示会运用 WDT 超时功能。

1. 首先编译并对目标器件编程。在编译时，选择适合于所用硬件的配置。
2. 在正常操作过程中，LED（PIC32 USB 入门工具包 II、PIC32MZ EC 入门工具包、PIC32MZ EF 入门工具包和 PIC32MZ DA 入门工具包上的 LED1，或 Explorer 16 开发板上的 LED D3）会发生闪烁。
3. 按下开关 1 （PIC32 USB 入门工具包 II、 PIC32MZ EC 入门工具包、 PIC32MZ EF 入门工具包和 PIC32MZ DA 入门工具包）或开关 S3 （Explorer 16 开发板）可以使 WDT 超时发生复位。如果演示成功， LED3 （PIC32 USB 入门工具包 II、 PIC32MZ EC 入门工具包、 PIC32MZ EF 入门工具包和 PIC32MZ DA 入门工具包）或 LED D5 （Explorer 16 开发板）会点亮。如果它们未点亮，则表示演示失败。

系统服务库示例

简介

示例应用程序提供了一些非常简单的单一用途示例，用于说明如何使用 MPLAB Harmony 系统服务库。

说明

系统服务有两种主要的实现类型：

- 系统服务库
- 低级支持

大多数系统服务库与系统的其他部分一样，遵循与设备驱动程序 （直接使用外设库来访问硬件）或中间件库 （使用设备驱动程序来访问硬件）相同的基本模型。

命令处理器系统服务示例

本主题介绍命令处理器系统服务示例。

简介

命令处理器系统服务库演示应用程序帮助。

说明

该分发包包含一个用于演示 MPLAB Harmony 命令处理器系统服务功能的固件项目。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的命令处理器系统服务库演示应用程序。

command_appio

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示应用程序通过接收用户键盘输入并在MPLAB X IDE的PIC AppIO窗口中显示输出字符串，来演示如何运用命令处理器系统服务。

该演示应用程序会执行以下操作：

1. 通过 MPLAB REAL ICE 在目标器件中启动应用程序。
2. 在初始化时，在 MPLAB X IDE 的 PIC AppIO 窗口中显示 “Ready to accept command input”（已准备好接受命令输入）。

3. 监听用户命令输入。
4. 支持该应用程序固有的两个简单命令。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了命令处理器系统服务演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `command_appio.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为`<install-dir>/apps/examples/system/command_appio`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`command_appio.X`	`<install-dir>/apps/examples/system/command_appio/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	使用 PIC32 USB 入门工具包 II、入门工具包 I/O 扩展板和 MPLAB REAL ICE 在线仿真器来演示应用程序 I/O。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

该演示需要 MPLAB REAL ICE 在线仿真器和入门工具包 I/O 扩展板。

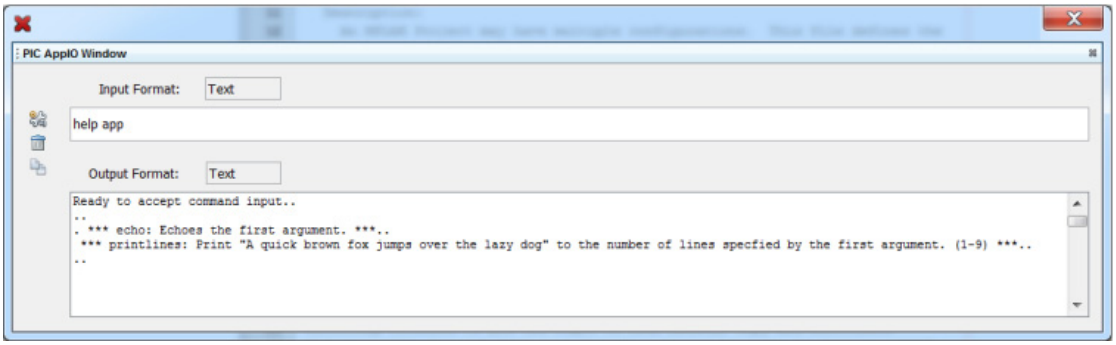
运行演示

介绍如何编译并运行命令处理器系统服务演示。

说明

执行以下操作来运行演示：

1. 确保 PIC32 USB 入门工具包 II、入门工具包 I/O 扩展板和 MPLAB REAL ICE 在线仿真器已连接并就绪。
2. 在 MPLAB X IDE 中，通过选择 *Window （窗口）* > *Debugging （调试）* > *PIC AppIO* 打开 PIC AppIO 窗口。
3. 确保 PIC AppIO 中的 Output Format （输出格式）设置为 “Text”（默认设置为 8 位十六进制）。
4. 调试项目。
5. 等待 Output Format 部分中显示 “Ready to accept command input”。
6. 在 Input Format （输入格式）栏中输入键盘输入并按 <ENTER>。
7. 输入 “help”会显示常规帮助部分。输入 “help app”会显示应用程序特有的命令，并包含简短的说明。
8. 尝试使用通过 “help app”列出的命令。



调试系统服务示例

本主题介绍调试系统服务示例。

简介

调试系统服务库演示应用程序帮助。

说明

该分发包包含一个用于演示 MPLAB Harmony 调试系统服务功能的固件项目。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的调试系统服务库演示应用程序。

debug_uart

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示应用程序通过将来自调试系统服务的消息经由控制台系统服务并使用 UART 通信协议送至个人计算机上运行的终端程序，来说明如何运用调试和控制台系统服务。

该演示应用程序会执行以下操作：

- 演示通过向终端输出字符串来进行直接控制台写操作
- 演示向终端写入格式化或未格式化的消息
- 演示如何使用全局错误级别
- 演示如何使用查询和完成回调通知来产生调试输出消息
- 演示在写队列溢出时会发生何种情况
- 演示在发生错误时清空控制台
- 演示如何使用查询和回调通知来进行控制台读操作
- 演示回显功能，即在读取字符时将它们回写到终端中

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了调试系统服务演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `debug_uart.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/system/debug_uart`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`debug_uart.X`	`<install-dir>/apps/examples/system/debug_uart/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16	pic32mx795_pim+e16	使用 Explorer 16 开发板和 PIC32MX795F512L PIM 来演示如何通过 UART RS-232 进行控制台通信。
pic32mz_ef_sk	pic32mz_ef_sk	使用 PIC32MZ EF 入门工具包来演示如何通过 UART RS-232 进行控制台通信。
pic32mz_da_sk	pic32mz_da_sk	使用 PIC32MZ DA 入门工具包来演示如何通过 UART RS-232 进行控制台通信。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 配合 Explorer 16 开发板

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

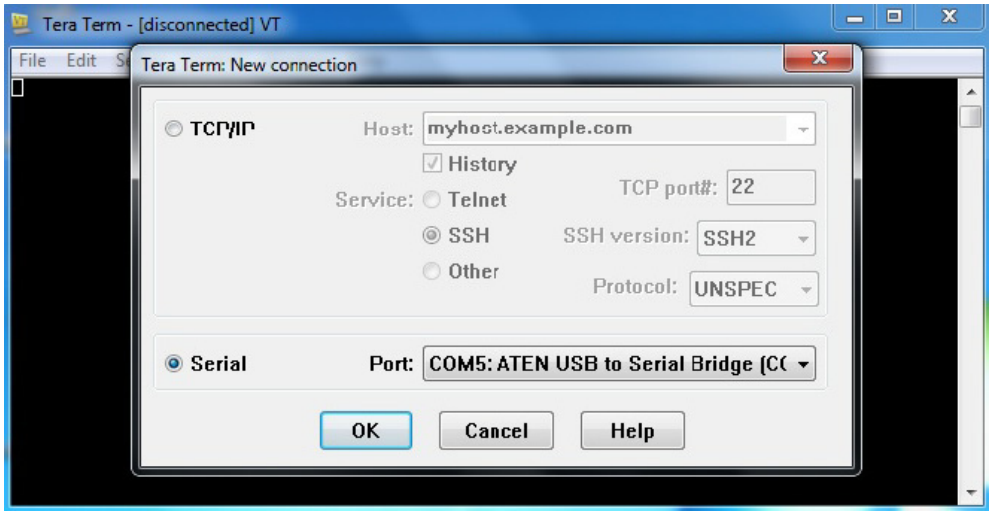
运行演示

介绍如何编译并运行调试系统服务演示。

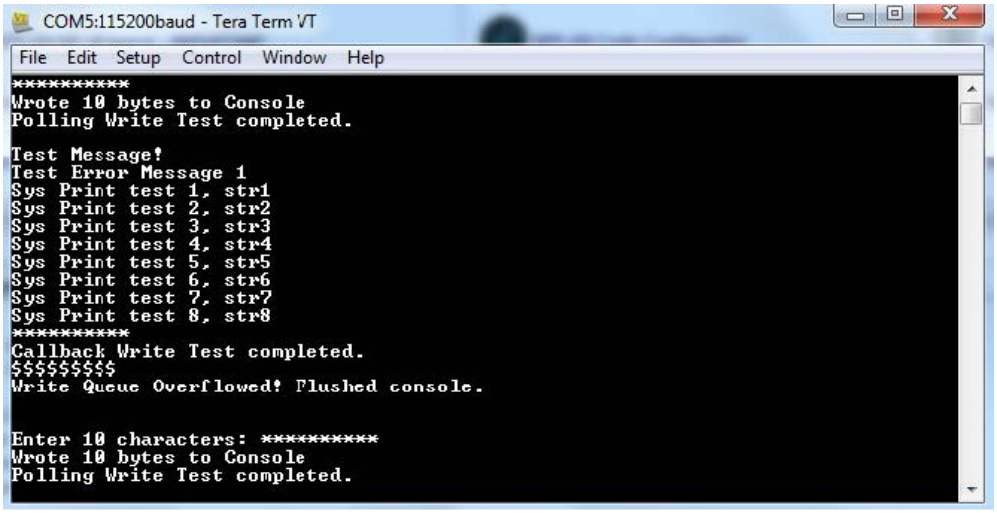
说明

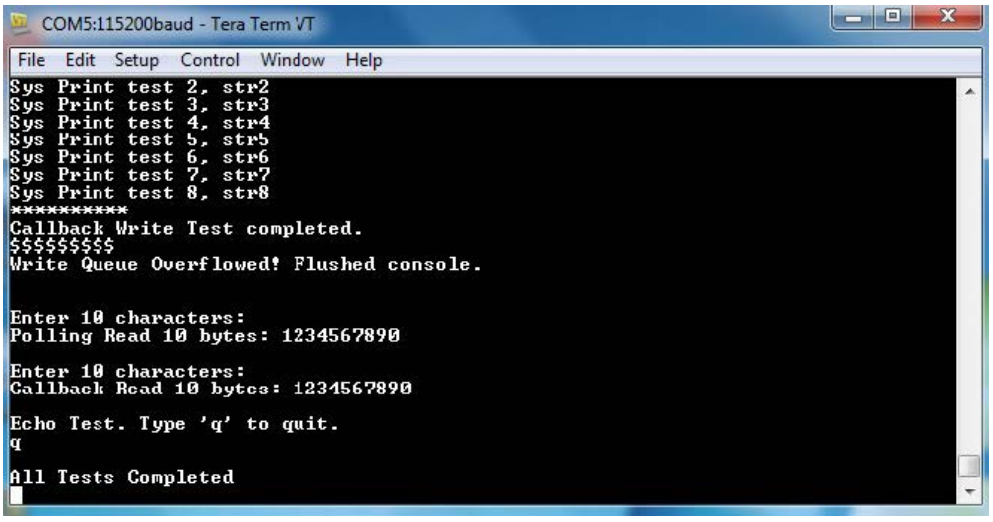
该演示会向 / 从个人计算机主机上运行的终端程序进行写入和读取。

1. 首先编译并对目标器件编程。在编译时，选择对应于所用硬件的配置。
2. 使用 RS-232 UART 连接，将 Explorer 16 开发板连接到主机个人计算机。
3. 为使终端程序可以识别 COM 器件，该演示必须在运行中。启动终端仿真器程序 （图中显示的是 Tera Term）。



4. 按下 SW1 来启动演示。
5. 按照终端上的指示信息进行操作。





debug_usb_cdc_2

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示应用程序通过将来自调试系统服务的消息经由控制台系统服务并使用USB-CDC通信协议送至个人计算机上运行的终端程序，来

说明如何运用调试和控制台系统服务。

该演示应用程序会执行以下操作：

- 演示通过向终端输出字符串来进行直接控制台写操作
- 演示向终端写入格式化或未格式化的消息
- 演示如何使用全局错误级别
- 演示如何使用查询和完成回调通知来产生调试输出消息
- 演示在写队列溢出时会发生何种情况
- 演示在发生错误时清空控制台
- 演示如何使用查询和回调通知来进行控制台读操作
- 演示回显功能，即在读取字符时将它们回写到终端中

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了调试系统服务演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `debug_usb_cdc_2.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/system/debug_usb_cdc_2`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`debug_usb_cdc_2.X`	`<install-dir>/apps/examples/system/debug_usb_cdc_2/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	使用 PIC32 USB 入门工具包 II，通过 USB 演示控制台通信。
pic32mz_ec_sk	pic32mz_ec_sk	使用 PIC32MZ EC 入门工具包，通过 USB 演示控制台通信。
pic32mz_ef_sk	pic32mz_ef_sk	使用 PIC32MZ EF 入门工具包，通过 USB 演示控制台通信。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

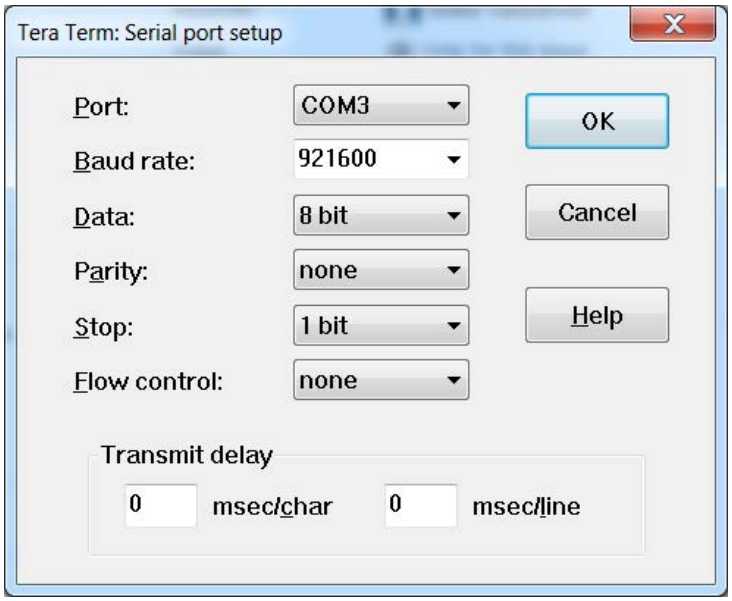
运行演示

介绍如何编译并运行调试系统服务演示。

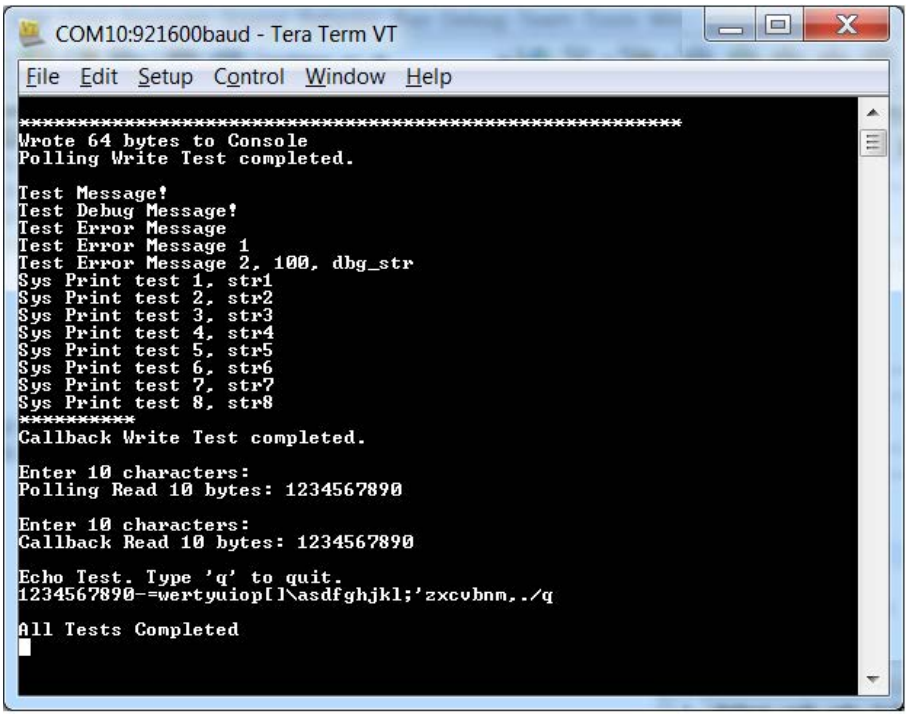
说明

该演示会向 / 从个人计算机主机上运行的终端程序进行写入和读取。

1. 首先编译并对目标器件编程。在编译时，对于 PIC32MZ EC 入门工具包选择 pic32mz_ec_sk 配置，对于 PIC32MZ EF 入门工具包选择 pic32mz_ef_sk 配置，或对于 PIC32MX USB 入门工具包 II 选择 pic32mx_usb_sk2 配置。
2. 使用 A 转 micro-A/B USB 电缆将入门工具包连接到主机个人计算机。
3. 为使终端程序可以识别 COM 器件，该演示必须在运行中。启动终端仿真器程序（图中显示的是 Tera Term），串行端口使用以下设置：（921600 波特率， 8 位数据，无奇偶校验， 1 个停止位，无流量控制）。



4. 终端将提示连接成功。按任意键启动演示。
5. 按照终端上的指示信息进行操作。



器件控制系统服务示例

本主题介绍器件控制系统服务示例。

简介

器件控制系统服务库演示应用程序帮助。

说明

该分发包包含一个用于演示 MPLAB Harmony 器件控制系统服务功能的固件项目。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的器件控制系统服务库演示应用程序。

devcon_cache_clean

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例应用程序演示在使用 DMA 从存储器向外传输数据时在怎样的情况下会出现高速缓存一致性问题。MPLAB Harmony 器件控制系统服务包含了各种高速缓存函数，该示例中使用它们来解决这种问题。

该应用程序会先分配两个缓冲区，以便使用 DMA 将一个缓冲区中的数据复制到另一个缓冲区中。先向源缓冲区中装入数据，然后启动 DMA 传输。在传输完成后，很明显两个缓冲区包含不同的值。这是一个与高速缓存一致性有关的问题。当 CPU 写入数据时，数据会存储在高速缓存中，但除非高速缓存行被清除或执行了显式高速缓存回写指令，否则数据不会被回写到 RAM 中。高速缓存指令通过器件控制系统服务进行访问，并用作底层的 MIPS® 高速缓存操作的包装器。

此特定示例中，源缓冲区中的数据在使用 DMA 传输到目标缓冲区中之前，不会被回写到 RAM 中。存储在高速缓存中但尚未被回写到 RAM 中的数据称为“脏数据”。在该应用程序的第二个部分中，演示了维持高速缓存一致性的正确方法。将数据写入源缓冲区中之后，执行函数 SYS_DEVCON_DataCacheClean，强制将数据回写到主存储器中。当启动 DMA 传输时，它会从 RAM 中拉出数据，然后将其传输到目标缓冲区 —— 这时两个缓冲区中的数据将一致。

如果应用程序按预期运行，红色和绿色 LED 将会点亮。只有在发生错误时，黄色 LED 才会点亮。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了器件控制系统服务高速缓存清除演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `devcon_cache_clean.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/system/devcon`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
devcon_cache_clean.X	<install-dir>/apps/examples/system/devcon/devcon_cache_clean/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	使用 PIC32MZ EC 入门工具包演示器件控制系统服务缓存函数。
pic32mz_ef_sk	pic32mz_ef_sk	使用 PIC32MZ EF 入门工具包演示器件控制系统服务缓存函数。
pic32mz_da_sk	pic32mz_da_sk	使用 PIC32MZ DA 入门工具包演示器件控制系统服务缓存函数。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行演示。

说明

该应用程序演示 SYS_DEVCON_DataCacheClean 函数的必要性和正确使用方式。

1. 首先编译并对目标器件编程。在编译时，选择对应于器件的正确配置。
2. 观察入门工具包开发板上的 LED。如果红色和绿色 LED 都点亮，说明应用程序执行成功。如果黄色 LED 点亮，说明发生了错误。

devcon_cache_invalidate

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该示例应用程序演示在使用 DMA 向存储器中传入数据时在怎样的情况下会出现高速缓存一致性问题。MPLAB Harmony 器件控制系统服务包含了各种高速缓存函数，该示例中使用它们来解决这种问题。

该应用程序会先分配三个缓冲区 —— 两个源缓冲区和一个目标缓冲区。两个源缓冲区中填入两组不同的数据。使用 DMA 将第一个缓冲区复制到目标缓冲区，该应用程序会进行检查以确保它们包含相同的数据 （确实相同）。然后，使用 DMA 将第二个源缓冲区复制到目标缓冲区，并再次比较数据。这一次的数据不匹配，同样，这也是因为缓存一致性问题。

在进行第一次 DMA 传输并读取目标缓冲区之后，数据会被拉入高速缓存。然后，第二次 DMA 传输会更新目标缓冲区中包含的数据，但只会更新 RAM 中包含的数据。在高速缓存看来，高速缓存中的陈旧数据仍然与现在处于 RAM 中的新数据匹配。高速缓存只会看到它已经存储了目标缓冲区的副本，因此它不需要在 CPU 执行第二组读操作时费力将新数据拉入自身。需要做的是通过某种方式指示 CPU 清除陈旧的高速缓存数据，并从主存储器中拉入新数据 —— 这称为“废弃”。该示例应用程序演示了解决该问题的正确技术。在第二次 DMA 传输之后，必须使用 SYS_DEVCON_DataCacheInvalidate 函数废弃目标缓冲区。这会将高速缓存中的相关数据标记为无效。现在，当 CPU 对目标缓冲区执行读操作时，会从主存储器中拉出新数据，并在送至 CPU 之前存储到高速缓存中。

如果应用程序按预期运行，红色和绿色 LED 将会点亮。只有在发生错误时，黄色 LED 才会点亮。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了器件控制系统服务高速缓存废弃演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `devcon_cache_invalidate.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/system/devcon`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`devcon_cache_invalidate.X`	`<install-dir>/apps/examples/system/devcon/devcon_cache_invalidate/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	使用 PIC32MZ EC 入门工具包演示器件控制系统服务高速缓存函数。
pic32mz_ef_sk	pic32mz_ef_sk	使用 PIC32MZ EF 入门工具包演示器件控制系统服务高速缓存函数。
pic32mz_da_sk	pic32mz_da_sk	使用 PIC32MZ DA 入门工具包演示器件控制系统服务高速缓存函数。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行演示。

说明

该应用程序演示 SYS_DEVCON_DataCacheInvalidate API 的必要性和正确使用方式。

1. 首先编译并对目标器件编程。在编译时，选择对应于器件的正确配置。
2. 观察入门工具包开发板上的 LED。如果红色和绿色 LED 都点亮，说明应用程序执行成功。如果黄色 LED 点亮，说明发生了错误。

devcon_sys_config_perf

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示提供了一个说明如何初始化服务并使用 SYS_DEVCON_PerformanceConfig API 配置最佳系统性能的示例。

该演示应用程序会执行以下操作：

- 初始化器件控制系统服务
- 调用 SYS_DEVCON_PerformanceConfig API
- 在忙循环中等待

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了器件控制系统服务演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `devcon_sys_config_perf.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/system/devcon`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`devcon_sys_config_perf.X`	`<install-dir>/apps/examples/system/devcon/devcon_sys_config_perf/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	使用 PIC32 以太网入门工具包演示各种器件控制系统服务高速缓存函数。
pic32mz_ec_sk	pic32mz_ec_sk	使用 PIC32MZ EC 入门工具包演示器件控制系统服务高速缓存函数。
pic32mz_ef_sk	pic32mz_ef_sk	使用 PIC32MZ EF 入门工具包演示器件控制系统服务高速缓存函数。

pic32mz_da_sk	pic32mz_da_sk	使用 PIC32MZ DA 入门工具包演示器件控制系统服务高速缓存函数。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行器件控制系统服务演示。

说明

该演示会初始化器件控制系统服务，并演示如何使用 SYS_DEVCON_PerformanceConfig API。

1. 选择所需的 MPLAB X IDE 项目配置：
   - pic32mz_ec_sk （对于 PIC32MZ EC 器件）
   - pic32mz_ef_sk （对于 PIC32MZ EF 器件）
   - pic32mz_da_sk （对于 PIC32MZ DA 器件）
   - pic32mx_eth_sk （对于 PIC32MX 器件）
2. 编译 MPLAB X IDE 项目中的选定配置，并通过从 Debug 菜单中选择 **Debug Main Project** 对演示板编程。该程序将编译、下载并运行。
3. 从 Debug 菜单中选择 **Pause** （暂停）。程序将在任务程序之一中暂停。
4. 要验证该器件是否已正确初始化，请选择 *Window > PIC Memory Views （PIC 存储器视图） > Peripherals （外设）*，并检查以下方面：
   - 对于 PIC32MZ，在 PRECON 寄存器中验证
   - PFMWS 设置为 2
   - PREFEN 设置为 3
   - 对于 PIC32MX，在 CHECON 寄存器中验证
   - PFMWS 设置为 2
   - PREFEN 设置为 3

DMA 系统服务示例

本主题介绍 DMA 系统服务示例。

简介

DMA 系统服务库演示应用程序帮助。

说明

该分发包包含一个用于演示 MPLAB Harmony DMA 系统服务功能的固件项目。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行该固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 DMA 系统服务库演示应用程序。

dma_crc

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该应用程序演示如何通过使用 MPLAB Harmony DMA 系统服务库来使用 PIC32 DMA 模块的特殊 CRC 计算功能。本部分介绍硬件要求，以及在 Microchip 开发工具中编译并执行演示项目的过程。

在该演示应用程序中，使用 DMA 系统服务来设置存储器到存储器的数据传输。此外，它还会使能 CRC 引擎来计算从源位置传输到目标位置的数据的 CRC。

要了解关于 MPLAB Harmony DMA 系统服务、配置 DMA 系统服务和 DMA 系统服务提供的 API 的更多信息，请参见帮助的 DMA 系统服务库部分。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 DMA 系统服务演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `dma_crc.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/examples/system/dma`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`dma_crc.X`	`<install-dir>/apps/examples/system/devcon/dma_crc/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk	pic32mz_ef_sk	该演示在 PIC32MZ EF 入门工具包板载的 PIC32MZ2048EFM144 器件上运行。该配置可用于在后台模式下使用 16 位多项式生成 CRC。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行演示。

说明

1. 编译演示应用程序。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。
2. 通过在 MPLAB X IDE 中单击 **Debug** 在调试模式下运行演示应用程序
3. LED2 将点亮。 LED2 点亮表示正在从源向目标传输数据，并在传输数据时计算 CRC。
4. 在 MPLAB X IDE 中单击 **Pause**。
5. 在 Variables （变量）窗口中，观察调试变量 `blockCrc`。 `blockCrc` 变量的值应为 0x31C3，该值是初始种子值为 0 的 16 位多项式的 CRC 值。

关于 CRC 计算的说明：各种网站上的 CRC 计算通常使用查表法来进行 CRC 计算。查表法 CRC 计算方法通常会使用额外的 0 位填充传入的报文 （对于 16 位多项式为 16 位，对于 32 位多项式为 32 位）。 PIC32 DMA CRC 发生不会填充任何额外的位。 PIC32 DMA CRC 计算引擎会严格计算输入数据缓冲区的 CRC。

要获得与网站类似的结果，必须追加额外的位；对于 16 位多项式计算为 16 位或两个零 （字节）。对于 32 位多项式计算为 32 位或四个零 （字节）。因此， DMA 系统服务 CRC 计算应用程序中的数据传输调用在数据大小中追加了多项式长度，如以下示例所示：

```
SYS_DMA_ChannelTransferAdd(channelHandle, flashBuff,
        (APP_DMA_CRC_BUFFER_SIZE + (crc.polyLength/8)) ,
        ramBuff,(APP_DMA_CRC_BUFFER_SIZE + (crc.polyLength/8)),
        (APP_DMA_CRC_BUFFER_SIZE + (crc.polyLength/8)));
```

外部存储器编程器演示

本主题介绍外部存储器编程器演示。

简介

外部存储器编程器演示应用程序帮助。

说明

本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行外部存储器编程器固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的编程器演示应用程序。

external_flash

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示实现了驻留在引导闪存中的自举程序。当自举程序在目标器件上运行时，该器件可以对外部 SPI 闪存器件 （SST25FV016B 或类似器件）编程。在对图形 PICtail 子板上的外部 SPI 闪存进行编程时，这会很有用。

该自举程序在操作上类似于 AN1388 《PIC32 自举程序》中介绍的自举程序，并与随相关的源代码归档文件提供的个人计算机应用程序配合使用。应用笔记和归档文件可从 Microchip 网站 （www.microchip.com）下载。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了外部闪存演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `external_flash.X` 项目，然后选择所需的配置。以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/programmer/external_flash`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`external_flash.X`	`<install-dir>/apps/programmer/external_flash/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
usbdevice_pic32mx_usb_sk2	pic32mx_usb_sk2	在 PIC32 USB 入门工具包 II 上演示外部 SPI 闪存编程功能。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行外部闪存演示。

说明

基于个人计算机的主机演示

使用这些配置时，请执行以下操作：

操作

自举程序的操作方式如下：

1. 选择适合于目标硬件的配置。
2. 编译并对器件编程。
3. 按住入门工具包上的 SW3，强制自举程序工作。
4. LED1 会开始闪烁，指示自举程序正在工作。如果先前烧写了某个程序，则可能需要先按住 SW3，之后再向电路板加电或复位电路板。
5. 从 AN1388 源代码归档文件中打开个人计算机主机程序。
6. 选择适当的通信路径：
   - UART—— 将波特率保留为 115,200
   - UDP—— 将 IP 地址保留为 192.168.1.11， UDP 端口保留为 6234
   - USB 设备 —— 将 VID 保留为 0x4D8， PID 保留为 0x03C
7. 单击 **Connect** 来连接自举程序并获取版本。
8. 如果连接了自举程序，个人计算机主机应用程序将会指示自举程序的版本。

此时，自举程序已准备好接受新的应用程序，用于烧写到程序闪存中。提供了一个演示应用程序，它在链接器脚本中配置为只在程序闪存中工作。

设置

创建一个包含外部 SPI 存储器数据 （如图形资源数据）的十六进制文件。

对器件编程

演示应用程序已经编译，现在可以使用自举程序将生成的十六进制文件烧写到器件中。

要将应用程序烧写到器件中：

1. 在 AN1388 介绍的个人计算机主机应用程序中，单击 **Load Hex File**。
2. 导航至要烧写到外部 SPI 闪存中的数据文件 （如图形资源数据）。选择文件，然后单击 **Open**。
3. 单击 **Erase-Program-Verify**。
4. 然后，数据会被传输给自举程序，自举程序会对外部 SPI 闪存编程。
5. 单击 **Disconnect** 释放主机应用程序。

注： 作为一种替代方法，可以按 Bootloader Library Help （自举程序库帮助）的 Bootloader Plug-in for MPLAB X IDE （用于 MPLAB X IDE 的自举程序插件）部分所述，使用可用于 MPLAB X IDE 的自举程序插件来烧写自举程序。

sqi_flash

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

使用大型位图图像或多页菜单屏幕 （需要几个图像或者大型或多个字体包等）的应用程序对于存储器的需求量很大，以用于存储其图形资源。在这些应用程序中，在片上存储这些图形资源可能会导致存储器不足或可能成本过高。解决方案是将图形资源存储到片外存储

器（如非易失性存储器），从而将片上存储器保留用于程序存储器，并支持更复杂的功能特性。可以使用应用程序中的 pic32mz_ef_sk_meb2_gfx_ext_res 配置，将外部图形资源存储到外部 SQI 闪存中。

外部存储器件的另一种用途是用于存储程序和数据。在 PIC32MZ 器件上，可以将 SQI 外设配置为在芯片内执行（eXecute-In-Place，XIP）模式下使用存储器。以这种方式配置时，外部存储器会出现在 PIC32 器件的存储空间中，正在运行的程序可以像引用任何其他存储空间（如程序闪存和 RAM）一样引用该存储器。当引用 KSEG2（高速缓存）或 KSEG3（非高速缓存）中的存储空间时， SQI 外设会自动与存储器件相连并在相应方向上传输数据，无需任何进一步的 CPU 干预。

当前应用程序 sqi_flash 用作外部存储器编程器，用于将资源刷入片外非易失性存储器，然后可以由其他应用程序访问，从而省下片上存储器供其他程序和资源使用。 sqi_flash 应用程序的 pic32mz_ef_sk_mebii_gfx_ext_res 配置要求资源以将被烧写到外部存储器中的资源文件的形式存储在外部。示例资源文件 `gfx_resources_ext.hex` 已经生成，并位于 `<install-dir>\apps\gfx\external_resources\firmware\src\system_config\pic32mz_ef_sk+meb2\` 文件夹中供用户使用。

MPLAB Harmony 图形设计器是 MPLAB Harmony 应用程序用于为应用程序配置图形设计的主要工具，它能够生成所需的资源文件，其中包含要存储在外部存储器上的所有图形图像的二进制数据。正如前面提到的，由 MHGC 生成的文件的默认名称为 `gfx_resources_ext.hex`。请参见 MPLAB X IDE 联机帮助和 *Volume II: MPLAB Harmony Configurator (MHC)*（卷 II：MPLAB Harmony 配置器（MHC））> MPLAB Harmony Graphics Composer User's Guide（MPLAB Harmony 图形设计器用户指南）部分来了解用法信息。关于创建 `gfx_resources_ext.hex` 文件的详细信息，请参见 *Applications Help（应用程序帮助）* > *Graphics Demonstrations（图形演示）* > external_resources 演示。

sqi_flash 应用程序要求将资源文件 `gfx_resources_ext.hex` 文件从计算机复制到 USB 闪存驱动器。然后，将 USB 闪存驱动器插入开发板，应用程序会在 USB 闪存驱动器中扫描该资源文件。sqi_flash 应用程序会逐个扇区地复制资源文件，并将二进制资源内容刷入 SQI 闪存外部存储器。

pic32_mz_ef_sk 配置需要一个 `image.hex` 文件，其中包含要刷入 sqi_flash 应用程序的程序和数据信息。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了编程器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `sqi_flash.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/programmer/sqi_flash`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`sqi_flash.X`	`<install-dir>/apps/programmer/sqi_flash/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk_mebii_gfx_ext_res	pic32mz_ef_sk+meb2	用于连接到PIC32MZ EF入门工具包和图形显示Powertip 4.3" 480×272电路板的多媒体扩展板 II（MEB II）的 SQI 闪存演示。PIC32MZ EF 入门工具包具有 4 MB 的 SQI 闪存器件，用作该应用程序的外部非易失性存储器。它是用于将外部图形资源刷入外部 SQI 闪存的配置。
pic32mz_ef_sk	pic32mz_ef_sk	用于 PIC32MZ EF 入门工具包的 SQI 闪存演示本身不需要任何其他接口。但是，USB闪存驱动器上的文件`image.hex`的映像名称是不同的。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包和 MEB II

MEB II 具有两种存储器设置选项：外部或内部；但是，该演示必须在内部模式下运行。如下配置 MEB II：

- EBIWE 和 LCD_PCLK （J9）必须闭合 （跳线位于 MEB II 扩展板背面）
- 确保所有 MHC 设置都定义为仅使用内部 SRAM

应用程序当前的内部/外部存储器设置可以通过选择 *Harmony Framework Configuration* > *Drivers* > *Graphics Controllers（图形控制器）*> *LCC* > *Use LCC Driver （使用 LCC 驱动程序）* > *Memory Interface Mode （存储器接口模式）*来验证。

配置 MHC

介绍演示应用程序所需的 MHC 设置。

说明

PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包与 pic32mz_ef_sk_mebii_gfx_ext_res 配置

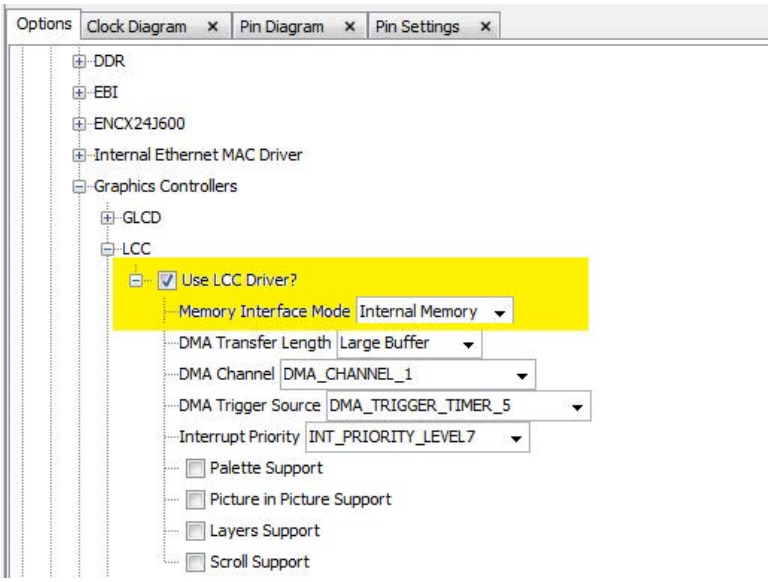
在使用 PIC32MZ EF 入门工具包并通过调试端口（J3）为其供电时，请在 JP1 上插入跳线，以确保 USB 闪存驱动器具有正确的电源。

配置 MHC

以下是一些从默认值修改为的 MHC 设置。请参见 Volume II: MPLAB Harmony Configurator (MHC) 来了解关于使用 MHC 的信息。

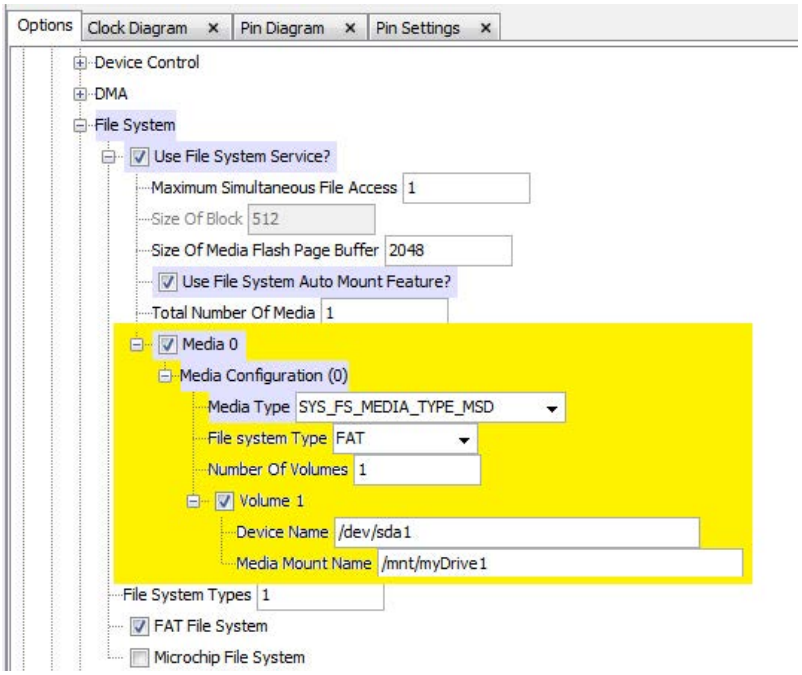
驱动程序

Graphics Controllers 选择部分显示已使用“Use LCC driver”选项选择了 LCC。所选择的 Memory Interface Mode 是 Internal Memory（内部存储器）。这是因为我们希望将程序装入内部存储器，并且使用跳线将硬件配置为与内部存储器匹配。



系统服务

选择 File System （文件系统），并使用以下设置：



BSP

为受支持的配置选择的 BSP 是 PIC32MZ EF 入门工具包与多媒体扩展板 II 组合 ——pic32mz_ef_sk_meb2。

时钟框图

打开时钟配置器部分，确保对于 pic32mz_sk_ef_meb2 配置所需的不同时钟设置选择以下值：

- 主振荡器频率：24 MHz， EC 模式
- USB PLL：480 MHz
- 系统时钟：200 MHz
- PB 时钟：100 MHz
- REFCLKO：参考时钟 （REFCLK）使用系统时钟 （SYSCLK）作为参考，并在 200 MHz 下生成用于 SPI 模块的主时钟

特定于应用程序的软件设置

SQI 外部存储器支持：由于 MPLAB Harmony 中未提供官方 SQI 驱动程序，所以外部存储器编程器使用 SQI 外设库。请注意，该项目中已经手动包含了用于支持 SQI 的源文件和头文件。

`SourceFiles/drv_nvm_flash_sqi_sst26.c HeaderFiles/drv_nvm_flash_sqi_sst26.h`

另外请注意，为了引用这些文件，在名为 SST26VF_B 的应用程序属性中添加了一条预处理器宏。

该应用程序会使用已装入说明中所提 USB 闪存驱动器上的 `gfx_resources_ext.hex` 文件。

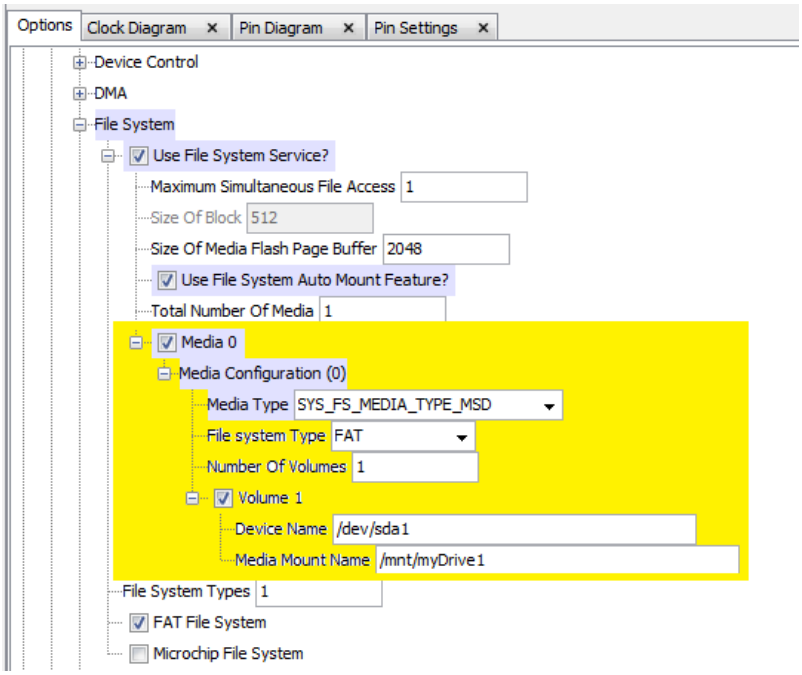
PIC32MZ EF 入门工具包与 pic32mz_ef_sk 配置

配置 MHC

以下是一些从默认值修改为的 MHC 设置。请参见 Volume II: MPLAB Harmony Configurator (MHC) 来了解关于使用 MHC 的信息。

系统服务

选择 File System，并使用以下设置：



BSP

为受支持的配置选择的 BSP 是 PIC32MZ EF 入门工具包 ——pic32mz_ef_sk。

时钟框图

打开时钟配置器部分，确保对于 pic32mz_ef_sk 配置所需的不同时钟设置选择以下值：

- 主振荡器频率：24 MHz， EC 模式
- USB PLL：480 MHz
- 系统时钟：200 MHz
- PB 时钟：100 MHz
- REFCLKO：参考时钟 （REFCLK）使用系统时钟 （SYSCLK）作为参考，并在 200 MHz 下生成用于 SPI 模块的主时钟

特定于应用程序的软件设置

SQI 外部存储器支持：由于 MPLAB Harmony 中未提供官方 SQI 驱动程序，所以外部存储器编程器使用 SQI 外设库。请注意，该项目中已经手动包含了用于支持 SQI 的源文件和头文件。

`SourceFiles/drv_nvm_flash_sqi_sst26.c HeaderFiles/drv_nvm_flash_sqi_sst26.h`

另外请注意，为了引用这些文件，在名为 SST26VF_B 的应用程序属性中添加了一条预处理器宏。

该编程器还会使用在自举程序演示中使用的相同的 Intex HEX 文件，但它并不使用自举程序库，也不使用自举程序链接器脚本。

运行演示

介绍如何编译并运行外部存储器编程器 SQI 闪存演示。

说明

sqi_flash 应用程序会将所需的 `image.hex` 文件或 `gfx_resources_ext.hex` 文件装入片外的外部存储器，以便满足需求量大的资源要求，这种资源要求在片上的内部存储器中是无法轻易满足的。外部存储器编程器的操作方式如下：

pic32mz_ef_sk_mebii_gfx_ext_res 配置：

1. sqi_flash 应用程序需要将刷入外部存储器的外部资源，它采用 HEX 文本文件的形式，默认名称为 `gfx_resources_ext.hex`。如果您没有资源二进制文件，可以从以下 MPLAB Harmony 安装文件夹中获取它：`<install-dir>\apps\gfx\external_resources\firmware\src\system_config\pic32mz_ef_sk+meb2\`。
2. 将 USB 闪存驱动器连接到计算机端口，并将 `gfx_resources_ext.hex` 文件从 `<install-dir>\apps\gfx\external_resources\firmware\src\system_config\pic32mz_ef_sk+meb2\` 复制到 USB 闪存驱动器上。
3. 将该文件复制到 USB 闪存驱动器上之后，从计算机中移除 USB 闪存驱动器，并将其插入 PIC32MZ EF 入门工具包电路板上的 USB 主机连接器
4. 使用外部存储器编程器应用程序 sqi_flash 对 PIC32MZ 器件编程。该应用程序会将 USB 闪存驱动器上的 `gfx_resources_ext.hex` 文件刷入外部 SQI 存储器。
5. 显示屏会向用户指示刷写过程开始，并在刷写完成时将状态更改为 “Finished Writing”（写入完成）。此外，绿色 LED3 也会点亮，指示操作完成。请确保当程序正在刷写存储器时，不要拔下 USB 闪存驱动器。

pic32mz_ef_sk 配置：

1. sqi_flash 应用程序需要将烧写到外部存储器中的程序 / 数据映像，它采用 HEX 文本文件的形式，默认名称为 `image.hex`。
2. 将 USB 闪存驱动器连接到计算机端口，并将 `image.hex` 文件从源目录复制到 USB 闪存驱动器上。
3. 将该文件复制到 USB 闪存驱动器上之后，从计算机中移除 USB 闪存驱动器，并将其插入 PIC32MZ EF 入门工具包电路板上的 USB 主机连接器。
4. 使用外部存储器编程器应用程序 sqi_flash 对 PIC32MZ 器件编程。该应用程序会将 USB 闪存驱动器上的 `image.hex` 文件刷入外部 SQI 存储器。
5. 绿色 LED （LED3）会点亮，指示操作完成。请确保当程序正在刷写存储器时，不要断开 USB 闪存驱动器。

文件系统演示

简介

MPLAB Harmony 文件系统演示帮助。

说明

本帮助文件包含关于 MPLAB Harmony 文件系统演示应用程序 （包含在 MPLAB Harmony 库分发包中）的说明和关联信息。

演示

说明如何运行演示应用程序。

nvm_fat_single_disk

该演示使用 NVM 闪存中包含了一个文件的 FAT12 映像，演示所有文件系统函数的工作方式。

说明

该演示给出了一个在器件闪存中实现 FAT12 磁盘的示例。该演示含有一个 FAT12 磁盘映像，该映像包含了主引导记录 （Master Boot Record，MBR）扇区、逻辑引导扇区、文件分配表和根目录区域。

该演示会打开名为 FILE.TXT 的现有文件，并对该文件执行所有文件系统相关的函数调用：SYS_FS_FileStat、 SYS_FS_FileSize、SYS_FS_FileSeek 和 SYS_FS_FileEOF。最后，会向该文件中写入字符串 “Hello World”。然后，读取该字符串并与先前写入文件中的字符串进行比较。如果字符串比较成功，则会通过 LED 指示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 NVM FAT 单磁盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `nvm_fat_single_disk.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/fs/nvm_fat_single_disk`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`nvm_fat_single_disk.X`	`<install-dir>/apps/fs/nvm_fat_single_disk/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_bt_sk_int_dyn	pic32mx_bt_sk	该配置在 PIC32 蓝牙入门工具包上运行。介质驱动程序配置为中断模式和动态操作。

pic32mx_usb_sk_int_dyn	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。介质驱动程序配置为中断模式和动态操作。
pic32mx_usb3_sk_int_dyn	pic32mx_usb_sk3	该配置在 PIC32 USB 入门工具包 III 上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 NVM FAT 单磁盘演示。

说明

选择 MPLAB X IDE 项目配置：

- pic32mx_usb_sk2_int_dyn （对于 PIC32MX 器件）
- pic32mx_bt_sk_int_dyn （对于 PIC32MX 器件）
- pic32mx_usb_sk3_int_dyn （对于 PIC32MX 器件）
- pic32mz_ec_sk_int_dyn （对于 PIC32MZ EC 器件）
- pic32mz_ef_sk_int_dyn （对于 PIC32MZ EF 器件）

编译 MPLAB X IDE 项目中的选定配置，并对演示板编程。演示的执行状态 （通过 / 失败）通过演示板上的 LED 指示。

演示板	演示成功	演示失败
PIC32 USB 入门工具包 II	LED3	LED1
PIC32 蓝牙入门工具包	绿色 LED	红色 LED
PIC32 USB 入门工具包 III	LED3	LED1
PIC32MZ EC 入门工具包 PIC32MZ EF 入门工具包	绿色 LED	红色 LED

关于演示：

该演示给出了以下方面的示例：

- 在器件闪存中实现 FAT12 磁盘
- 打开文件进行读取或写入
- 实现文件系统函数

- 关闭文件

该演示含有一个 FAT12 磁盘映像，该映像包含了主引导记录（MBR）扇区、逻辑引导扇区、文件分配表和根目录区域。该映像是在文件 `nvm_disk_images.c`（它是特定于项目配置的文件，包含在 MPLAB X IDE 项目的 `nvm_disk_images` 逻辑文件夹中）中实现的。该映像包含单个名为 `FILE.TXT` 的文件，其中包含字符串 “Data”。

该演示会打开名为 `FILE.TXT` 的现有文件，并对该文件执行所有文件系统相关的函数调用：SYS_FS_FileStat、 SYS_FS_FileSize、SYS_FS_FileSeek 和 SYS_FS_FileEOF。最后，会向该文件中写入字符串 “Hello World”。然后，读取该字符串并与先前写入文件中的字符串进行比较。如果字符串比较成功，则会通过 LED 指示。

演示应用程序逻辑以状态机方式在 `main.c` 文件的 APP_Tasks 函数中实现。

1. 磁盘先使用 SYS_FS_Mount 函数挂载。`/dev/nvma1` 路径指示挂载命令挂载一个内部闪存卷。该卷挂载在 FAT 类型的文件系统下，并挂载在 `/mnt/myDrive/` 处。
2. 如果挂载成功，应用程序会通过 SYS_FS_FileOpen 函数打开文件 `FILE.TXT` 进行读写（SYS_FS_FILE_OPEN_READ_PLUS）。打开文件成功后，会收到有效的文件句柄。
3. 如果文件打开成功，文件 `FILE.TXT` 的状态会使用 SYS_FS_FileStat 函数存储在 appData.fileStatus 结构中。
4. 如果文件状态检查成功，则通过将文件句柄传递给 SYS_FS_FileSize 函数来检查 `FILE.TXT` 的大小。
5. 如果文件大小检查成功，则将文件的大小与先前作为 appData.fileStatus 结构一部分接收到的大小元素进行比较。如果这两个值匹配，则该代码转到下一步，即文件游标移动。
6. 然后，通过调用函数 SYS_FS_FileSeek 并将参数作为 SYS_FS_SEEK_SET 传递 （从文件起始位置进行游标移动）来将文件指针相对于文件起始位置移动 4 个字节。
7. 如果文件游标移动操作成功，文件指针将到达文件的末尾。这是因为 `FILE.TXT` 的内容只有一个 4 字节的字符串 “Data”。通过调用 SYS_FS_FileEOF 函数验证是否为文件末尾。如果该函数返回 “true”（到达文件末尾），则执行下一步。
8. 在下一步中，通过调用函数 SYS_FS_FileSeek 并传递 SYS_FS_SEEK_END 参数 （从文件末尾进行游标移动）再次将文件指针相对于文件末尾移动，移动量为 [(-1)* 文件大小]。
9. 如果文件游标移动操作成功，文件指针将到达文件的起始位置。然后，使用 SYS_FS_FileRead 函数从文件中读取 4 字节 （读到缓冲区中）。
10. 如果读操作成功，则将文件的内容 （位于缓冲区中）与 “Data” 字符串进行比较。如果比较通过，则该代码转到下一步。
11. 在下一步中，通过调用函数 SYS_FS_FileSeek 并传递 SYS_FS_SEEK_END 参数 （从文件末尾进行游标移动）再次将文件指针相对于文件末尾移动，移动量为 [(-1)* 文件大小]。
12. 如果文件游标移动操作成功，则使用 SYS_FS_FileWrite 函数将字符串 “Hello World” 写入文件。
13. 如果写操作成功，则将文件指针移至文件的起始位置。
14. 如果文件游标移动成功，则使用 SYS_FS_FileRead 函数读取文件的内容 （读到缓冲区中）。
15. 如果读操作成功，则使用 strcmp 函数将文件的内容 （位于缓冲区中）与 “Hello World” 字符串进行比较。 LED 将指示演示成功。

nvm_mpfs_single_disk

该演示使用 NVM 闪存中包含了两个文件的 MPFS 映像，演示所有文件系统函数的工作方式。

说明

该演示给出了一个在器件闪存中实现 MPFS 磁盘的示例。该演示含有一个处于内部闪存中的 MPFS 磁盘映像。该磁盘映像包含两个文件，名为：

- `FILE.txt`，大小 = 11 字节。该文件的内容为：“Hello World”。
- `TEST.txt`，大小 = 72 字节。该文件的内容为：“This file contains a test string and it is meant for testing.1234567890”。

该演示会对文件执行所有文件系统相关的函数调用：SYS_FS_FileRead、SYS_FS_FileStat、SYS_FS_FileSize、SYS_FS_FileSeek 和 SYS_FS_FileEOF. 如果所有测试均成功，则会通过 LED 指示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 NVM MPFS 单磁盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 nvm_mpfs_single_disk.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/fs/nvm_mpfs_single_disk。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
nvm_mpfs_single_disk.X	<install-dir>/apps/fs/nvm_mpfs_single_disk/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_bt_sk_int_dyn	pic32mx_bt_sk	该配置在 PIC32 蓝牙入门工具包上运行。介质驱动程序配置为中断模式和动态操作。
pic32mx_usb_sk_int_dyn	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。介质驱动程序配置为中断模式和动态操作。
pic32mx_usb3_sk_int_dyn	pic32mx_usb_sk3	该配置在 PIC32 USB 入门工具包 III 上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

不需要硬件相关配置或跳线设置更改。

PIC32 蓝牙入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 NVM MPFS 单磁盘演示。

说明

选择 MPLAB X IDE 项目配置：

- pic32mx_usb_sk2_int_dyn （对于 PIC32MX 器件）

- pic32mx_usb_sk3_int_dyn （对于 PIC32MX 器件）
- pic32mx_bt_sk_int_dyn （对于 PIC32MX 器件）
- pic32mz_ec_sk_int_dyn （对于 PIC32MZ EC 器件）
- pic32mz_ef_sk_int_dyn （对于 PIC32MZ EF 器件）

编译 MPLAB X IDE 项目中的选定配置，并对演示板编程。演示的执行状态 （通过 / 失败）通过演示板上的 LED 指示。

演示板	演示成功	演示失败
PIC32 USB 入门工具包 II PIC32 USB 入门工具包 III	LED3	LED1
PIC32 蓝牙入门工具包	绿色 LED	红色 LED
PIC32MZ EC 入门工具包 PIC32MZ EF 入门工具包	绿色 LED	红色 LED

关于演示：

该演示给出了以下方面的示例：

- 在器件闪存中实现 MPFS 磁盘
- 打开文件进行读取
- 实现所有文件系统函数
- 关闭文件

该演示给出了一个在器件闪存中实现 MPFS 磁盘的示例。该演示含有一个处于内部闪存中的 MPFS 磁盘映像。该磁盘映像包含两个文件，名为：

- `FILE.txt`，大小 = 11 字节。该文件的内容为：“Hello World”。
- `TEST.txt`，大小 = 72 字节。该文件的内容为：“This file contains a test string and it is meant for testing.1234567890”。

演示应用程序逻辑以状态机方式在 `main.c` 文件的 APP_Tasks 函数中实现。

1. 磁盘先使用 SYS_FS_Mount 函数挂载。 `/dev/nvma1` 路径指示挂载命令挂载一个内部闪存卷。该卷挂载在 MPFS2 类型的文件系统下，并挂载在 `/mnt/myDrive/` 处。
2. 如果挂载成功，应用程序会通过 SYS_FS_FileOpen 函数打开文件 `FILE.txt` 进行读取。
3. 如果打开成功，应用程序会通过 SYS_FS_FileOpen 函数打开另一个文件 `TEST.txt` 进行读取。
4. 如果打开成功，应用程序通过向 SYS_FS_FileSize 函数传递在文件打开过程中获得的句柄来检查文件 `FILE.txt` 的大小。
5. 如果文件大小与 11 字节的已知值匹配，则应用程序使用 SYS_FS_FileSeek 函数将文件 `TEST.txt` 的文件指针相对于文件末尾移动 10 个字节。
6. 如果文件游标移动成功，则使用 SYS_FS_FileRead 函数将文件 `TEST.txt` 的 10 字节内容读到应用程序缓冲区中。
7. 如果读操作成功，则使用 strncmp 函数将应用程序缓冲区内容与已知字符串 `1234567890` 进行比较。
8. 如果比较成功，则应用程序使用 SYS_FS_FileEOF 函数检查文件 “`TEST.txt`”的文件指针是否已到达文件末尾。
9. 如果已到达文件末尾， LED 会指示演示成功。

nvm_sdcard_fat_mpfs_multi_disk

该演示使用具有 MPFS 和 FAT 文件映像的 NVM 和安全数字 （SD）卡，执行从一个介质上的文件到另一个介质的读 / 写 / 验证操作。

说明

该演示给出了一个使用 MPLAB Harmony 文件系统来访问多个磁盘和多个文件系统上的文件的示例。该演示含有一个处于内部闪存中的 MPFS 磁盘映像。该磁盘映像包含一个名为 `abc.txt` 的文件，其内容为“Hello World”。另一个磁盘为 SD 卡，它被格式化为 FAT（FAT16 或 FAT32）。该演示会从在内部闪存上实现的磁盘中读取 `abc.txt` 的内容，并将该内容写入 SD 卡上的 `FILE.TXT`。成功写入后，会通过点亮 LED 指示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 NVM SD 卡 FAT MPFS 多磁盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `nvm_sdcard_fat_mpfs_multi_disk.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/fs/nvm_sdcard_fat_mpfs_multi_disk`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`nvm_sdcard_fat_mpfs_multi_disk.X`	`<install-dir>/apps/fs/nvm_sdcard_fat_mpfs_multi_disk/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16_int_dyn	pic32mx795_pim+e16	该配置在使用 PIC32MX795F512L PIM 以及用于 SD 和 MMC 的 PICtail 子板的 Explorer 16 开发板上运行。介质驱动程序配置为中断模式和动态操作。
pic32mx470_pim_e16_int_dyn	pic32mx470_pim+e16	该配置在使用 PIC32MX470F512L PIM 以及用于 SD 和 MMC 的 PICtail 子板的 Explorer 16 开发板上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk+meb2	该配置在连接到 MEB II 的 PIC32MZ EC 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk+meb2	该配置在连接到 MEB II 的 PIC32MZ EF 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。

配置硬件

介绍如何配置支持的硬件。

说明

Explorer 16开发板、PIC32MX450/470F512L接插模块（PIM）或PIC32MX795F512L接插模块（PIM）和用于SD和MMC的PICtail子板

对于所有基于 Explorer 16 开发板的演示板，请根据以下指示信息进行操作。

由于一些外设功能通过外设引脚选择 （PPS）功能进行复用，所以必须通过连接 PIM 引脚来修改以下 PIM 上的硬件，如下：

PIC32MX470F512L PIM （MA320002-2）——PIM 引脚 99 至 PIM 引脚 24 用于 CS

关于 PIM 引脚位置，请参见 PIC32MX450/470F512L 接插模块 （PIM）或 PIC32MX795F512L 接插模块 （PIM）。

对于两种 PIM，均使用以下通用测试设置：

1. 将 PIM 插入 Explorer 16 开发板 PIM 连接器。
2. 用于SD和MMC的PICtail子板上的跳线JP1、JP2和JP3应连接到其相应连接器的接点1和2（见用于SD和MMC的PICtail子板）。

3. 将子板插入 Explorer 16 开发板上的 PICtail Plus 连接器。
4. 将 SD 卡插入子板上的 SD 卡连接器。
5. 将电路板上电。

PIC32MZ EC 入门工具包和 MEB II

不需要任何硬件设置更改。将 microSD 卡插入连接器并将电路板上电。

PIC32MZ EF 入门工具包和 MEB II

不需要任何硬件设置更改。将 microSD 卡插入连接器并将电路板上电。

运行演示

介绍如何编译并运行 NVM SD 卡 FAT MPFS 多磁盘演示。

说明

选择 MPLAB X IDE 项目配置：

- pic32mx795_pim_e16_int_dyn （对于 PIC32MX 器件）
- pic32mx470_pim_e16_int_dyn （对于 PIC32MX 器件）
- pic32mz_ec_sk_int_dyn （对于 PIC32MZ EC 器件）
- pic32mz_ef_sk_int_dyn （对于 PIC32MZ EF 器件）

插入 SD 卡，其中包含文件 `FILE.TXT` （该文件包含一些任意的现有内容）。

编译 MPLAB X IDE 项目中的选定配置，并运行代码。在几秒钟之后，当 LED 点亮时，从 SD 卡 PICtail 子板中移除 SD 卡，并将其插入个人计算机上的 SD 卡读卡器。

检查名为 `FILE.TXT` 的文件的内容。该文件应包含字符串 “Hello World”。

演示板	演示成功	演示失败
Explorer 16 开发板	LED5 （D5）	LED6 （D6）
PIC32MZ EC 入门工具包 PIC32MZ EF 入门工具包	绿色 LED	红色 LED

关于演示：

该演示给出了以下方面的示例：

- 实现多磁盘演示， MPFS 位于内部闪存 （NVM）上， FAT 文件系统 （FAT16/FAT32）位于 SD 卡上
- 从两个不同磁盘打开两个文件，从第一个磁盘读取文件的内容，并写入到第二个磁盘的文件中
- 关闭文件

该演示含有一个处于内部闪存中的 MPFS 磁盘映像。该磁盘映像包含一个名为 `abc.txt` 的文件，其内容为 “Hello World”。另一个磁盘为 SD 卡，它被格式化为 FAT （FAT16 或 FAT32）。该演示会从在内部闪存上实现的磁盘中读取 `abc.txt` 的内容，并将该内容写入 SD 卡上的 `FILE.TXT`。成功写入后，会通过点亮 LED 指示。

演示应用程序逻辑以状态机方式在 `main.c` 文件的 APP_Tasks 函数中实现。

1. 使用 SYS_FS_Mount 函数挂载第一个磁盘。 `/dev/nvma1` 路径指示挂载命令挂载一个内部闪存卷。该卷挂载在 MPFS2 类型的文件系统下，并挂载在 `/mnt/myDrive1/` 处。
2. 如果挂载成功，则使用 SYS_FS_Mount 函数挂载第二个磁盘。`/dev/mmcblka1` 路径指示挂载命令挂载一个 SD 卡卷。该卷挂载在 FAT 类型的文件系统下，并挂载在 `/mnt/myDrive2/` 处。
3. 如果挂载成功，应用程序会使用SYS_FS_FileOpen函数从`/mnt/myDrive1/`中打开文件`abc.txt`进行读取，从`/mnt/myDrive2/`中打开 `FILE.TXT` 进行写入。
4. 如果文件打开成功，应用程序会使用 SYS_FS_FileRead 从内部闪存卷的 `abc.txt` 中读取 13 个字节。
5. 如果读操作成功，应用程序会从内部闪存卷中关闭 `abc.txt`，然后使用 SYS_FS_FileWrite 函数将 13 个字节写入 SD 卡卷的 `FILE.TXT`。如果文件写操作成功，应用程序会从 SD 卡卷中关闭 `FILE.TXT`。
6. 如果文件关闭成功， LED 会指示操作成功。

nvm_sdcard_fat_multi_disk

该演示使用 NVM 和安全数字 （Secure Digital，SD）卡作为介质，在 NVM 介质上搜索某个文件，打开并读取该文件，然后将数据写入 SD 卡介质上的另一个文件中。

说明

该演示给出了一个使用 MPLAB Harmony 文件系统来访问多个介质上的文件的示例。该演示含有一个 FAT12 磁盘映像，该映像包含主引导记录 （MBR）扇区、逻辑引导扇区、文件分配表和根目录区域，它被放置在内部闪存 （NVM）中。此外，使用 SD 卡作为另一个磁盘，它可能实现 FAT16 或 FAT32（取决于 SD 卡的格式）。该演示会在 NVM 介质中搜索名为 `FILE.TXT` 的文件，打开并读取 NVM 中该文件的内容，并将文件内容复制到 SD 卡上的文件 `FILE.TXT` 中。在复制成功后，会在文件中另外添加一个字符串 “Test is successful”。如果写操作成功，则会通过 LED 指示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 NVM SD 卡 FAT 多磁盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `nvm_sdcard_fat_multi_disk.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/fs/nvm_sdcard_fat_multi_disk`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`nvm_sdcard_fat_multi_disk.X`	`<install-dir>/apps/fs/nvm_sdcard_fat_multi_disk/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16_int_dyn	pic32mx795_pim+e16	该配置在使用 PIC32MX795F512L PIM 以及用于 SD 和 MMC 的 PICtail 子板的 Explorer 16 开发板上运行。介质驱动程序配置为中断模式和动态操作。
pic32mx470_pim_e16_int_dyn	pic32mx470_pim+e16	该配置在使用 PIC32MX470F512L PIM 以及用于 SD 和 MMC 的 PICtail 子板的 Explorer 16 开发板上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk+meb2	该配置在连接到 MEB II 的 PIC32MZ EC 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk+meb2	该配置在连接到 MEB II 的 PIC32MZ EF 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。

配置硬件

介绍如何配置支持的硬件。

说明

Explorer 16开发板、PIC32MX450/470F512L接插模块（PIM）或PIC32MX795F512L接插模块（PIM）和用于SD和MMC的PICtail子板

对于所有基于 Explorer 16 开发板的演示板，请根据以下指示信息进行操作。

由于一些外设功能通过外设引脚选择 （PPS）功能进行复用，所以必须通过连接 PIM 引脚来修改以下 PIM 上的硬件，如下：

PIC32MX470F512L PIM （MA320002-2）——PIM 引脚 99 至 PIM 引脚 24 用于 CS

关于 PIM 引脚位置，请参见 PIC32MX450/470F512L 接插模块 （PIM）或 PIC32MX795F512L 接插模块 （PIM）。

对于两种 PIM，均使用以下通用测试设置：

1. 将 PIM 插入 Explorer 16 开发板 PIM 连接器。
2. 用于SD和MMC的PICtail子板上的跳线JP1、JP2和JP3应连接到其相应连接器的接点1和2（见用于SD和MMC的PICtail子板）。
3. 将子板插入 Explorer 16 开发板上的 PICtail Plus 连接器。
4. 将 SD 卡插入子板上的 SD 卡连接器。
5. 将电路板上电。

PIC32MZ EC 入门工具包和 MEB II

不需要任何硬件设置更改。将 microSD 卡插入连接器并将电路板上电。

PIC32MZ EF 入门工具包和 MEB II

不需要任何硬件设置更改。将 microSD 卡插入连接器并将电路板上电。

运行演示

介绍如何编译并运行 NVM SD 卡 FAT 多磁盘演示。

说明

选择 MPLAB X IDE 项目配置：

- pic32mx795_pim_e16_int_dyn （对于 PIC32MX 器件）
- pic32mx470_pim_e16_int_dyn （对于 PIC32MX 器件）
- pic32mz_ec_sk_int_dyn （对于 PIC32MZ EC 器件）
- pic32mz_ef_sk_int_dyn （对于 PIC32MZ EF 器件）

插入 SD 卡，其中包含文件 `FILE.TXT`，该文件包含 “abc”（一些任意的现有内容）。

编译 MPLAB X IDE 项目中的选定配置，并运行代码。在几秒钟之后，当 LED 点亮时，从电路板上移除 SD 卡，并将其插入个人计算机上的 SD 卡读卡器。检查名为 `FILE.TXT` 的文件的内容。该文件应包含字符串 “This data from NVM Disk Test is successful”。

演示板	演示成功	演示失败
Explorer 16 开发板	LED5 （D5）	LED6 （D6）
PIC32MZ EC 入门工具包 PIC32MZ EF 入门工具包	绿色 LED	红色 LED

关于演示：

该演示给出了以下方面的示例：

- 实现多磁盘演示， FAT12 位于内部闪存 （NVM）上， FAT 文件系统 （FAT16/FAT32）位于 SD 卡上
- 从两个不同磁盘打开两个文件，从第一个磁盘读取文件的内容，并写入到第二个磁盘的文件中
- 关闭文件

该演示含有一个 FAT12 磁盘映像，该映像包含主引导记录 （MBR）扇区、逻辑引导扇区、文件分配表和根目录区域，它被放置在内部闪存中 （NVM）。此外，使用 SD 卡作为另一个磁盘，它可能实现 FAT16 或 FAT32 （取决于 SD 卡的格式）。该演示会在 NVM 介质中搜索名为 `FILE.TXT` 的文件，打开并读取 NVM 中该文件的内容，并将文件内容复制到 SD 卡上的 `FILE.TXT` 中。在复制成功后，会在文件中另外添加一个字符串 “Test is successful”。如果写操作成功，则会通过 LED 指示。

演示应用程序逻辑以状态机方式在 `main.c` 文件的 APP_Tasks 函数中实现。

1. 使用 SYS_FS_Mount 函数挂载第一个磁盘。/dev/nvma1 路径指示挂载命令挂载一个内部闪存卷。该卷挂载在 FAT 类型的文件系统下，并挂载在 /mnt/myDrive1/ 处。
2. 如果挂载成功，则使用 SYS_FS_Mount 函数挂载第二个磁盘。/dev/mmcblka1 路径指示挂载命令挂载一个 SD 卡卷。该卷挂载在 FAT 类型的文件系统下，并挂载在 /mnt/myDrive2/ 处。
3. 如果挂载成功，应用程序会使用 SYS_FS_DirOpen 函数打开 /mnt/myDrive1/ 的根目录。
4. 如果目录打开成功，应用程序会使用通配符 FIL*.* 搜索文件 FILE.TXT。用于目录搜索的函数为 SYS_FS_DirSearch。
5. 如果目录搜索返回找到的文件，则使用 SYS_FS_DirClose 函数关闭目录。此外，将搜索到的文件名与 FILE.TXT 名称进行比较。
6. 如果文件名匹配成功，应用程序会使用 SYS_FS_FileOpen 函数从 /mnt/myDrive1/ 中打开文件 FILE.txt 进行读取，从 /mnt/myDrive2/ 中打开 FILE.TXT 进行写入。
7. 如果文件打开成功，应用程序会使用 SYS_FS_FileRead 从内部闪存卷的 FILE.TXT 中读取 27 个字节。
8. 如果读操作成功，应用程序会从内部闪存卷中关闭FILE.TXT，然后使用SYS_FS_FileWrite将27个字节写入SD卡卷的FILE.TXT。
9. 如果写操作成功，则使用 SYS_FS_FileCharacterPut 和 SYS_FS_FileStringPut 向文件中写入一个字符和一个字符串。
10. 如果写操作成功，应用程序会从 SD 卡卷中关闭 FILE.TXT。
11. 如果文件关闭成功， LED 会指示操作成功。

sdcard_fat_single_disk

该演示使用具有 FAT 文件系统的安全数字 （SD）卡作为介质，对使用长文件名 （Long File Name，LFN）的文件执行读 / 写 / 验证操作，并创建目录。

说明

该演示给出了一个使用 MPLAB Harmony 文件系统来访问并修改 SD 卡内容的示例。该演示会打开 SD 卡上名为 FILE_TOO_LONG_NAME_EXAMPLE_123.JPG 的文件，读取其内容，创建一个名为 Dir1 的目录，并在该目录内将文件内容写入到另一个文件 FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG （在一个目录内，在一个文件中创建另一个文件的副本）。

输入文件 FILE_TOO_LONG_NAME_EXAMPLE_123.JPG 未随发行软件包提供。它可以是用户选择的任意 JPEG （图像）文件，然后重命名为 FILE_TOO_LONG_NAME_EXAMPLE_123.JPG。选择 JPEG 文件用于测试的原因在于，通过将 SD 卡插入计算机并打开 FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG 文件，可轻松验证由 FS 演示创建的副本文件 FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG 的正确性。如果文件可以在计算机上打开供查看，则认为测试已通过。否则，如果文件未打开 （即，已损坏），则认为测试失败。由于演示会创建一个名为 Dir1 的目录， SD 卡上不存在同名文件夹是很重要的。如果 SD 卡上已存在名为 Dir1 的目录，演示会失败。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SD 卡 FAT 单磁盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 sdcard_fat_single_disk.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/fs/sdcard_fat_single_disk。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
sdcard_fat_single_disk.X	<install-dir>/apps/fs/sdcard_fat_single_disk/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16_int_dyn	pic32mx795_pim+e16	该配置在使用 PIC32MX795F512L PIM 以及用于 SD 和 MMC 的 PICtail 子板的 Explorer 16 开发板上运行。介质驱动程序配置为中断模式和动态操作。
pic32mx470_pim_e16_int_dyn	pic32mx470_pim+e16	该配置在使用 PIC32MX470F512L PIM 以及用于 SD 和 MMC 的 PICtail 子板的 Explorer 16 开发板上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk+meb2	该配置在连接到 MEB II 的 PIC32MZ EC 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk+meb2	该配置在连接到 MEB II 的 PIC32MZ EF 入门工具包上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_da_sk_adma	pic32mz_da_sk	该配置在 PIC32MZ DA 入门工具包上运行。介质驱动程序配置为使用 SD 主机控制器 ADMA2 传输模式操作。

配置硬件

介绍如何配置支持的硬件。

说明

Explorer 16开发板、PIC32MX450/470F512L接插模块（PIM）或PIC32MX795F512L接插模块（PIM）和用于SD和MMC的PICtail子板

对于所有基于 Explorer 16 开发板的演示板，请根据以下指示信息进行操作。

由于一些外设功能通过外设引脚选择 （PPS）功能进行复用，所以必须通过连接 PIM 引脚来修改以下 PIM 上的硬件，如下：

PIC32MX470F512L PIM （MA320002-2）——PIM 引脚 99 至 PIM 引脚 24 用于 CS

关于 PIM 引脚位置，请参见 PIC32MX450/470F512L 接插模块 （PIM）或 PIC32MX795F512L 接插模块 （PIM）。对于两种 PIM，均使用以下通用测试设置：

1. 将 PIM 插入 Explorer 16 开发板 PIM 连接器。
2. 用于SD和MMC的PICtail子板上的跳线JP1、JP2和JP3应连接到其相应连接器的接点1和2（见用于SD和MMC的PICtail子板）。
3. 将子板插入 Explorer 16 开发板上的 PICtail Plus 连接器。
4. 将 SD 卡插入子板上的 SD 卡连接器。
5. 将电路板上电。

PIC32MZ EC 入门工具包和 MEB II

不需要任何硬件设置更改。将 microSD 卡插入连接器并将电路板上电。

PIC32MZ EF 入门工具包和 MEB II

不需要任何硬件设置更改。将 microSD 卡插入连接器并将电路板上电。

PIC32MZ DA 入门工具包和 MEB II

不需要任何硬件设置更改。

运行演示

介绍如何编译并运行 SD 卡 FAT 单磁盘演示。

说明

选择 MPLAB X IDE 项目配置：

- pic32mx795_pim_e16_int_dyn （对于 PIC32MX 器件）
- pic32mx470_pim_e16_int_dyn （对于 PIC32MX 器件）
- pic32mz_ec_sk_int_dyn （对于 PIC32MZ EC 器件）
- pic32mz_ef_sk_int_dyn （对于 PIC32MZ EF 器件）
- pic32mz_da_sk_adma （对于 PIC32MZ DA 器件）

插入 SD 卡，其中包含名为 `FILE_TOO_LONG_NAME_EXAMPLE_123.JPG` 的文件。该文件可以为任意大小。

编译 MPLAB X IDE 项目中的选定配置，并运行代码。在几秒钟之后，当 LED 点亮时，从 SD 卡 PICtail 子板（对于 PIC32MX）或 MEB II（对于 PIC32MZ）中移除 SD 卡，并将其插入个人计算机上的 SD 卡读卡器。在检查 SD 卡的内容时，目录 `Dir1` 将已创建。在 `Dir1` 文件夹中，将存在名为 `FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG` 的文件。文件 `FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG` 将为 `FILE_TOO_LONG_NAME_EXAMPLE_123.JPG` 的副本。验证这两个文件都可以在个人计算机上打开供查看。

演示板	演示成功	演示失败
Explorer 16 开发板	LED5 （D5）	LED6 （D6）
PIC32MZ 嵌入式连接 （EC）入门工具包	绿色 LED	红色 LED

关于演示：

该演示给出了以下方面的示例：

- 在 SD 卡上实现 FAT 文件系统 （FAT16/FAT32）
- 实现长文件名 （LFN）
- 打开文件进行读取或写入
- 关闭文件

该演示会打开 SD 卡上名为 `FILE_TOO_LONG_NAME_EXAMPLE_123.JPG` 的文件，读取其内容，创建一个名为 `Dir1` 的目录，并在该目录内将文件内容写入到另一个文件 `FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG`（在一个目录内，在一个文件中创建另一个文件的副本）。

演示应用程序逻辑以状态机方式在 `main.c` 文件的 APP_Tasks 函数中实现。

1. 磁盘先使用 SYS_FS_Mount 函数挂载。 `/dev/mmcblka1` 路径指示挂载命令挂载一个 SD 卡卷。该卷挂载在 FAT 类型的文件系统下，并挂载在 `/mnt/myDrive/` 处。
2. 如果挂载成功，则通过向 SYS_FS_Unmount 函数传递挂载名称 `/mnt/myDrive` 来卸载该卷。执行该卸载操作只是为了进行演示。除非确实需要，否则真实的应用程序并不需要进行卸载。
3. 如果卸载成功，则重复挂载过程。
4. 如果挂载成功，应用程序会通过 SYS_FS_FileOpen 函数打开文件 `FILE_TOO_LONG_NAME_EXAMPLE_123.JPG` 进行读取。
5. 如果文件打开成功，应用程序会使用 SYS_FS_DirectoryMake 函数创建一个名为 `Dir1` 的目录。
6. 如果目录创建成功，应用程序会通过 SYS_FS_FileOpen 函数打开目录 `Dir1` 内的文件 `FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG` 进行写入。文件写操作的属性选择为 “SYS_FS_FILE_OPEN_WRITE”。因此，如果该文件不存在，将创建该文件。
7. 如果文件打开成功，则使用 SYS_FS_FileRead 函数将文件 `FILE_TOO_LONG_NAME_EXAMPLE_123.JPG` 的 512 字节内容读取到应用程序缓冲区中。
8. 如果文件读取成功，则使用 SYS_FS_FileWrite 函数从应用程序缓冲区中将 512 字节写入到 `FILE_TOO_LONG_NAME_EXAMPLE_123_1.JPG` 文件。
9. 如果写操作成功，则检查是否到达 `FILE_TOO_LONG_NAME_EXAMPLE_123.JPG` 的文件末尾。如果未到达文件末尾，则继续执行读取和写入过程 （步骤 7 和步骤 8）。
10. 在到达文件末尾时，使用 SYS_FS_FileClose 函数关闭这两个文件。
11. 最后， LED 会指示演示成功。

sdcard_msd_fat_multi_disk

该演示使用 USB 闪存驱动器和安全数字 （SD）卡作为介质。应用程序会在 USB 闪存驱动器上搜索某个文件，打开并读取该文件，然后将数据写入 SD 卡介质上的另一个文件中。

说明

该演示会使用通配符 “`mch*.*`” 搜索一个文件，读取该文件的内容，然后将该文件的内容写入 SD 卡。

演示应用程序逻辑以状态机方式在 app.c 文件的 APP_USB_MSDTasks 和 APP_SDCardTasks 函数中实现。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SD 卡 MSD FAT 多磁盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 sdcard_msd_fat_multi_disk.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/fs/sdcard_msd_fat_multi_disk。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
sdcard_msd_fat_multi_disk.X	<install-dir>/apps/fs/sdcard_msd_fat_multi_disk/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包与 MEB II 上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包与 MEB II 上运行。介质驱动程序配置为中断模式和动态操作。
pic32mz_da_sk_meb2	pic32mz_da_sk+meb2	该配置在 PIC32MZ DA 入门工具包与 MEB II 上运行。介质驱动程序配置为中断模式和动态操作。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SD 卡 MSD FAT 多磁盘演示。

说明

该演示给出了以下方面的示例：

- 在 SD 卡和 USB 闪存驱动器上实现 FAT 文件系统
- 打开文件进行读取或写入
- 搜索文件

- 检查是否到达文件末尾
- 关闭文件

执行以下操作来运行演示：

1. 选择所用硬件所需的 MPLAB X IDE 项目配置。
2. 将 `mchpLogo.bmp` 文件从 `<install-dir>/apps/fs/sdcard_msd_fat_multi_disk/firmware/src` 复制到 USB 闪存驱动器中。
3. 将 SD 卡插入 MEB II。
4. 将 USB 闪存驱动器卡插入入门工具包。
5. 编译 MPLAB X IDE 项目中的选定配置，并运行代码。
6. 该演示会在 USB 闪存驱动器上搜索 `mchpLogo.bmp`，并将它从 USB 闪存驱动器复制到 SD 卡中。绿色 LED 指示该文件已成功从 USB 闪存驱动器复制到 SD 卡。
7. 从 MEB II 板上移除 SD 卡，并将它连接到个人计算机，以验证 `mchpLogo.bmp` 文件在 SD 卡上可用。

演示板	演示成功	演示失败
PIC32MZ EC 入门工具包 PIC32MZ EF 入门工具包 PIC32MZ DA 入门工具包	绿色 LED 点亮	红色 LED 点亮

该演示会使用通配符 “`mch*.*`” 搜索一个文件，读取该文件的内容，然后将该文件的内容写入 SD 卡。

演示应用程序逻辑以状态机方式在 `app.c` 文件的 APP_USB_MSDTasks 和 APP_SDCardTasks 函数中实现。

演示过程如下：

1. 使用 SYS_FS_Mount 函数挂载 SD 卡。 `/dev/mmcblka1` 路径指示挂载函数挂载一个 SD 卡卷。该卷使用 FAT 文件系统挂载，并挂载为 `/mnt/sdDrive/`。
2. 如果挂载成功， SD 卡状态机会一直等待，直到 USB 闪存驱动器连接并挂载。
3. USB MSD 任务函数会打开 USB 主机协议栈的一个实例，并在挂载 USB 闪存驱动器之前使能 USB 主机操作。
4. 当连接 USB 闪存驱动器后，立即使用 SYS_FS_Mount 函数挂载它。 `/dev/sda1` 路径指示挂载函数挂载一个 SD USB 闪存驱动器卷。该卷使用 FAT 文件系统挂载为 `/mnt/msdDrive/`。
5. 如果 USB 闪存驱动器挂载成功，应用程序会打开 USB 闪存驱动器上的根目录，并通过 SYS_FS_DirSearch 使用通配符 “`mch*.*`” 搜索文件。
6. 如果搜索操作成功，应用程序会使用 SYS_FS_FileOpen 以读模式打开文件。
7. 如果文件打开成功，应用程序会使用 SYS_FS_CurrentDriveSet 将当前工作目录设置为 `sdDrive`，并以写模式在 SD 卡上创建一个新文件。新文件的名称与从 SYS_FS_DirSearch 返回的相同。
8. 如果文件创建成功，应用程序会使用 SYS_FS_FileRead 从 USB 闪存驱动器上的文件中读取数据，并使用 SYS_FS_FileWrite 将其写入 SD 卡上的文件中，直到到达文件末尾。
9. 在到达文件末尾时，使用 SYS_FS_FileClose 函数关闭这两个文件。
10. 绿色 LED 指示操作成功。

图形演示

本主题介绍图形演示。

简介

图形库演示应用程序帮助

说明

该分发包包含一系列与图形相关的固件项目，用于演示 MPLAB Harmony 图形库的功能。本帮助文件介绍硬件要求，以及在 Microchip 图形板上运行这些固件项目的过程。

本帮助文件包含关于 MPLAB Harmony 图形演示应用程序（包含在 MPLAB Harmony 库分发包中）的说明和关联信息。请参见 *Volume I: Getting Started With MPLAB Harmony （卷 I：MPLAB Harmony 入门）* > *Release Information （发行信息）* > Release Notes （发行说明）来了解与特定应用程序关联的发行信息。

要了解关于 MPLAB Harmony 图形库、配置该库和所提供的 API 的更多信息，请参见图形库文档。

图形演示	说明
lcc	在使用板载的基于软件的图形控制器时，演示 alpha 混合、性能、图像和动画。
object	演示图形库支持的各种用户界面窗口小部件。该演示是使用 MPLAB Harmony 图形设计器构造的。
primitive	演示使用 MPLAB Harmony 图形设计器构造的线、矩形、圆和形状等基元类型，以及处于多个屏幕上的图像。用户通过循环迭代来观察每个屏幕。
s1d13517	演示滑屏播放、速度测试、梯度和 alpha 混合。
ssd1926	利用 Solomon Systech SSD1926 图形控制器来演示 JPEG 图像的解码和显示。
wvga_lcc	演示如何使用低成本无控制器 （Low-Cost Controllerless， LCC）功能与 WVGA 显示屏。
wvga_glcd	演示如何使用图形 LCD （Graphics LCD， GLCD）控制器与 WVGA 显示屏。

演示

本主题介绍如何运行本发行版中包含的图形库演示应用程序。

external_resources

演示如何访问存储在外部存储器中的图形资源。

说明

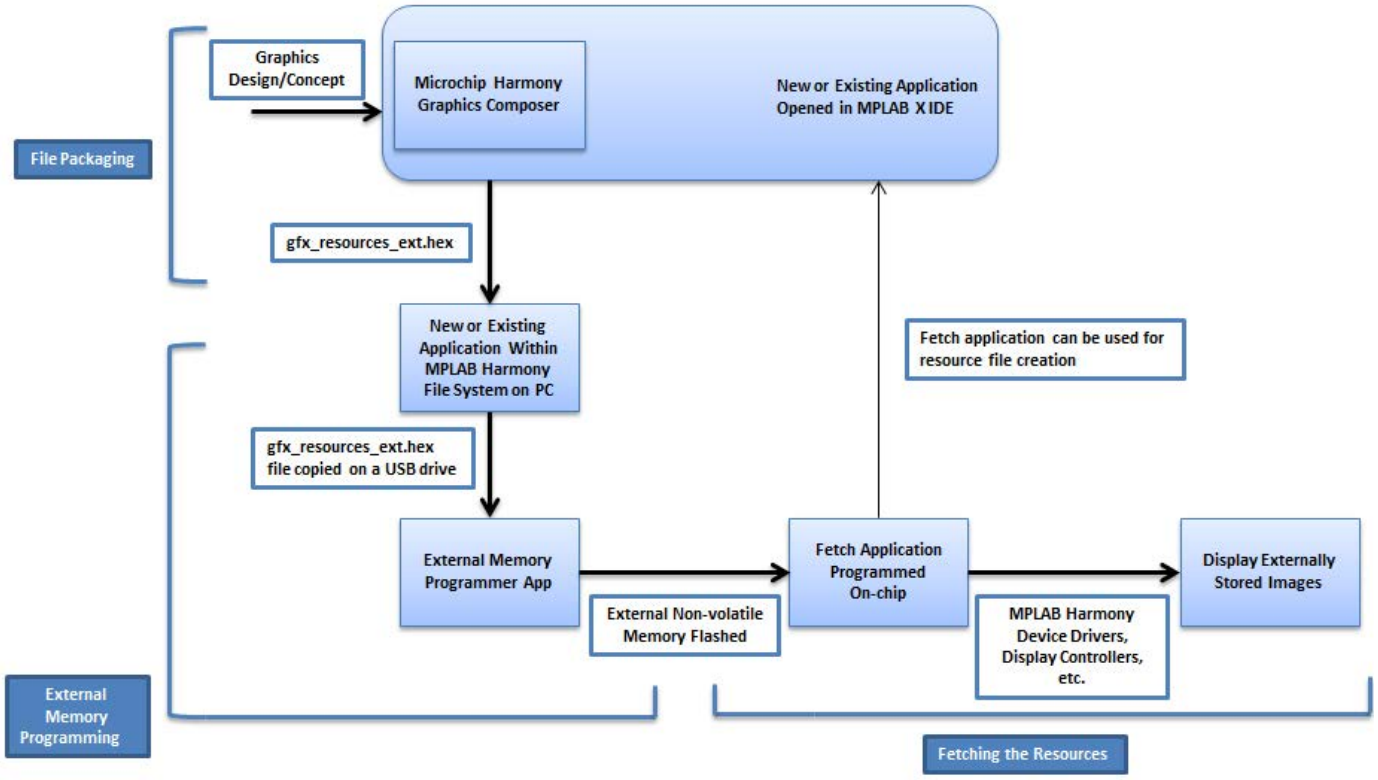
该演示说明如何访问存储在外部存储器中的图形资源。

对于具有大量图形资源要求（例如，存储大型位图文件，或存储需要多个图像或者大型或多个字体包的多页菜单屏幕等）的应用程序，这种功能会很有用。在这些情况下，在片上存储这些图形资源可能会导致存储器不足或可能成本过高。解决方案是将图形资源存储到片外存储器 （如非易失性存储器），从而将片上存储器保留用于程序存储器，并支持更复杂的功能特性。

为了演示如何访问存储在外部存储器器件上的图形资源，需要三个组件：

- 文件打包
- 自举程序应用程序
- 获取应用程序

图 1：外部资源流程图



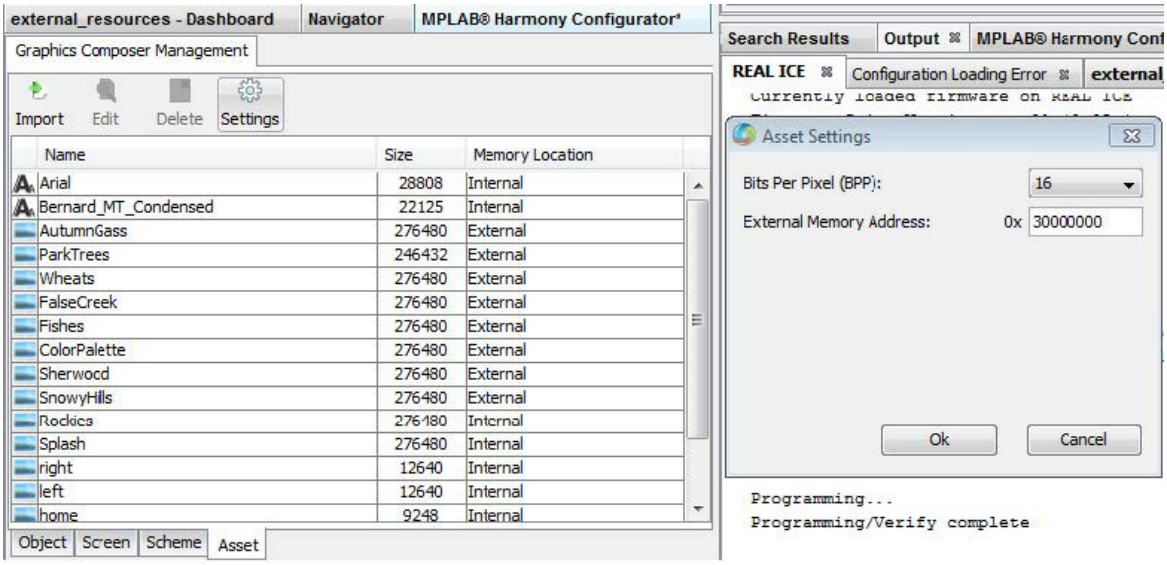
文件打包

要在外部存储的资源需要打包成一个资源文件，刷入外部存储器中。 MPLAB Harmony 图形设计器是 MPLAB Harmony 应用程序用于为应用程序配置图形设计的主要工具，它能够生成所需的资源文件，其中包含要存储在外部存储器上的所有图形图像的二进制数据。

请参见 MPLAB X IDE 联机帮助和 *Volume II: MPLAB Harmony Configurator (MHC)*> MPLAB Harmony Graphics Composer User's Guide 来了解用法信息。

图形设计器能够通过将图像和字体等不同资源导入应用程序中来管理应用程序资产。图 2 中的 Graphics Composer Management （图形设计器管理）窗口显示了为每个图形资源选择存储位置，即选择外部或内部存储器。生成图形设计器配置将会创建 `gfx_resources_ext.hex` 文件。在 MPLAB Harmony 图形设计器的未来版本中，计划包含将不同资源存储到不同外部存储器上，以及为每个独立资源指定目标外部存储器 （如 SQI 和 SPI 等）的功能。目前，选择用于存储资源的特定外部存储器通过外部存储器的起始地址来指示。如图 2 所示， Asset（资产）窗口中的 Settings 选项卡具有一个用于输入外部存储器地址的字段。在该项目中，由于我们使用的是 SQI 闪存，所以指定了起始地址 0x30000000。该信息会被包含到 `gfx_resources_ext.hex` 文件中。

图 2：Graphics Composer Management 窗口



存储器写入器应用程序

存储器写入器应用程序需要将包含外部图形资源二进制数据的`gfx_resources_ext.hex`文件刷入所需的外部非易失性存储器。存储器写入器应用程序 sqi_flash 使用处于主机模式的 USB 闪存驱动器作为`gfx_resources_ext.hex`文件的来源。存储器写入器会在处于主机模式的 USB 闪存驱动器中扫描该文件，并将它刷入 SQI 外部闪存。关于用法信息，请参见*外部存储器编程器演示* > sqi_flash。

获取应用程序

第三个组件是实际获取所存储的外部图形资源并使用或处理这些资源的应用程序。

external_resources 应用程序将演示如何获取外部存储的图形图像和如何使用 MPLAB Harmony 框架驱动程序及显示控制器等，并在屏幕上显示这些资源。该应用程序被创建为对于其图形资源具有大量存储器需求，超过可用的片上存储器空间。所使用的图形资源是大型位图图像，以滑屏播放方式显示，通过触摸输入在滑屏之间切换。 external_resources 应用程序通过使用 MPLAB Harmony 图像设计器，生成图形设计和位图图像在屏幕上的配置，并且它还会生成由外部存储器编程器应用程序用于刷入外部存储器的`gfx_resources_ext.hex`文件。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了外部资源演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开`external_resources.X`项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为`<install-dir>/apps/gfx/external_resources`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`external_resources.X`	`<install-dir>/apps/gfx/external_resources/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于`./firmware/src/system_config`中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	用于连接到 PIC32MZ EF 入门工具包和图形显示 Powertip 4.3" 480×272 电路板的多媒体扩展板 II （MEB II）的外部资源演示。 PIC32MZ EF 入门工具包中的子板具有 4 MB 的 SQI 闪存器件，用作该应用程序的外部非易失性存储器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包、 MEB II 和图形显示 Powertip 4.3" 480×272 电路板

MEB II 具有两种存储器设置选项：外部或内部；但是，该演示必须在内部模式下运行。如下配置 MEB II：

- EBIWE 和 LCD_PCLK （J9）必须闭合 （跳线位于 MEB II 扩展板背面）
- 确保所有 MHC 设置都定义为仅使用内部 SRAM

应用程序当前的内部/外部存储器设置可以通过选择 *Harmony Framework Configuration* > *Drivers* > *Graphics Controllers* > *LCC* > *Use LCC Driver* > *Memory Interface Mode* 来验证。

配置 MHC

介绍演示所需的 MPLAB Harmony 配置器设置。

说明

以下是一些从默认值修改为的 MHC 设置。

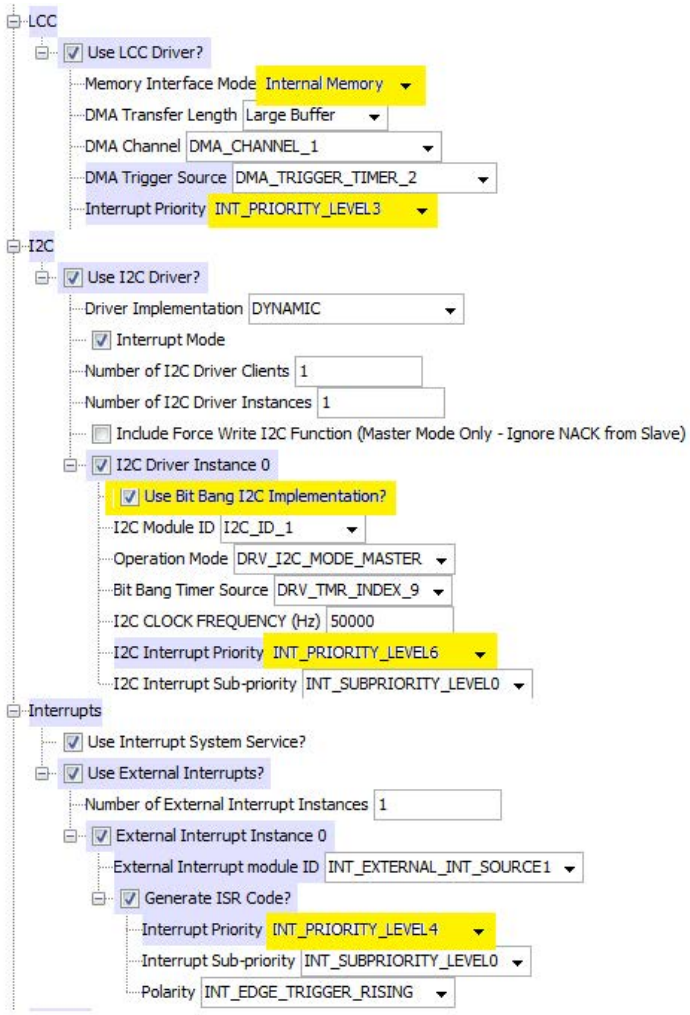
驱动程序

Graphics Controller选择部分显示使用“Use LCC driver”选择了LCC驱动程序。所选择的Memory Interface Mode是Internal Memory。这是因为我们希望将程序装入内部存储器，并且使用跳线将硬件配置为与内部存储器匹配。

除了 LCC，还选择了 I2C。对于 I2C，选择 I2C 的软件实现 （软件实现使用位响应 （bit-bang）方法），而不是硬件 I2C。它是特定于 PIC32MZ EF 入门工具包的。另外，很重要的一点是，位响应方法的 I2C 实现需要更改中断优先级。

所用模块的中断优先级必须按以下顺序从高到低安排 （详情请参见下图）：

- I2C 中断具有最高优先级 （例如，优先级为 6）
- MTCH6301 中断 （外部中断）低于 I2C，但高于 LCC DMA （例如，优先级为 5）
- LCC DMA 中断低于 MTCH6301 中断 （例如，优先级为 4）

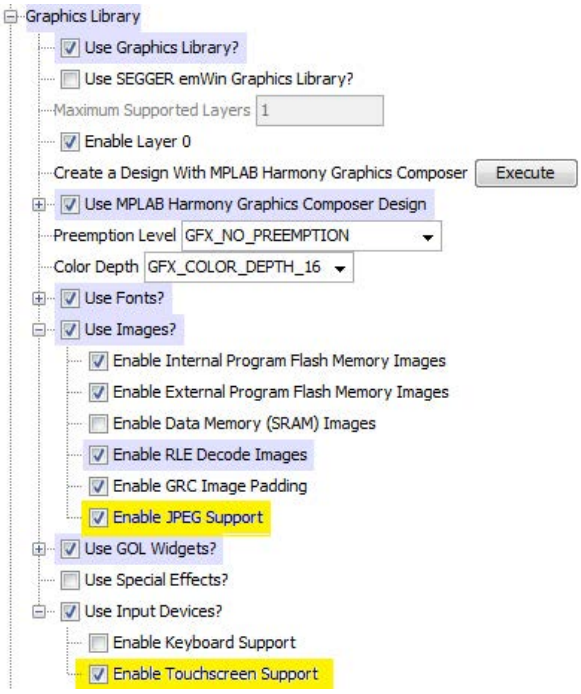


图形库

该选择会使能应用程序所需的所有图形对象库（Graphics Object Library，GOL）功能。此外，它还会使能通过 MPLAB Harmony 图形设计器执行的图形设计功能。在图形设计器中进行设计时，会自动选择不同的 GOL 组件 （如按钮、窗口小部件、图像和屏幕等），和在图形设计器设计窗口中使用的一样。

特别选择的选项为：

- Use Input devices （使用输入设备） > Enable Touch Support （使能触摸支持）
- Use Images （使用图像） > Enable JPEG Support （使能 JPEG 支持）



中断

选择了 “Use External Interrupts”（使用外部中断）选项，以接受触摸输入作为应用程序中的按钮操作和滑屏播放切换的外部中断。

BSP

为受支持的配置选择的 BSP 是 PIC32MZ EF 入门工具包与 MEB II 的组合：pic32mz_ef_sk_meb2。

时钟框图

打开时钟配置器部分，确保对于 pic32mz_sk_ef_meb2 配置所需的不同时钟设置选择以下值：

- 主振荡器频率：24 MHz
- USB PLL：48 MHz
- 系统时钟：200 MHz
- 外设总线时钟：100 MHz
- REFCLKO：参考时钟 （REFCLK）使用系统时钟 （SYSCLK）作为参考，并在 200 MHz 下生成用于 SPI 模块的主时钟

引脚表

时钟、调试和电源引脚需要根据 BSP 进行选择。应注意的特定于应用程序的引脚选择是针对触摸所作的选择：必须选择端口 / 引脚 RE8 作为 INT1。

特定于应用程序的软件设置

*SQI 外部存储器支持：*MPLAB Harmony 中未提供官方 SQI 驱动程序。因此，使用 MPLAB Harmony 中的 SQI 外设库。请注意，该项目中已经手动包含了用于支持 SQI 的源文件和头文件：

- `SourceFiles/framework/sqi_flash/drv_nvm_flash_sqi_sst26.c`
- `HeaderFiles/framework/drv_nvm_flash_sqi_sst26.h`

另外请注意，为了引用这些文件，在名为 SST26VF_B 的应用程序属性中添加了一条预处理器宏。

运行演示

介绍如何编译并运行外部资源演示。

说明

1. 在该步骤中，我们使用 external_resources 应用程序的 MPLAB Harmony 图形设计器部分来创建 `gfx_resources_ext.hex` 文件。打开计算机上的 `<install-dir>\apps\gfx\external_resourcesexternal_resources` 文件夹。要存储在外部非易

失性存储器上的外部资源在 gfx_resources_ext.hex 文件中提供。在该演示中，文件已经生成并位于 <install-dir>\apps\gfx\external_resources\firmware\src\system_config\pic32mz_ef_sk+meb2\ 文件夹中。外部存储器编程器应用程序需要该文件来刷入 SQI 存储器。如果用户选择更改图形设计或外部资源，则应使用 MPLAB X IDE 打开项目 external_resources，并应使用图形设计器更改图形。对图形进行任何更改后，需要先重新生成配置和重新生成 gfx_resources_ext.hex 文件，之后再由外部存储器编程器应用程序使用。该过程需要 MPLAB X IDE 和 MPLAB Harmony。请参见 MPLAB X IDE 联机帮助和 *Volume II: MPLAB Harmony Configurator (MHC)* > MPLAB Harmony Graphics Composer User's Guide 来了解用法信息。

2. 将 USB 闪存驱动器连接到计算机端口，并将 gfx_resources_ext.hex 文件从 apps\gfx\external_resources\firmware\src\system_config\pic32mz_ef_sk+meb2\ 复制到闪存驱动器上。
3. 接下来，将 USB 闪存驱动器连接到 PIC32MZ EF 入门工具包子板上的主机连接器。
4. 使用 pic32mz_ef_sk_meb2_gfx_ext_res 配置，通过编程器 /sqi_flash 应用程序对电路板编程。该应用程序会将 USB 闪存驱动器上的 gfx_resources_ext.hex 文件刷入外部 SQI 存储器。关于用法信息，请参见*外部存储器编程器演示* > sqi_flash。
5. external_resources 应用程序是实际获取并显示存储在外部 SQI 存储器中的外部资源的第三个组件。将 external_resources 应用程序烧写到器件中。该应用程序通过在内部和外部存储器之间分布图像来演示如何一起使用内部和外部资源。第一个菜单屏幕具有两个按钮，一个使用外部资源，一个使用内部资源。选择一个按钮就会启动滑屏播放，它可以接受触摸输入且根据该输入在滑屏之间切换，并在屏幕上显示预存的外部图像或内部图像。内部图像包含位图以及 JPEG 图像，用于演示 JPEG 支持和内部资源的运行时 JPEG 解码。该菜单还具有简要介绍演示的第三个按钮。
6. 请注意，将外部资源刷入 SQI 存储器后，不需要在每次使用 external_resources 应用程序时都重新刷入资源。但是，如果用户更改了 external_resources 演示中的图形设计，则需要重新生成配置，并需要使用步骤 2 将新的 gfx_resources_ext.hex 文件重新刷入外部存储器。这是为了确保刷入外部存储器的外部图形资源与采用资源的新图形设计器设计同步。不这样做会导致屏幕上不能正确显示外部存储的图形图像。

注： 目前，已对于内部存储使能了 JPEG 解码支持。在演示期间，可以观察到从外部的片外存储器中获取图像产生的延时导致在屏幕存储器上渲染图像时的显示填充速度很慢。在屏幕上显示 JPEG 图像时，由于 JPEG 图像运行时解码引起的延时，所以也会看到类似的延时。

lcc

利用软件图形控制器演示图形库的高级功能。

说明

该演示提供通过软件图形控制器显示 alpha 混合和梯度颜色的功能。该演示会渲染 4 个文件夹，用于说明所有 PIC32 器件的 alpha 混合、梯度、画中画 （Picture-in-Picture， PIP）和性能。

对于 PIC32MX 器件，该演示在 PICtail Plus 低成本无控制器 （LCC）子板上运行，对于 PIC32MZ 器件，它在多媒体扩展板 II （MEB II）上运行。

该演示需要使用单片机外部的存储器。存储器在电路板上提供，但必须使用硬件跳线进行配置。

重要提示！ 要设置外部存储器，必须在电路板上设置跳线。未配置该跳线设置会使显示屏无法工作，虽然软件可能仍会运行。关于相应跳线设置的详细信息，请参见配置硬件。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了低成本无控制器 （LCC）演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 lcc.X 项目，然后选择所需的配置。以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/gfx/lcc。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
lcc.X	<install-dir>/apps/gfx/lcc/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_lcc_pictail_wqvga	pic32mx_usb_sk2+lcc_pictail+wqvga	用于连接到 PIC32 USB 入门工具包 II 的低成本无控制器（LCC）图形PICtail Plus子板和图形显示Powertip 4.3" 480×272 电路板的 LCC 演示。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	用于连接到 PIC32MZ 嵌入式连接（EC）入门工具包的多媒体扩展板 II （MEB II）的 LCC 演示。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	用于连接到 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包的多媒体扩展板II（MEB II）的LCC演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和低成本无控制器 （LCC）图形 PICtail Plus 子板

对子板进行以下更改：

- 将 J4 设置到 2-3，供外部 SRAM 使用。为了使能显示操作，必须设置该跳线。确保所有 MHC 设置都定义为仅使用外部 SRAM。
- 将跳线 J6-J19 设置到 （2-3），以使用 16 位 （565）颜色模式

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

MEB II 具有两种存储器设置选项：外部或内部；但是，该演示必须在外部模式下运行；因此， EBIOE 和 LCD_PCLK （J9）必须闭合。该跳线位于 MEB II 扩展板的背面。确保所有 MHC 设置都定义为仅使用外部 SRAM

应用程序当前的内部 / 外部存储器设置可以通过选择 *Harmony Framework Configuration > Drivers > Graphics Controllers > LCC > Use LCC Driver > Memory Interface Mode* 来验证。

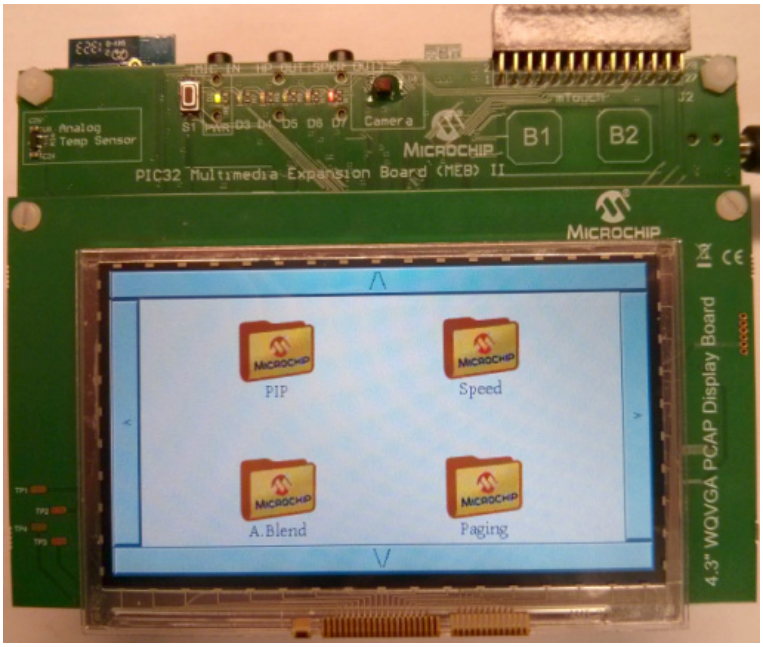
运行演示

介绍如何编译并运行 LCC 演示。

说明

该演示说明图形库如何与低成本无控制器 （LCC）软件显示控制器接口。它说明了如何通过图形库实现 alpha 混合、梯度、平移和双缓冲功能。

1. 将演示项目装入 MPLAB X IDE。
2. 编译、下载演示项目并在目标板上运行。
3. 使用图形按钮在屏幕之间导航。



演示输出

该演示会渲染 4 个文件夹，用于说明 alpha 混合、梯度、画中画 （PIP）和性能。触摸这些文件夹之一，以激活演示。触摸箭头按钮时，会在箭头方向上移动屏幕

- **PIP**：演示如何移动采用不同图像和应用了梯度的锥形表示的光标。
- **Speed （速度）**：在随机位置绘制随机大小的随机条形，从而演示绘制条形的速度
- **Alpha blend （Alpha 混合）**：演示应用于三个按钮的 alpha 混合功能，该功能会将它们与背景进行混合。背景是 MPLAB Harmony 的 “Integrate” 和 “Harmonize” 徽标的图像
- **Paging （分页）**：演示如何使用多个表面来实现背景颜色变化效果。在该演示中，背景颜色会随天气而变化。该演示还会绘制一个云的图像，其中包含显示日期、时间和温度的文本。

故障排除提示！ 在执行应用程序时，如果显示是静态的，并且是全亮或全黑，则可能说明存储器设置不正确。更多信息请参见配置硬件 。

object

图形库的标准用户界面窗口小部件的示例。

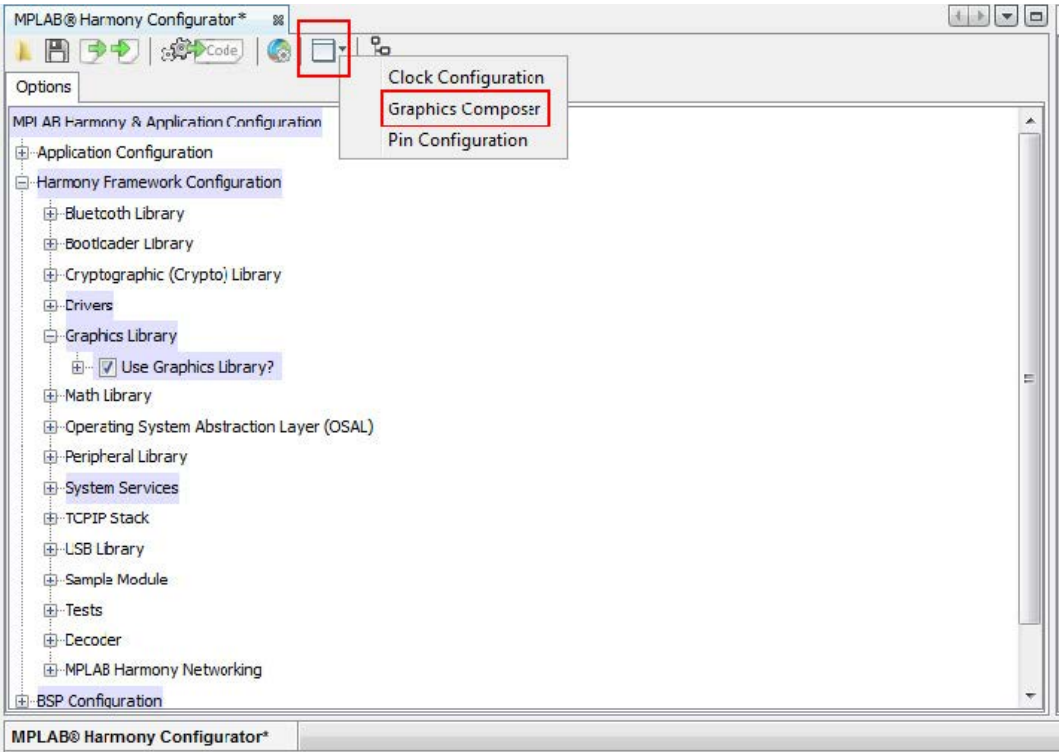
说明

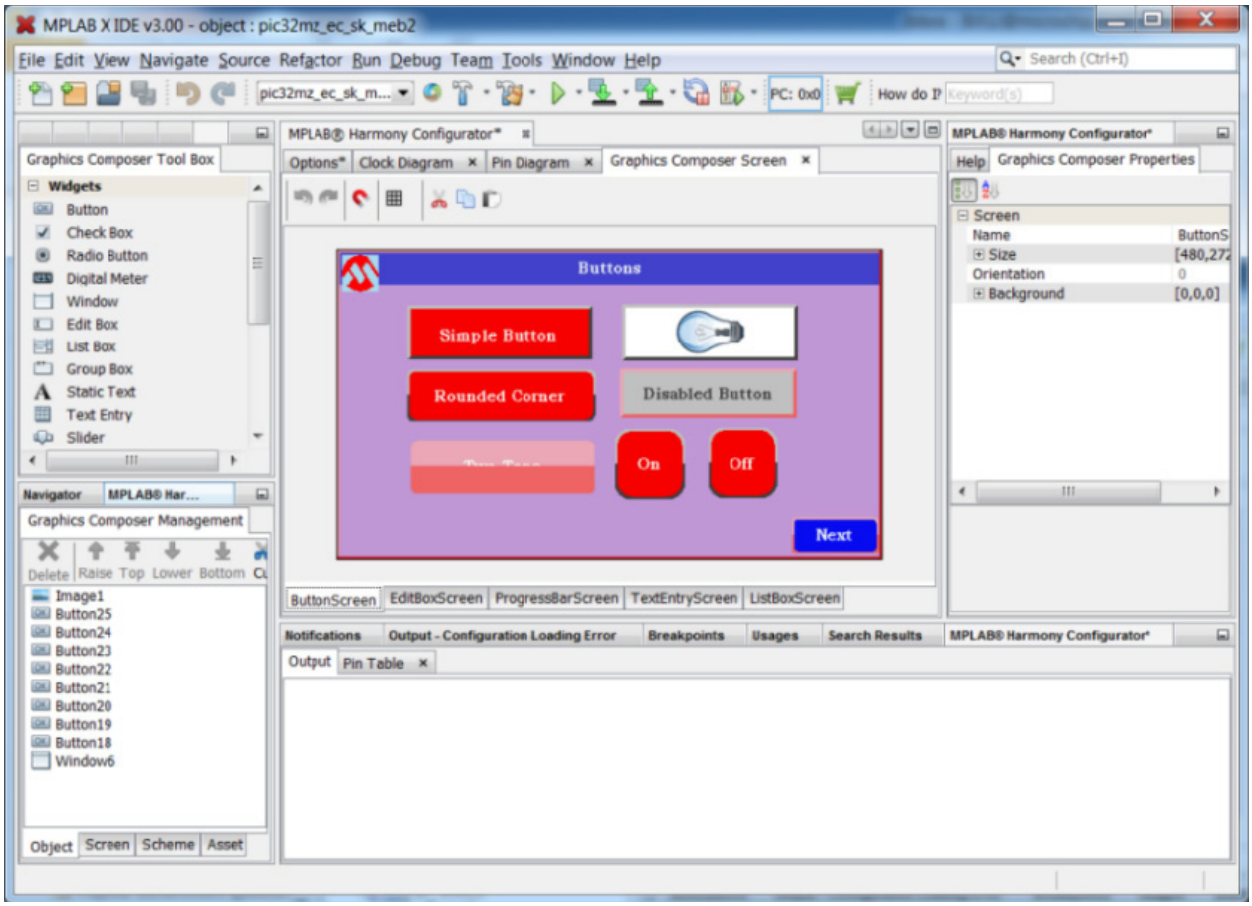
对象演示支持使用 MPLAB Harmony 图形设计器通过图形库的图形对象层 （GOL）中存在的标准类型的对象 / 窗口小部件来进行屏幕设计。该演示由几个示例页面组成。

重要提示！ 该演示必须在内部存储器模式下运行。关于设置可用存储器的更多详细信息，请参见配置硬件。

查看屏幕设计

通过在 MPLAB Harmony 配置器工具栏中使用 Launch Utility （启动实用程序）图标选择 **Graphics Composer** （图形设计器），可以从 MPLAB Harmony 配置器中激活 MPLAB Harmony 图形设计器，从而可以查看屏幕设计。更多信息请参见 MPLAB Harmony Graphics Composer User's Guide。





编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了图形对象库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `object.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/gfx/object`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`object.X`	`<install-dir>/apps/gfx/object/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_s1d_pictail_vga	pic32mx_usb_sk2+s1d_pictail+vga	用于连接到 PIC32 USB 入门工具包 II 的图形 LCD 控制器 PICtail Plus S1D13517 子板与图形显示 Truly 5.7" 640x480 电路板的对象演示。
pic32mx_usb_sk2_s1d_pictail_wvga	pic32mx_usb_sk2+s1d_pictail+wvga	用于连接到 PIC32 USB 入门工具包 II 的图形 LCD 控制器 PICtail Plus S1D13517 子板与图形显示 Truly 7" 800x480 电路板的对象演示。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	用于连接到 PIC32MZ 嵌入式连接（EC）入门工具包的多媒体扩展板 II （MEB II）的对象演示。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	用于连接到 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包的多媒体扩展板 II（MEB II）的对象演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

图形 LCD 控制器 PICtail Plus Epson S1D13517 子板

不需要硬件相关配置或跳线设置更改。

MEB II

MEB II 具有两种存储器设置选项：外部或内部；但是，该演示必须在内部模式下运行。如下配置 MEB II：

- EBIWE 和 LCD_PCLK （J9）必须闭合 （跳线位于 MEB II 扩展板背面）
- 确保所有 MHC 设置都定义为仅使用内部 SRAM

应用程序当前的内部 / 外部存储器设置可以通过选择 *Harmony Framework Configuration* > *Drivers* > *Graphics Controllers* > *LCC* > *Use LCC Driver* > *Memory Interface Mode* 来验证。

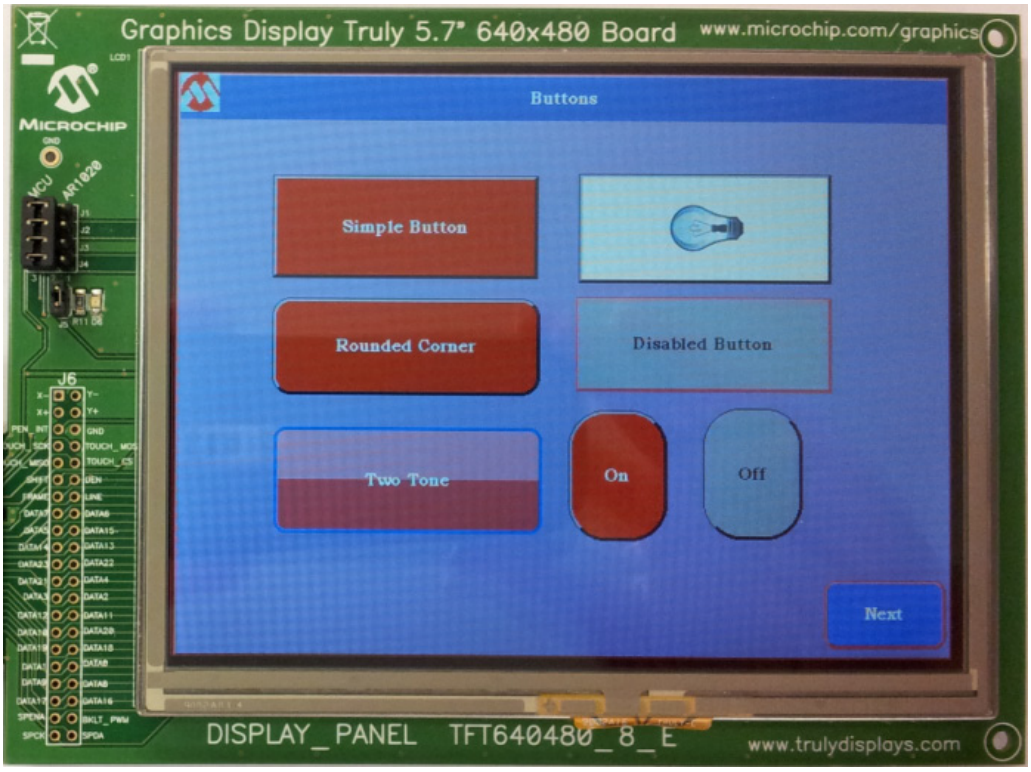
运行演示

介绍如何编译并运行对象演示。

说明

该演示使用 MPLAB Harmony 图形设计器，通过图形库的对象层 （GOL）的预制图形对象 （窗口小部件）功能来说明如何进行屏幕设计。按钮、复选框、编辑框、分组框、进度栏、单选按钮和文本输入等对象在多个显示屏幕上渲染。

1. 将演示项目装入 MPLAB X IDE。
2. 编译、下载演示项目并在目标板上运行。
3. 初始屏幕将类似于下图：



4. 通过触摸屏幕上的各个窗口小部件来体验该演示。
5. 使用 Next （下一页）和 Previous （上一页）按钮来完成演示过程。

演示输出

该演示会将每个对象演示渲染到一个窗口中。为了从一个窗口导航到另一个，在每个渲染窗口的左侧和右侧都提供了箭头按钮。可用的演示如下：

- **按钮：**渲染一个标题为 “Buttons”（按钮）的窗口，并绘制多个矩形、锥形和圆形按钮。每个按钮演示 MPLAB Harmony 图形库支持的不同状态。
- **复选框：**演示在使用复选框选择对齐方式后按钮文本对齐方式的变化。此外，还会演示复选框窗口小部件支持的不同状态。
- **单选按钮：**演示单选按钮功能和该功能支持的状态
- **分组框：**演示 MPLAB Harmony 图形库的分组框功能。分组框下的文本对齐方式根据分组框的状态选择进行修改。此外，还会演示窗口标题字符串的变化。
- **进度栏：**通过绘制一个其值在运行时更新的进度栏，演示 MPLAB Harmony 图形库的进度栏功能，并且还会演示平台更新进度栏值的速度能多快
- **列表框：**通过在列表框中选择条目来演示 MPLAB Harmony 图形库的列表框功能。条目可以通过选择中心对齐复选框对齐到中心。
- **数字仪表：**演示 MPLAB Harmony 图形库的数字仪表窗口小部件。显示的值通过不断按下 Accelerate/Deaccelerate （加速 / 减速）按钮产生变化。
- **仪表：**演示 MPLAB Harmony 图形库的仪表窗口小部件。它显示为整圆和四分之一圆表盘，并演示仪表的可配置显示属性。显示的值通过不断按下 Accelerate/Deaccelerate 按钮产生变化。

故障排除提示！ 在执行应用程序时，如果显示是静态的，并且是全亮或全黑，则可能说明存储器设置不正确。更多信息请参见配置硬件 。

primitive

一个使用 MPLAB Harmony 图形设计器来演示图形库的基元绘制功能的示例。

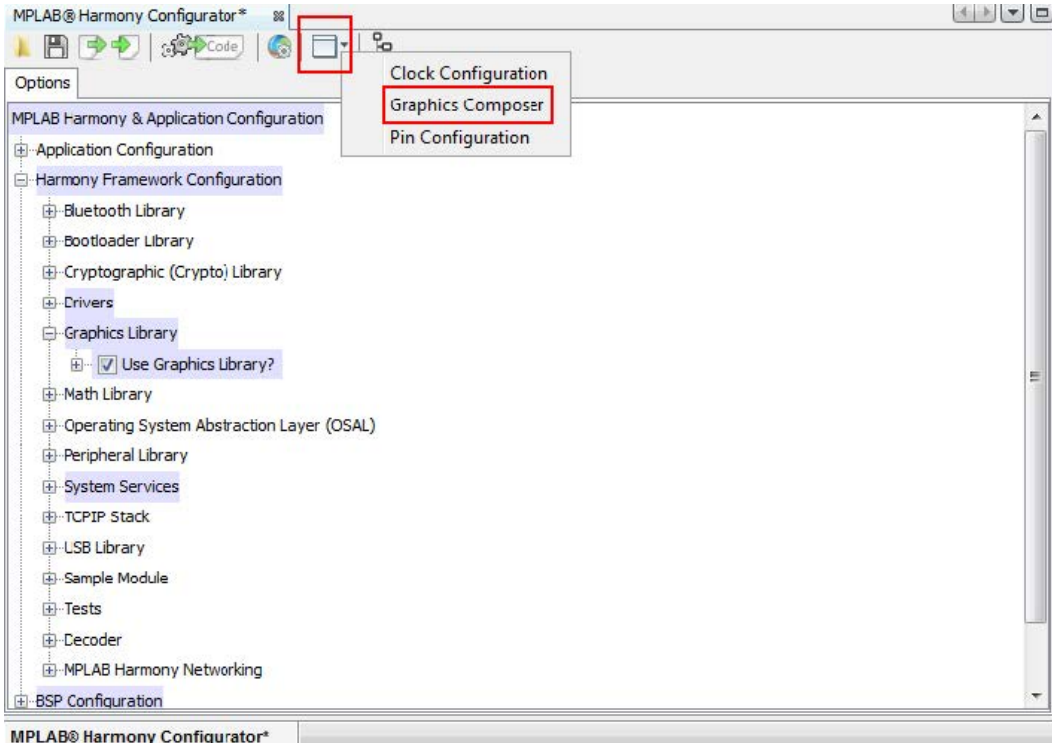
说明

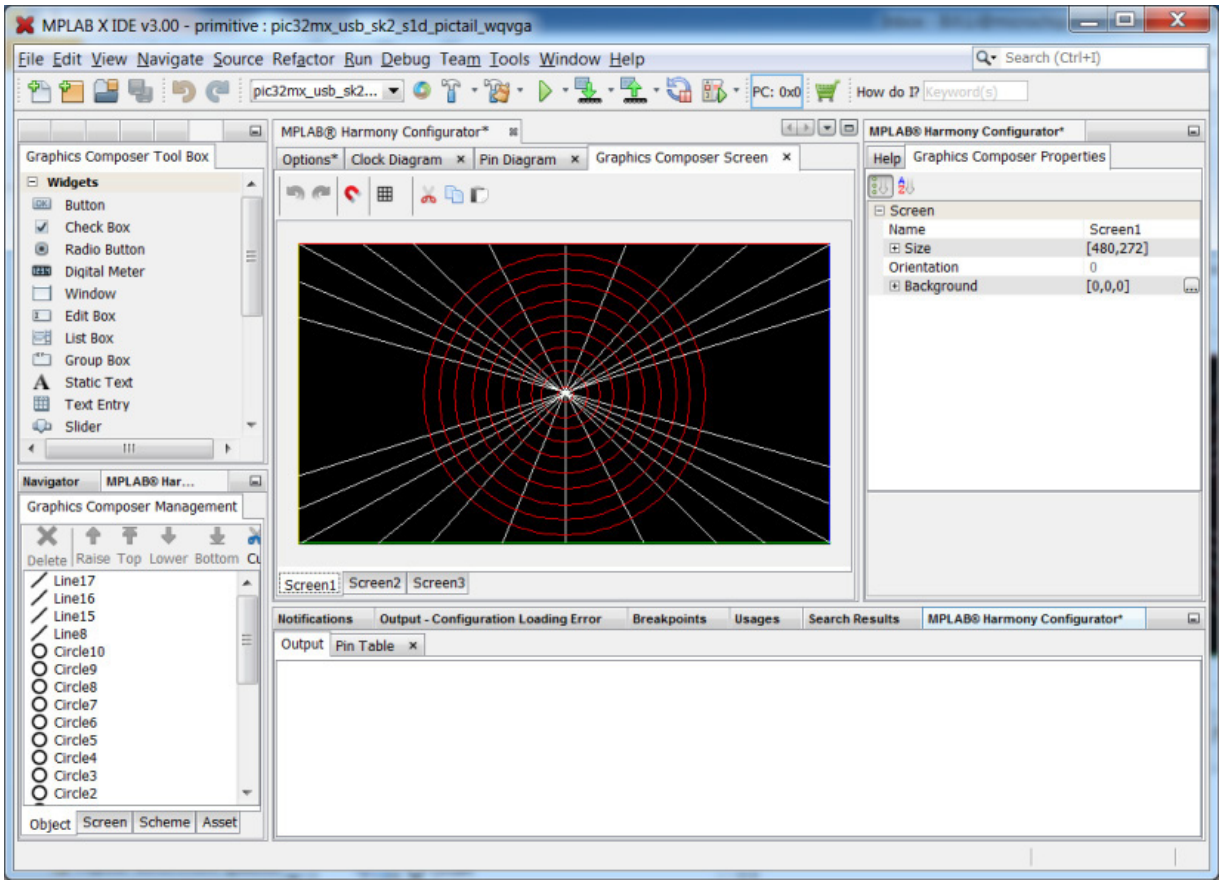
基元示例演示几何、图像和字体的基元渲染。该演示会显示各种圆、线、矩形、图像深度、带锯齿和无锯齿字体。

该演示可以在内部存储器或外部存储器模式下运行。外部存储器在 MEB II 扩展板上提供。关于设置可用存储器的更多详细信息，请参见配置硬件。

查看屏幕设计

通过在 *Graphics Library（图形库）> Create a Design With MPLAB Harmony Graphics Composer（使用 MPLAB Harmony 图形设计器创建设计）*中单击 **Execute**（执行），可以从 MPLAB Harmony 配置器中激活 MPLAB Harmony 图形设计器，从而可以查看屏幕设计。





编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了图形基元库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `primitive.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/gfx/primitive`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`primitive.X`	`<install-dir>/apps/gfx/primitive/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
bt_audio_dk	bt_audio_dk	用于 PIC32 蓝牙音频开发工具包的基元演示。

pic32mx_125_sk_lcc_pictail_qvga	pic32mx_125_sk+lcc_pictail+qvga	用于连接到 PIC32MX1/2/5 入门工具包的低成本无控制器（LCC）图形 PICtail Plus 子板和图形显示 Truly 3.2" 320×240 电路板的基元演示。
pic32mx_usb_sk2_lcc_pictail_wqvga	pic32mx_usb_sk2+lcc_pictail+wqvga	用于连接到 PIC32 USB 入门工具包 II 的低成本无控制器（LCC）图形 PICtail Plus 子板和图形显示 Powertip 4.3" 480×272 电路板的基元演示。
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	用于连接到 PIC32 USB 入门工具包 II 的多媒体扩展板（MEB）的基元演示。
pic32mx_usb_sk2_s1d_pictail_wqvga	pic32mx_usb_sk2+s1d_pictail+wqvga	用于连接到 PIC32 USB 入门工具包 II 的图形控制器 PICtail Plus Epson S1D13517 子板和图形显示 Powertip 4.3" 480×272 电路板的基元演示。
pic32mx_usb_sk2_ssd_pictail_qvga	pic32mx_usb_sk2+ssd_pictail+qvga	用于连接到 PIC32 USB 入门工具包 II 的图形 LCD 控制器PICtail Plus SSD1926子板和图形显示Truly 3.2" 320x240 电路板的基元演示。
pic32mz_ec_sk_meb2	pic32mz_ec_sk_meb2	用于连接到 PIC32MZ 嵌入式连接（EC）入门工具包的多媒体扩展板 II （MEB II）的基元演示。
pic32mz_ec_sk_s1d_pictail_wqvga	pic32mz_ec_sk+s1d_pictail+wqvga	用于连接到 PIC32MZ 嵌入式连接（EC）入门工具包的图形控制器 PICtail Plus Epson S1D13517 子板和图形显示Powertip 4.3" 480×272 电路板的基元演示。
pic32mz_ef_sk_meb2	pic32mz_ef_sk_meb2	用于连接到 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包的多媒体扩展板 II（MEB II）的基元演示。
pic32mz_ef_sk_s1d_pictail_wqvga	pic32mz_ef_sk+s1d_pictail+wqvga	用于连接到 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包的图形控制器 PICtail Plus Epson S1D13517 子板和图形显示 Powertip 4.3" 480×272 电路板的基元演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

MEB II 具有两种存储器选项：外部或内部：

- 在外部模式下运行演示时， EBIOE 和 LCD_PCLK （J9）必须闭合。该跳线 （J9）位于 MEB II 扩展板的背面。
- 在内部模式下运行演示时， EBIWE 和 LCD_PCLK （J9）必须闭合。该跳线 （J9）位于 MEB II 扩展板的背面。

应用程序当前的内部 / 外部存储器设置可以通过选择 *Harmony Framework Configuration > Drivers > Graphics Controllers > LCC > Use LCC Driver > Memory Interface Mode* 来验证。

图形 LCD 控制器 PICtail Plus SSD1926 子板

一般设置：

要将入门工具包与子板配合使用，需要 PIC32MZ 系列入门工具包适配器板。

设置 PMP 接口：

该演示只能在 8 位 PMP 模式下运行；因此，将 JP2 设置到 PARALLEL-8 位。在 MHC 中，可以通过选择 *Harmony Framework Configuration >Drivers > PMP > Transfer Size （传输大小）*来验证 PMP 模式的设置。确保该设置为 PMP_DATA_SIZE_8_BITS。

图形 LCD 控制器 PICtail Plus Epson S1D13517 子板

一般设置：

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包

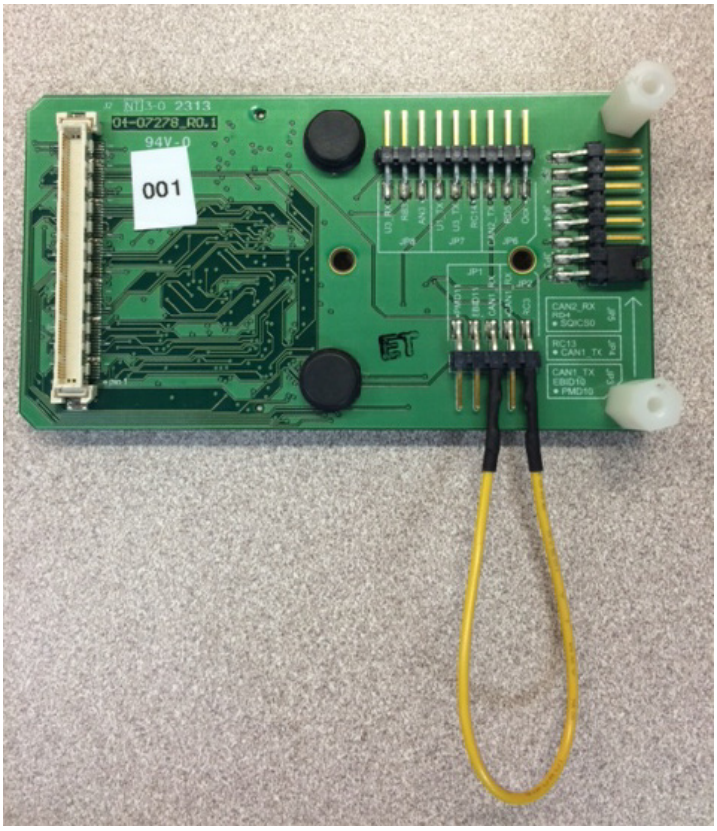
PIC32MZ 入门工具包具有 PKOB 模式 （JP2）：

- 在使用 PKOB 编程时， JP2 必须闭合
- 在使用 MPLAB REAL ICE 编程时， JP2 必须开路

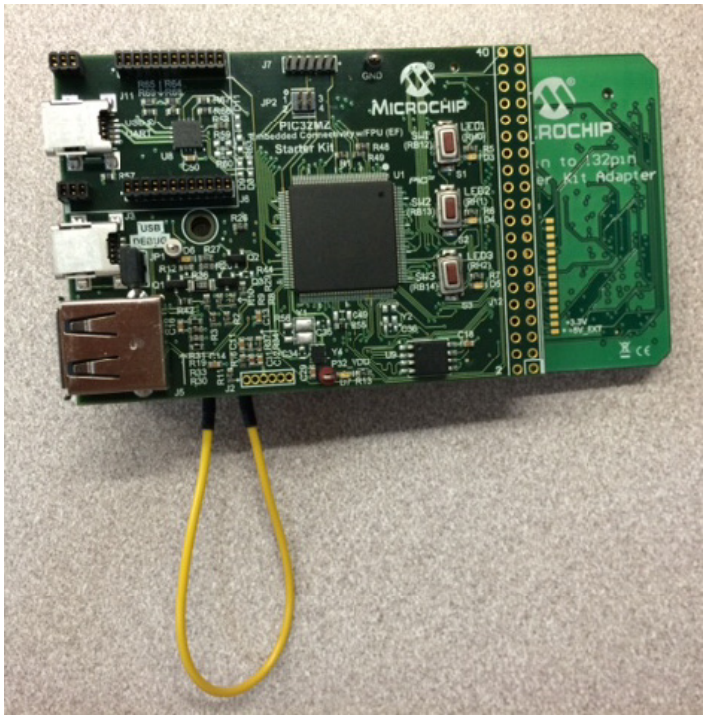
要将入门工具包与子板配合使用，需要 PIC32MZ 系列入门工具包适配器板。

- 为显示输出设置跳线 JP1 （CAN_RX）和 JP2 （RC3），如以下各图所示：

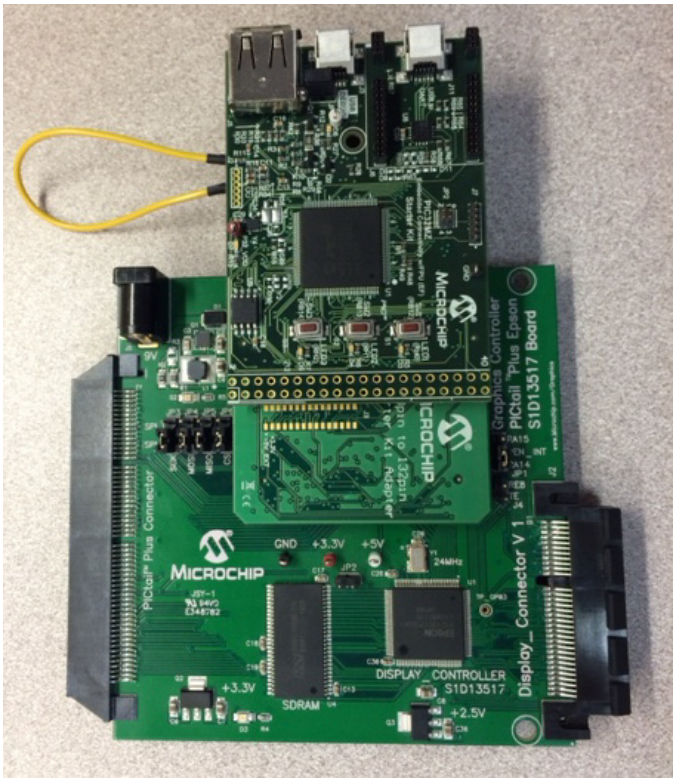
适配器板背面



连接到适配器板的 PIC32MZ EF 入门工具包



连接到图形控制器 PICtail™ Plus Epson S1D13517 子板的 PIC32MZ EF 入门工具包和适配器板



连接到图形控制器 PICtail™ Plus Epson S1D13517 子板和图形显示 Truly 7" 800x480 电路板的 PIC32MZ EF 入门 工具包和适配器板



注： 为了确保正确的演示显示操作，该电路板配置应保持平整，并避开会导致图形控制器 PICtail Plus Epson S1D13517 子板和图形显示 Truly 5.7" 和 7" 电路板之间存在的 PICtail 显示连接器发生弯曲的障碍物。

设置 PMP 接口：

该演示只能在 8 位 PMP 模式下运行；因此，将 JP2 开路。在 MHC 中，可以通过选择 *Harmony Framework Configuration >Drivers > PMP > Transfer Size* 来验证 PMP 模式的设置。确保该设置为 PMP_DATA_SIZE_8_BITS。

- 在 8 位 PMP 接口模式下运行演示时，将 JP2 开路
- 在 16 位 PMP 接口模式下运行演示时，将 JP2 闭合

低成本无控制器 （LCC）图形 PICtail Plus 子板

- 将 J4 设置到 2-3，以使用外部 SRAM。确保 MHC 设置被设为仅使用外部 SRAM。
- 将跳线 J6-J19 设置到 （2-3），以使用 16 位 （565）颜色模式

应用程序当前的内部 / 外部存储器设置可以通过选择 *Harmony Framework Configuration > Drivers > Graphics Controllers > LCC > Use LCC Driver > Memory Interface Mode* 来验证。

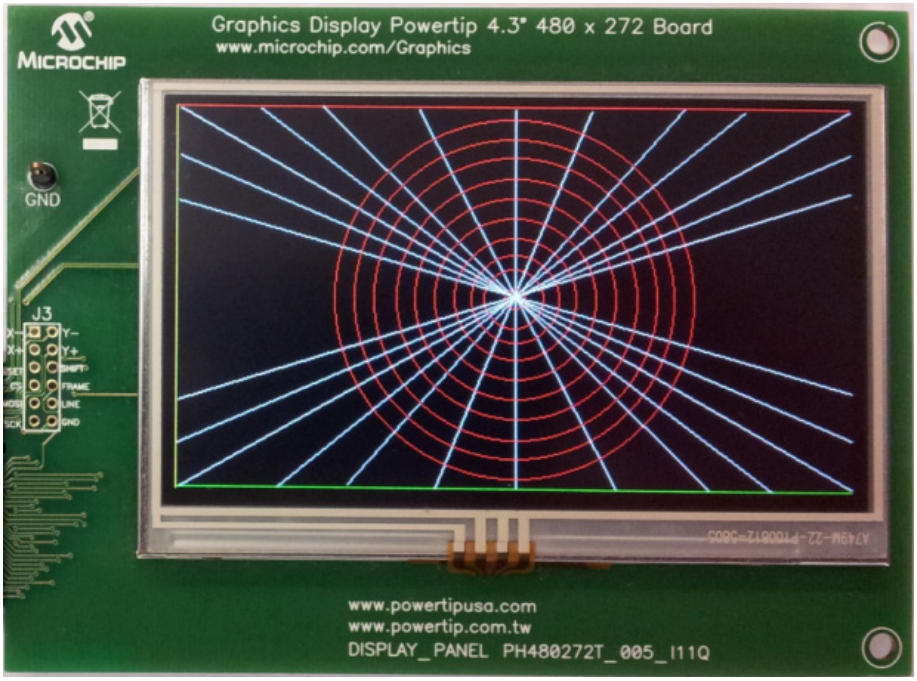
运行演示

介绍如何编译并运行基元演示。

说明

该演示通过渲染二维几何、字体和图像来说明图形库基元层的基本基元功能。

1. 将演示项目装入 MPLAB X IDE。
2. 编译、下载演示项目并在目标板上运行。
3. 第一个屏幕将类似于下图：



演示输出

该演示的输出如下：

- 绘制矩形
- 绘制颜色填充的矩形
- 绘制一个 16 BPP 图像

故障排除提示！ 在执行应用程序时，如果显示是静态的，并且是全亮或全黑，则可能说明存储器设置不正确。更多信息请参见配置硬件 。

s1d13517

图形控制器 PICtail™ Plus Epson S1D13517 电路板演示。

说明

该演示使用位于图形控制器 PICtail™ Plus Epson S1D13517 电路板上的 Epson （S1D13517）图形控制器来说明梯度、滚动、背景绘制、速度测试、 alpha 混合和应用功能。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 S1D13517 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `s1d13517.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/gfx/s1d13517`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`s1d13517.X`	`<install-dir>/apps/gfx/s1d13517/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_s1d_pictail_vga	pic32mx_usb_sk2+s1d_pictail+vga	用于连接到 PIC32 USB 入门工具包 II 的图形控制器 PICtail Plus Epson S1D13517 子板和图形显示 Truly 5.7" 640×480 电路板的 S1D13517 演示。
pic32mx_usb_sk2_s1d_pictail_wvga	pic32mx_usb_sk2+s1d_pictail+wvga	用于连接到 PIC32 USB 入门工具包 II 的图形控制器 PICtail Plus Epson S1D13517子板和图形显示Truly 7" 800×400 电路板的 S1D13517 演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

图形 LCD 控制器 PICtail Plus Epson S1D13517 子板

一般设置：

- 通过设置 4 个 SPI1 或 SPI2 跳线来设置所使用的 SPI 通道
- 将 JP1 保留为开路

设置 PMP 接口：

- 在 8 位 PMP 接口模式下运行演示时，将 JP2 开路
- 在 16 位 PMP 接口模式下运行演示时，将 JP2 闭合

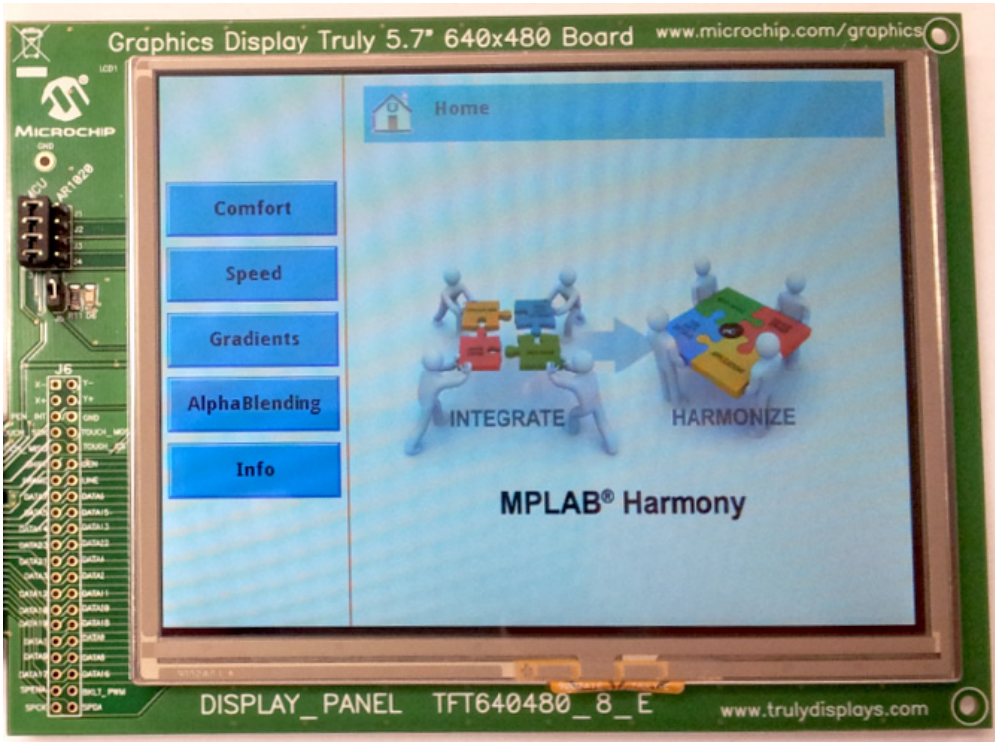
运行演示

介绍如何编译并运行 S1D13517 演示。

说明

该演示说明图形库如何与 Solomon Systech S1D13517 外部显示控制器接口。它说明了如何通过图形库实现 alpha 混合、梯度、平移和双缓冲功能。

1. 将演示项目装入 MPLAB X IDE。
2. 编译、下载演示项目并在目标板上运行。
3. 使用触摸按钮来导航到各个屏幕。



演示输出

使用 Epson （S1D13517）图形控制器来演示梯度、滚动、背景绘制、速度测试、 alpha 混合和应用功能。显示屏左侧的触摸演示按钮用于运行每个演示。

- **Comfort （舒适）：**演示空调控制菜单
- **Speed （速度）：**演示绘制随机形状和大小的填充矩形的速度
- **Gradients （梯度）：**使用起始 （红色）和结束 （黑色）颜色来演示梯度功能的不同模式
- **AlphaBlending （Alpha 混合）：**演示前景中的 alpha 值在运行时从最大值变为最小值时，背景如何与着色的前景进行混合
- **Info （信息）：**显示关于每个演示的信息

ssd1926

说明如何使用位于 PICtail™ Plus SSD1926 子板上的 Solomon Systech （SSD1926）图形控制器来显示 JPEG 图像。

说明

SSD1926 演示说明如何利用 Solomon Systech SSD1926 图形控制器的 JPEG 解码器功能对来自 SD 卡的 JPEG 图像进行解码。

该演示展示了 LCD 控制器 Solomon Systech SSD1926 独特的硬件功能。它的两个独特的功能是 SSD1926 上的硬件 JPEG 解码器和硬件 4 位 SD 卡接口。

该演示只能使用 QVGA （320x240）显示屏运行，不能使用 WQVGA （480×272）运行。原因是 SSD1926 上只有 256 KB 的 RAM 可用。如果配置为使用 16 bpp 颜色深度的 WQVGA 显示屏，则需要 255 KB 的帧缓冲区空间，没有足够的 RAM 留给 JPEG 解码。因此，该演示只能在 QVGA 设置上运行。

该演示使用 Microchip 存储器磁盘驱动器文件系统 （Memory Disk Drive File System， MDDFS）接口库与 Microchip 图形库，在安装了 SD 卡插座的图形 LCD 控制器 PICtail™ Plus SSD1926 电路板 （AC164127-5）上工作。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SSD1926 演示的可用配置。

说明

注： 默认情况下，该项目配置为在 8 位 PMP 接口模式下运行。目前不支持 16 位 PMP 模式。

要编译该项目，必须在 MPLAB X IDE 中打开 `ssd1926.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/gfx/ssd1926`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`ssd1926.X`	`<install-dir>/apps/gfx/ssd1926/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_ssd_pictail_qvga	pic32mx_usb_sk2+ssd_pictail+qvga	用于连接到 PIC32 USB 入门工具包 II 的图形 LCD 控制器 PICtail Plus SSD1926 子板和图形显示 Truly 3.2" 320x240 电路板的 SSD1926 演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

图形 LCD 控制器 PICtail Plus SSD1926 子板

一般设置：

- 通过将 JP3 设置到 RD1-FLASH CS，将 SPI 闪存片选信号设置为使用 RD1
- 将 JP1 保留为开路

设置 PMP 接口：

- 只有 8 位 PMP 接口模式支持该演示。要运行该演示，请将 JP2 设置到 PARALLEL-8 位

运行演示

介绍如何编译并运行 SSD1926 演示。

说明

该演示说明 GFX 库如何与 Epson SSD1926 外部显示控制器接口。它会显示从连接到显示控制器的 SD 卡中读取的照片帧。

1. 插入 SD 卡，并装入来自演示的 resources 文件夹的图像。
2. 将演示项目装入 MPLAB X IDE。
3. 编译、下载演示项目并在目标板上运行。
4. 观察存储在 SD 卡上的一系列照片帧。

演示输出

在 SD 卡内装入来自演示的 resources 文件夹的图像后，演示以下方面：

- Microchip 徽标
- 样例视频
- 场景 1
- 场景 2
- 场景 3
- 场景 4
- 场景 5

wvga_glcd

提供关于演示的信息。

说明

使用图形 LCD （GLCD）控制器和 MTCH6303 PCAP 触摸控制器以及 5" WVGA PCAP 显示板来演示图形库的高级功能 （见**注**）。

注： 该演示针对的是 PIC32MZ DA 早期采用者，它独立于官方 MPLAB Harmony 1.07 发行版分发。关于获取安装程序和 / 或 5" WVGA PCAP 显示板的信息，请与您当地的 Microchip 销售办事处联系。

该应用程序的目的是演示 GLCD 控制器同时渲染 4 个图层 （1 个基底图层加上 3 个具有独立 alpha 和颜色模式的图层）的功能，以及演示 MTCH6303 的基本功能。

该演示还具有内置的调试控制台功能，用于说明 MTCH6303 的单点触摸精度，以及演示如何使用控制台系统服务作为屏幕校准工具。

该演示是针对 PIC32MZ DA 早期采用者开发的， MPLAB Harmony 图形设计器和图形对象库仅支持其图形功能的一个子集。它不会使用 GPU 外设。由于触摸系统服务目前不支持 MTCH6303 驱动程序，所以应用程序会直接访问驱动程序。

该演示包含以下特性：

- 24 位真彩色 （16.7M）
- 3 个同时渲染的图层
- 每像素动态 alpha 混合
- 全局层动态 alpha 混合
- 触摸互动圆圈处于图层 1，显示在图层 0 的图像之上
- 处于图层 2 的持久性菜单通过硬件混合在图层 1 和 0 之上
- JPEG 压缩图像存储在内部闪存中
- 实时 JPEG 解码和渲染
- 采用 DDR2 SDRAM 存储器
- 非同步双缓冲渲染
- WVGA 显示屏使用 40+ Hz 刷新速率
- 集成的 MTCH6303 PCAP 触摸接口 （非手势单点触摸）

编译应用程序

本主题指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了图形 LCD （GLCD）控制器 WVGA 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wvga_glcd.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/gfx/wvga_glcd`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wvga_glcd.X`	`<install-dir>/apps/gfx/wvga_glcd/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_da_sk_meb2_wvga	pic32mz_da_sk+meb2+wvga	用于连接到 PIC32MZ 图形 （DA）入门工具包的多媒体扩展板 II （MEB II）的 GLCD 演示。

应用程序头文件

下表列出并说明了演示使用的应用程序头文件，它位于 `./firmware/src` 中。

项目配置名称	说明
`app.h`	该头文件提供应用程序的函数原型和数据类型定义。其中一些是系统需要的 （如 APP_Initialize 和 APP_Tasks 原型），一些则只在应用程序内部使用 （如 APP_STATES 定义）。为了方便，它们都在该文件中定义。

配置硬件

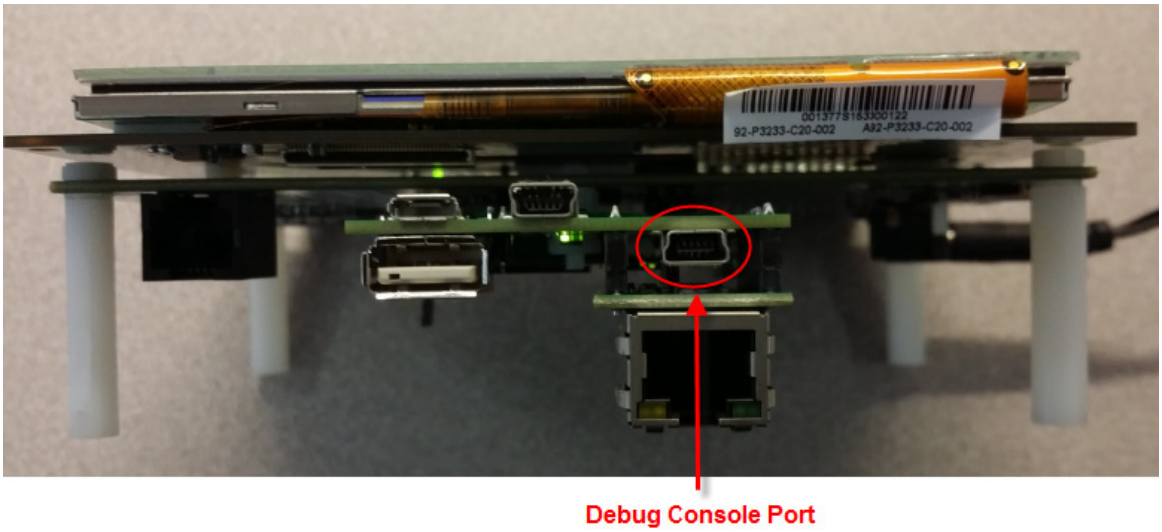
介绍如何配置支持的硬件。

说明

PIC32MZ DA 入门工具包和 MEB II

EBIOE 和 LCD_PCLK （J9）必须闭合。该跳线 （J9）位于 MEB II 扩展板的背面。

该演示还会通过 UART over USB 使能调试控制台。您需要一条 USB A 型转 USB B 型的电缆。将该电缆与端口连接，如下图所示。此外，使用以下串行端口设置来配置终端仿真器软件：8 位，无奇偶校验， 1 个停止位，无流量控制。



运行演示

介绍如何编译并运行 GLCD WVGA 演示。

说明

该演示具有 3 种模式：

- 首页模式

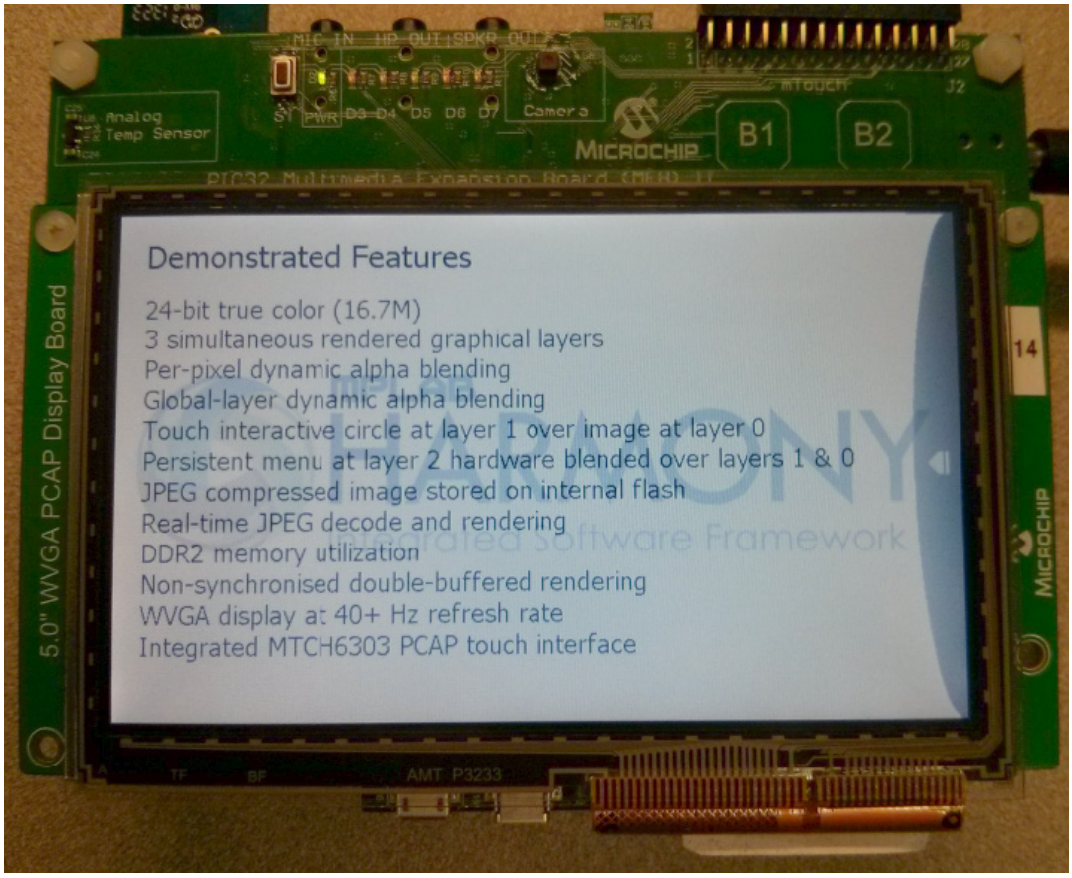
- 全屏滑屏播放模式
- 触摸响应气泡模式

侧边菜单在最上层上渲染，它通过 alpha 混合实时混合到两个图层之上。

首页模式

该模式包含一个由应用程序演示的功能的列表，并同时演示全局层 alpha 混合功能。

首页模式示例



全屏滑屏播放模式

该模式演示运行时 JPEG 解码、图形库双缓冲和 DDR2 SDRAM 上的 GLCD 渲染

全屏滑屏播放模式示例



触摸响应气泡模式

该模式演示 MTCH6303 触摸的精度，以及实时每像素 alpha 混合功能。

触摸响应气泡模式示例



应用程序函数和原型

a) 函数

	名称	说明
	APP_DrawBubble	在触摸响应气泡模式期间绘制气泡
	APP_DrawMenu	基于应用程序的当前状态在图层 2 上绘制菜单
	APP_FillMenuBackBuffer	在后台缓冲区中预高速缓存菜单图像资产
	APP_HandleHomeSlide	在首页模式期间更改透明度和图像
	APP_Initialize	MPLAB Harmony 演示应用程序初始化程序
	APP_IsCircle	检测点 (x,y) 是否属于气泡的一部分
	APP_MoveBubble	在触摸响应气泡模式下执行气泡移动
	APP_PrepareHomeMode	在应用程序中为首页模式准备缓冲区和图层
	APP_ProcessMenu	处理菜单的触摸事件并处理模式切换
	APP_ReceiveMenuTouch	接收并记录触摸菜单事件
	APP_SelectNewSlide	选择新的滑屏
	APP_Tasks	MPLAB Harmony 演示应用程序任务函数
	APP_TouchEventHandler	触摸事件回调程序。

b) 数据类型和常量

	名称	说明
	APP_DATA	存放应用程序数据
	APP_MENU_STATES	应用程序菜单状态枚举
	APP_MODES	应用程序模式枚举
	APP_STATES	应用程序状态枚举

	APP_GLCD_LAYER0_BUFFER_ADDR	定义 GLCD 图层 0 的辅助缓冲区的存储器基址
	APP_GLCD_LAYER1_ALPHA_RESOLUTION	决定 Microchip 和 Harmony 徽标渐变和显示的平滑度
	APP_GLCD_LAYER1_BUFFER_ADDR	定义 GLCD 图层 1 的辅助缓冲区的存储器基址
	APP_GLCD_LAYER2_BUFFER_ADDR	定义 GLCD 图层 2 的辅助缓冲区的存储器基址
	APP_GLCD_LAYER2_HOR_RES	GLCD 图层 2 的水平分辨率
	APP_HOME_MODE_SEMI_TRANSPARENT_ALPHA_VALUE	首页模式使用该值来使 Microchip 和 Harmony 徽标保持在半透明级别
	APP_TRANSPARENT_ALPHA_VALUE	GLCD 驱动程序接受为 100% 透明的值

说明

列出可供演示使用的函数和原型，它们在 app.h 头文件中提供。

a) 函数

APP_DrawBubble 函数

在触摸响应气泡模式期间绘制气泡

文件

wvga_glcd_app.h

C

```
void APP_DrawBubble();
```

返回值

无。

说明

每个SYS_TASK循环都会调用该函数，用于在图层1上渲染气泡。气泡的绘制位置基于MTCH6303触摸驱动程序报告的最新触摸位置。

该函数会在局部管理双缓冲方案，以改善视觉效果。该函数负责渲染整个图层 1。这意味着它会对完整 800x480 表面的每个像素 “强制”应用颜色和 alpha 值。尝试过各种优化方案，没有一种可以提供令人满意的视觉效果。

函数

void APP_DrawBubble (void)

APP_DrawMenu 函数

基于应用程序的当前状态在图层 2 上绘制菜单

文件

wvga_glcd_app.h

C

```
void APP_DrawMenu();
```

返回值

无。

说明

该函数用于在运行时基于 appData.mode 和 appData.menuMode 构造菜单。它会处理菜单切换动作的绘制。

它通过引用在后台缓冲区 （见 APP_FillMenuBackBuffer）中预先缓存的菜单资产来执行所有这些操作。

函数

void APP_DrawMenu (void)

APP_FillMenuBackBuffer 函数

在后台缓冲区中预缓存菜单图像资产

文件

wvga_glcd_app.h

C

```
void APP_FillMenuBackBuffer();
```

返回值

无。

说明

该函数用于将 JPEG 图像资产预先高速缓存到某个存储器位置。高速缓存每个图像的位置的起始地址存储在 appData 中，之后用于绘制菜单。

此外，该函数还通过使用限定范围内的掩码 （RGB 000000 至 D0D0D0）应用全透明 alpha 值，产生菜单平滑的弧形前部。

最后，当产生的菜单在渲染时为半透明时，会对图像资产的每个像素应用半透明 alpha 值。

函数

void APP_FillMenuBackBuffer (void)

APP_HandleHomeSlide 函数

在首页模式期间更改透明度和图像

文件

wvga_glcd_app.h

C

```
void APP_HandleHomeSlide();
```

返回值

无。

说明

该函数会使图层 1 的全局层 alpha 值产生波动。它会通过指示 GLCD 渲染预先高速缓存了 JPEG 图像的独立缓冲区，在 Harmony 和 Microchip 徽标之间切换图层 1 图像。

函数

void APP_HandleHomeSlide (void)

APP_Initialize 函数

MPLAB Harmony 演示应用程序初始化程序

文件

wvga_lcc_app.h

C

```
void APP_Initialize();
```

返回值

无。

说明

该程序用于初始化 MPLAB Harmony 演示应用程序。该函数会打开必需的驱动程序，初始化定时器，并向 USART 驱动程序注册应用程序回调。

备注

该程序必须从 SYS_Initialize 函数中调用。

前提条件

在调用该程序之前，应先调用所有其他系统初始化程序 （在 “SYS_Initialize”中）。

示例

```
APP_Initialize();
```

函数

void APP_Initialize (void)

APP_IsCircle 函数

检测点 (x,y) 是否属于气泡的一部分

文件

wvga_glcd_app.h

C

```
bool APP_IsCircle(uint32_t centerX, uint32_t centerY, uint32_t x, uint32_t y);
```

返回值

如果在圆中，则为 true

说明

使用半径检查方案来确定点是否属于圆的一部分

参数

参数	说明
uint32_t centerX, uint32_t centerY	圆的中心点的笛卡尔坐标
uint32_t x, uint32_t y	要检查的点的笛卡尔坐标

函数

bool APP_IsCircle(uint32_t centerX, uint32_t centerY, uint32_t x, uint32_t y)

APP_MoveBubble 函数

在触摸响应气泡模式下执行气泡移动

文件

wvga_glcd_app.h

C

```
void APP_MoveBubble();
```

返回值

无。

说明

每个 APP_TASK 循环都会调用该函数，用于基于最新的触摸位置重新调整气泡的当前位置。该函数会在最新触摸位置与当前气泡位置之间执行线性插值，而不是正好在触摸点渲染气泡。这会产生令人愉悦的运动 / 跟随效果。

函数

void APP_MoveBubble (void)

APP_PrepareHomeMode 函数

在应用程序中为首页模式准备缓冲区和图层

文件

wvga_glcd_app.h

C

```
void APP_PrepareHomeMode();
```

返回值

无。

说明

该函数用于在独立的缓冲区中预先高速缓存两个要在图层 1 上渲染的 JPEG 图像 （Harmony 和 Microchip 徽标）。它会将图层 1 的混合功能设置为由全局层 alpha 值控制。它会将图层 1 的全局层全局值设置为 0 （完全不透明）。

然后，它会指示 GLCD 在图层 0 上渲染功能列表，在图层 1 上渲染 Harmony 徽标。

函数

void APP_PrepareHomeMode (void)

APP_ProcessMenu 函数

处理菜单的触摸事件并处理模式切换

文件

wvga_glcd_app.h

C

```
void APP_ProcessMenu();
```

返回值

无。

说明

每个 APP_TASK 循环都会调用该函数，用于检测是否发生了触摸菜单事件。当它检测到用户请求在模式之间切换时，它会处理所有必需的初始设置。

函数

void APP_ProcessMenu (void)

APP_ReceiveMenuTouch 函数

接收并记录触摸菜单事件

文件

wvga_glcd_app.h

C

```
void APP_ReceiveMenuTouch();
```

返回值

无。

说明

该函数用于处理传入的触摸事件并记录它，供应用程序之后在 APP_ProcessMenu 中处理

函数

void APP_ReceiveMenuTouch (void)

APP_SelectNewSlide 函数

选择新的滑屏

文件

wvga_lcc_app.h

C

```
void APP_SelectNewSlide();
```

返回值

无。

说明

该函数在滑屏模式期间调用，用于启动 GFX_HGC_ChangeScreen 调用。这会触发由 MPLAB Harmony 图形设计器生成的代码，使屏幕发生变化。为滑屏选择的 JPEG 图像在运行时采用双缓冲方案进行解码。对绘制缓冲区进行解码时，会在解码完成时与渲染缓冲区交换。如果不使用该方案，由于 JPEG 解码器的逐块解码算法，渲染图像时会产生窗帘效果。双缓冲由 GFX_Primitive 层自动处理。

函数

void APP_SelectNewSlide (bool userTiggered)

APP_Tasks 函数

MPLAB Harmony 演示应用程序任务函数

文件

wvga_lcc_app.h

C

```
void APP_Tasks();
```

返回值

无。

说明

该程序是 MPLAB Harmony 演示应用程序的任务函数。它定义应用程序的状态机和核心逻辑。

备注

该程序必须从 SYS_Tasks() 程序中调用。

前提条件

在调用该函数之前，应先调用系统和应用程序初始化函数 （“SYS_Initialize”）。

示例

```
APP_Tasks();
```

函数

void APP_Tasks (void)

APP_TouchEventHandler 函数

触摸事件回调程序。

文件

wvga_lcc_app.h

C

```
void APP_TouchEventHandler(DRV_MTCH6303_BUFFER_EVENT event, DRV_MTCH6303_BUFFER_HANDLE
bufferHandle, uintptr_t context );
```

说明

该函数是一个回调函数，由MTCH6303驱动程序在发生触摸事件时调用。触摸事件的数据会被高速缓存在appData.mostRecentTouchX和 appData.mostRecentTouchY 中。

函数

void APP_TouchEventHandler(

DRV_MTCH6303_BUFFER_EVENT event,

DRV_MTCH6303_BUFFER_HANDLE bufferHandle,

uintptr_t context);

b) 数据类型和常量

APP_DATA 结构

存放应用程序数据

文件

wvga_lcc_app.h

C

```
typedef struct {
  SYS_TMR_HANDLE sysTmrHandle;
  APP_STATES state;
  APP_MODES mode;
  DRV_HANDLE hDrvMTCH6303;
  DRV_MTCH6303_BUFFER_HANDLE hTouchDataBuff;
  DRV_MTCH6303_TOUCH_DATA touchData;
  int32_t staleTracker;
  bool touchProcessed;
  int16_t currentTouchBehavior;
  int16_t mostRecentTouchX;
  int16_t mostRecentTouchY;
  uint32_t frameCount;
  int slidePauseTime;
  bool hasTipsBeenRead;
  GFX_XCHAR secondString[2];
  APP_LANGUAGES language;
} APP_DATA;
```

成员

成员	说明
SYS_TMR_HANDLE sysTmrHandle;	定时器系统服务句柄，用于在应用程序处于滑屏播放模式时跟踪滑屏之间的延时
APP_STATES state;	应用程序的当前状态
APP_MODES mode;	应用程序的当前模式
DRV_HANDLE hDrvMTCH6303;	MTCH6303 驱动程序句柄

DRV_MTCH6303_BUFFER_HANDLE hTouchDataBuff;	MTCH6303 缓冲区句柄
DRV_MTCH6303_TOUCH_DATA touchData;	MTCH6303 触摸数据
int32_t staleTracker;	它用于跟踪当前触摸数据是有效还是已过时
bool touchProcessed;	如果接收到新的触摸信息，则为 true。它可以是有效触摸输入或无接触事件
int16_t currentTouchBehavior;	它用于跟踪当前触摸事件是 PRESS、MOVE、STILLPRESS、RELEASE 还是 INVALID （无接触）
int16_t mostRecentTouchX;	最新有效触摸位置 x 坐标
int16_t mostRecentTouchY;	最新有效触摸位置 y 坐标
uint32_t frameCount;	帧计数跟踪器，用于帮助降低触摸数据的采样速率。需要该计数是为了帮助确定触摸移动、仍然按下和释放行为。
int slidePauseTime;	根据用户通过 Settings 屏幕对它所作的设置，跟踪滑屏之间的停顿间隔时间
bool hasTipsBeenRead;	在处于滑屏播放模式时，跟踪是否需要显示滑屏播放的指示信息屏幕
GFX_XCHAR secondString[2];	它以 GFX_XCHAR 格式显示 Settings 屏幕的停顿间隔时间的值
APP_LANGUAGES language;	跟踪用户通过 Settings 屏幕设置的当前语言

说明

APP_DATA

该结构用于存放应用程序的数据。

备注

具体信息请参见行内的注释。

APP_MENU_STATES 枚举

应用程序菜单状态枚举

文件

wvga_glcd_app.h

C

```
typedef enum {
  APP_MENU_STATE_INIT,
  APP_MENU_STATE_COLLAPSED,
  APP_MENU_STATE_COLLAPSED_PRESSED,
  APP_MENU_STATE_EXPANDED,
  APP_MENU_STATE_EXPAND_TRANSITION,
  APP_MENU_STATE_COLLAPSE_TRANSITION
} APP_MENU_STATES;
```

成员

成员	说明
APP_MENU_STATE_INIT	初始状态，启动时菜单只会处于该模式
APP_MENU_STATE_COLLAPSED	菜单折叠时的状态，渲染显示未点亮的箭头按钮的资产
APP_MENU_STATE_COLLAPSED_PRESSED	菜单折叠时的状态，渲染显示点亮的箭头按钮的资产
APP_MENU_STATE_EXPANDED	菜单展开时的状态，由两个资产一同构成菜单、提示和按钮栏
APP_MENU_STATE_EXPAND_TRANSITION	从折叠到展开的过渡状态
APP_MENU_STATE_COLLAPSE_TRANSITION	从展开到折叠的过渡状态

说明

APP_MENU_MODES

该枚举定义有效的应用程序菜单状态枚举。这些状态决定滑动菜单在各种情况下的行为。

备注

具体信息请参见行内的注释。

APP_MODES 枚举

应用程序模式枚举

文件

wvga_lcc_app.h

C

```
typedef enum {
  APP_MODE_MAIN_MENU,
  APP_MODE_TOUCH_TEST,
  APP_MODE_SLIDES,
  APP_MODE_SETTINGS,
  APP_MODE_FEATURE_LIST
} APP_MODES;
```

成员

成员	说明
APP_MODE_MAIN_MENU	主菜单
APP_MODE_TOUCH_TEST	触摸测试
APP_MODE_SLIDES	滑屏播放
APP_MODE_SETTINGS	设置
APP_MODE_FEATURE_LIST	功能列表

说明

APP_MODES

该枚举定义有效的应用程序模式枚举。支持 5 种模式：主菜单、触摸测试、设置、滑屏播放和功能列表

备注

具体信息请参见行内的注释。

APP_STATES 枚举

应用程序状态枚举

文件

wvga_lcc_app.h

C

```
typedef enum {
  APP_STATE_TOUCH_INIT = 0,
  APP_STATE_TOUCH_EVENT_REGISTER,
  APP_STATE_MAIN_MENU,
  APP_STATE_SLIDE_SHOW,
  APP_STATE_TOUCH_TEST,
  APP_STATE_SETTINGS,
  APP_STATE_FEATURE_LIST
} APP_STATES;
```

成员

成员	说明
APP_STATE_TOUCH_INIT = 0	初始状态
APP_STATE_TOUCH_EVENT_REGISTER	触摸登记，该状态专门用于 MTCH6303 驱动程序。 通常情况下，触摸系统服务会处理处于该状态的活动。 但是，当前发行版未提供 SYS Touch MTCH6303 支持。
APP_STATE_MAIN_MENU	主菜单状态
APP_STATE_SLIDE_SHOW	滑屏播放状态
APP_STATE_TOUCH_TEST	触摸测试状态
APP_STATE_SETTINGS	设置状态
APP_STATE_FEATURE_LIST	功能列表状态

说明

APP_STATES

该枚举定义有效的应用程序状态。这些状态决定应用程序在各种情况下的行为。

备注

具体信息请参见行内的注释。

APP_GLCD_LAYER0_BUFFER_ADDR 宏

定义 GLCD 图层 0 的辅助缓冲区的存储器基址

文件

wvga_glcd_app.h

C

```
#define APP_GLCD_LAYER0_BUFFER_ADDR GFX_GLCD_LAYER0_DBL_BASEADDR
```

说明

APP_GLCD_LAYER0_BUFFER_ADDR

定义该存储器地址位置是为了支持图层 0 的双缓冲渲染方案。

备注

所选择的地址空间处于 PIC32MZ2048DAB288 器件特有的 DDR2 地址空间 （0xA8000000-0xA9FFFFFF）。

APP_GLCD_LAYER1_ALPHA_RESOLUTION 宏

决定 Microchip 和 Harmony 徽标渐变和显示的平滑度

文件

wvga_glcd_app.h

C

```
#define APP_GLCD_LAYER1_ALPHA_RESOLUTION 7500
```

说明

APP_GLCD_LAYER1_ALPHA_RESOLUTION

提高 0 至 255 alpha 值范围的分辨率

备注

针对编译优化 O3 和 Os 进行微调

APP_GLCD_LAYER1_BUFFER_ADDR 宏

定义 GLCD 图层 1 的辅助缓冲区的存储器基址

文件

wvga_glcd_app.h

C

```
#define APP_GLCD_LAYER1_BUFFER_ADDR GFX_GLCD_LAYER1_DBL_BASEADDR
```

说明

APP_GLCD_LAYER1_BUFFER_ADDR

定义该存储器地址位置是为了支持图层 1 的双缓冲渲染方案。

备注

所选择的地址空间处于 PIC32MZ2048DAB288 器件特有的 DDR2 地址空间 （0xA8000000-0xA9FFFFFF）。

APP_GLCD_LAYER2_BUFFER_ADDR 宏

定义 GLCD 图层 2 的辅助缓冲区的存储器基址

文件

wvga_glcd_app.h

C

```
#define APP_GLCD_LAYER2_BUFFER_ADDR GFX_GLCD_LAYER2_DBL_BASEADDR
```

说明

APP_GLCD_LAYER2_BUFFER_ADDR

定义该存储器地址位置是为了支持图层 2 的资产预装入后台缓冲区渲染方案。

备注

所选择的地址空间处于 PIC32MZ2048DAB288 器件特有的 DDR2 地址空间 （0xA8000000-0xA9FFFFFF）。

APP_GLCD_LAYER2_HOR_RES 宏

GLCD 图层 2 的水平分辨率

文件

wvga_glcd_app.h

C

```
#define APP_GLCD_LAYER2_HOR_RES GFX_GLCD_LAYER2_RES_X
```

说明

APP_GLCD_LAYER2_HOR_RES

行缓冲区与屏幕像素的各种位置交叉映射需要该宏

备注

所选择的地址空间处于 PIC32MZ2048DAB288 器件特有的 DDR2 地址空间 （0xA8000000-0xA9FFFFFF）。

APP_HOME_MODE_SEMI_TRANSPARENT_ALPHA_VALUE 宏

首页模式使用该值来使 Microchip 和 Harmony 徽标保持在半透明级别

文件

wvga_glcd_app.h

C

```
#define APP_HOME_MODE_SEMI_TRANSPARENT_ALPHA_VALUE 180
```

说明

APP_HOME_MODE_SEMI_TRANSPARENT_ALPHA_VALUE

该值代表 RGBA8888 颜色方案中的水印状半透明模式

APP_TRANSPARENT_ALPHA_VALUE 宏

GLCD 驱动程序接受为 100% 透明的值

文件

wvga_glcd_app.h

C

```
#define APP_TRANSPARENT_ALPHA_VALUE 255
```

说明

APP_TRANSPARENT_ALPHA_VALUE

该值代表 RGBA8888 颜色方案中的 100% 透明模式

wvga_lcc

提供关于演示的信息。

说明

使用低成本无控制器功能和 MTCH6303 PCAP 触摸控制器以及 5" WVGA PCAP 显示板来演示图形库的当前功能 （见**注**）。

注： 关于获取 5" WVGA PCAP 显示板的信息，请与您当地的 Microchip 销售办事处联系。

该演示包含以下特性：

- WVGA 显示屏
- 集成 MTCH6303 PCAP 触摸输入
- 低成本无控制器 （LCC）图形驱动程序
- 16 位颜色深度支持 （65535 种唯一颜色）
- JPEG 和 BMP 图像存储在内部闪存中
- 实时 JPEG 解码和渲染
- 16 位 Unicode 字符字体支持
- 多语言本地化菜单 （繁体中文和简体中文）
- 控制台输出调试

编译应用程序

本主题指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了低成本
无控制器 （LCC） WVGA 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wvga_lcc.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/gfx/wvga_lcc`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wvga_lcc.X`	`<install-dir>/apps/gfx/wvga_lcc/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2_wvga	pic32mz_ec_sk+meb2+wvga	用于连接到 PIC32MZ EC 入门工具包的多媒体扩展板 II （MEB II）的 LCC 演示。
pic32mz_ef_sk_meb2_wvga	pic32mz_ef_sk+meb2+wvga	用于连接到 PIC32MZ EF 入门工具包的多媒体扩展板 II （MEB II）的 LCC 演示。

配置硬件

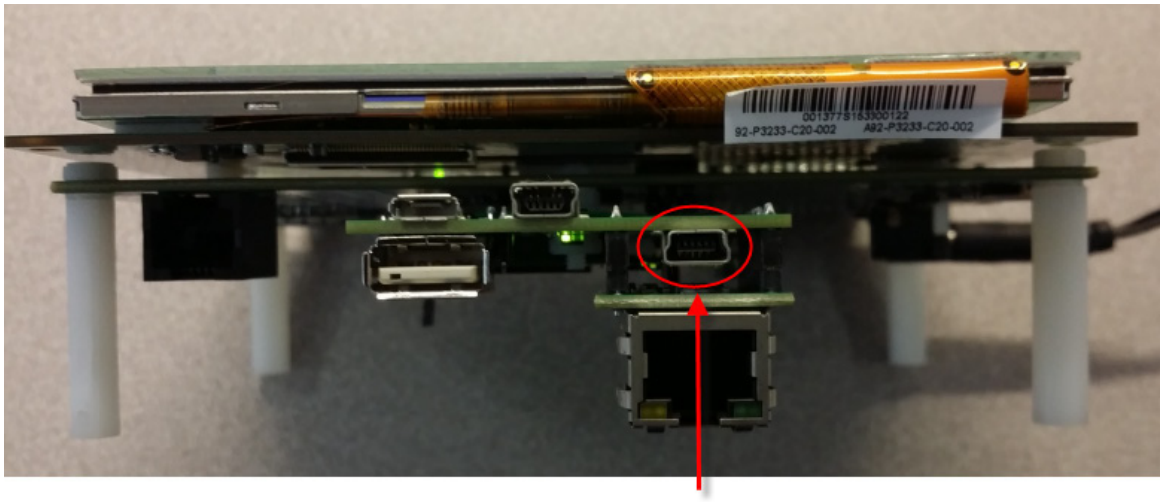
介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包和 MEB II

EBIOE 和 LCD_PCLK （J9）必须闭合。该跳线 （J9）位于 MEB II 扩展板的背面。

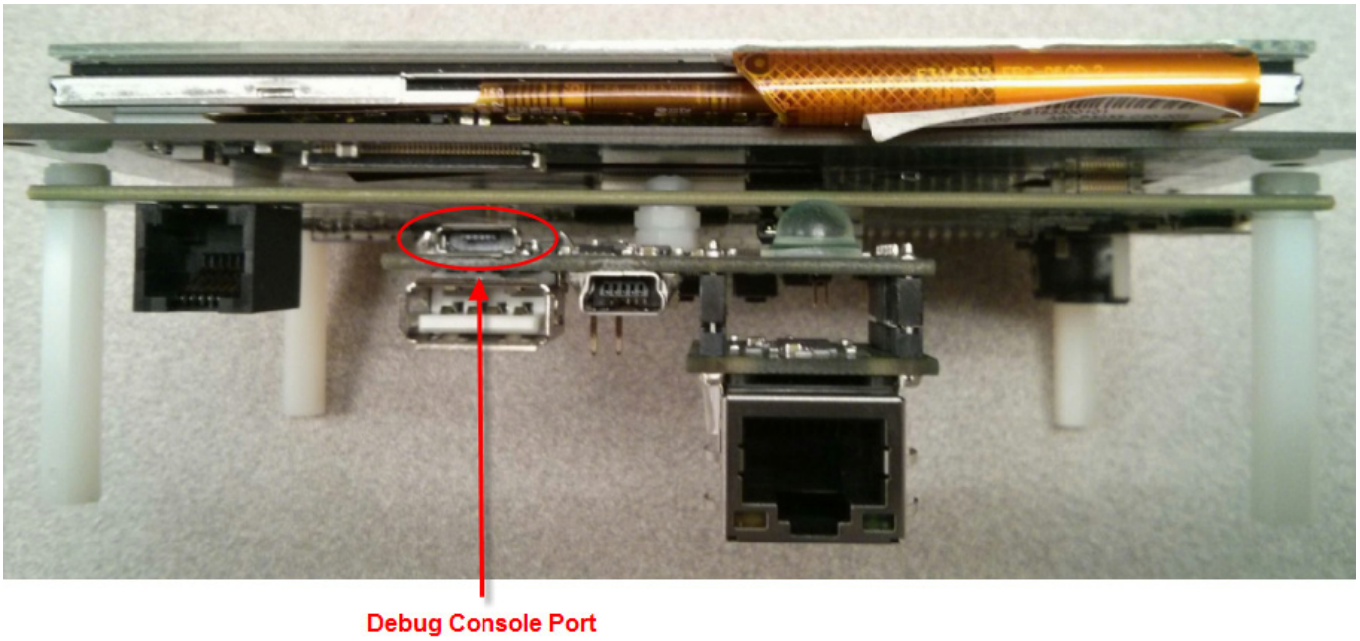
该演示还会通过 UART over USB 使能调试控制台。您需要一条 USB A 型转 USB B 型的电缆。将该电缆与端口连接，如下图所示。此外，使用以下串行端口设置来配置终端仿真器软件：8 位，无奇偶校验， 1 个停止位，无流量控制。



PIC32MZ EC 入门工具包和 MEB II

EBIOE 和 LCD_PCLK （J9）必须闭合。该跳线 （J9）位于 MEB II 扩展板的背面。

该演示还会通过 USB 使能调试控制台。您需要一条 USB A 型转 USB micro-A/B 的电缆。将该电缆与端口连接，如下图所示。此外，使用以下串行端口设置来配置终端仿真器软件：921600， 8 位，无奇偶校验， 1 个停止位，无流量控制。



注： USB 控制台的输出会进行缓冲，直到它读取单个字符后才会输出。这是由设计决定的，因为控制台模块无法确定终端窗口和 USB 连接是否已建立。

运行演示

介绍如何编译并运行 LCC WVGA 演示。

说明

在成功上电时，该演示会显示一个与下图所示类似的菜单：



该演示分为 4 个部分：

1. Slideshow （滑屏播放）模式包含一系列以 16 bpp 颜色深度渲染的 JPEG 图像。该演示会在定时器发生超时时循环到下一个滑屏，定时器可以通过 Settings 子菜单配置。点击图像会跳转到下一个滑屏，无须等待定时器。按下并保持会阻止图像发生循环。
2. Touch Test （触摸测试）模式会启动一个跟踪并记录触摸位置的简单屏幕。 X 和 Y 坐标通过右侧的数字仪表窗口小部件显示。 X/Y 坐标也会通过调试控制台显示。
3. 在 Settings 菜单下，您可以使用滑块来更改滑屏的时间。另外一项功能是该演示的多语言支持。您可以通过触摸任意单选按钮来更改对于整个演示显示的语言，可在英语和繁体或简体中文之间切换。
4. 演示的 Features （功能）可以通过主菜单上的最后一个选项来显示。

应用程序函数和原型

函数

	名称	说明
	APP_ChangeMode	更改当前应用程序模式
	APP_GenerateSysMsgGet	生成系统消息触摸数据
	APP_GoToNextSlide	将滑屏播放设置到下一个滑屏
	APP_HandleLanguageSetting	更改当前应用程序语言设置
	APP_HandleTouchTest	将当前 x/y 触摸值发送到 Touch Test 屏幕中的数字仪表
	APP_ProcessModeState	基于当前选定的模式更改当前应用程序状态
	APP_RedrawRectangle	在触摸测试屏幕中重新绘制矩形
	APP_SetSlidePauseTime	设置滑屏停顿间隔时间
	APP_TouchMessageCallback	触摸事件回调程序。
	APP_UpdateFeatureList	基于语言设置，更新要显示的功能列表
	APP_UpdateLanguageTexts	基于语言设置，更新要显示的设置文本
	APP_UpdateMainMenu	基于语言设置，更新要显示的主菜单文本
	APP_UpdateSlideShowTips	基于语言设置，更新要显示的滑屏播放指示信息文本

数据类型和常量

	名称	说明
	APP_LANGUAGES	该应用程序支持的语言的枚举

说明

列出可供演示使用的函数和原型，它们在 `app.h` 头文件中提供。

函数

APP_ChangeMode 函数

更改当前应用程序模式

文件

wvga_lcc_app.h

C

```
void APP_ChangeMode(APP_MODES mode);
```

说明

该函数设置为在屏幕按钮产生释放事件时调用，用于帮助在模式之间导航。在被调用时，该函数会设置所有必需的步骤，以切换到所请求的模式。

函数

void APP_ChangeMode(APP_MODES mode);

APP_GenerateSysMsgGet 函数

生成系统消息触摸数据

文件

wvga_lcc_app.h

C

```
bool APP_GenerateSysMsgGet();
```

返回值

如果触摸事件已处理，则为 true。

备注：该函数位于此处是因为 MTCH6303 驱动程序尚未被集成到 SYS Touch 中。该函数执行通常 SYS Touch 应执行的操作

说明

在 APP_Tasks() 每次循环中都会调用该函数，用于确定按下 / 仍然按下 / 释放情况，然后通过系统消息送出结果，消息的接收方应为 GOL 库，它使用这些数据来处理窗口小部件行为。

函数

bool APP_GenerateSysMsgGet(void)

APP_GoToNextSlide 函数

将滑屏播放设置到下一个滑屏

文件

wvga_lcc_app.h

C

```
void APP_GoToNextSlide();
```

说明

在被调用时，该函数会设置所有必需的步骤，以切换到下一个滑屏。

函数

void APP_GoToNextSlide (void);

APP_HandleLanguageSetting 函数

更改当前应用程序语言设置

文件

wvga_lcc_app.h

C

```
void APP_HandleLanguageSetting(APP_LANGUAGES language);
```

说明

该函数设置为在单选按钮产生选择语言的选择事件时调用。调用后，该函数会设置所有必需的步骤，以切换到所请求的语言。

函数

void APP_HandleLanguageSetting(APP_LANGUAGES language);

APP_HandleTouchTest 函数

将当前 x/y 触摸值发送到 Touch Test 屏幕中的数字仪表

文件

wvga_lcc_app.h

C

```
void APP_HandleTouchTest();
```

说明

负责将欧几里得坐标值报告给 Touch Test 屏幕中的数字仪表

函数

void APP_HandleTouchTest (void);

APP_ProcessModeState 函数

基于当前选定的模式更改当前应用程序状态

文件

wvga_lcc_app.h

C

```
void APP_ProcessModeState();
```

说明

如果用户请求更改模式，则会调用该函数来处理到新状态的切换。

函数

void APP_ProcessModeState (void);

APP_RedrawRectangle 函数

在触摸测试屏幕中重新绘制矩形

文件

wvga_lcc_app.h

C

```
void APP_RedrawRectangle();
```

说明

调用后，该函数会通过在触摸屏测试区域重画一个矩形来清除测试区域。

函数

void APP_RedrawRectangle(void);

APP_SetSlidePauseTime 函数

设置滑屏停顿间隔时间

文件

wvga_lcc_app.h

C

```
void APP_SetSlidePauseTime(int timeInSec);
```

说明

该函数设置为在 Settings 菜单中的滑块栏产生触摸事件时调用。

函数

void APP_SetSlidePauseTime(int timeInSec);

APP_TouchMessageCallback 函数

触摸事件回调程序。

文件

wvga_lcc_app.h

C

```
void APP_TouchMessageCallback(SYS_MSG_OBJECT * pMsg);
```

说明

它是来自 GFX_HGC 层的回调，用于通知应用程序发生了触摸事件。 pMsg 包含当前触摸行为，触摸位置则通过触摸系统服务报告

备注

该函数在 APP_Initialize() 中通过 GFX_HGC_RegisterAppTouchCallback 注册。只有在不使用 MTCH6303 触摸驱动程序时，才会在编译中包含该函数。这是因为 MPLAB Harmony v1.07 的 MTCH6303 触摸驱动程序不支持触摸系统服务。

函数

void APP_TouchMessageCallback (SYS_MSG_OBJECT *pMsg);

APP_UpdateFeatureList 函数

基于语言设置，更新要显示的功能列表

文件

wvga_lcc_app.h

C

```
void APP_UpdateFeatureList();
```

说明

在处于功能列表模式期间， APP_Tasks() 会调用该函数，用于基于语言设置 （英语、繁体中文和简体中文）绘制要显示的一组正确的功能列表

函数

void APP_UpdateFeatureList(void);

APP_UpdateLanguageTexts 函数

基于语言设置，更新要显示的设置文本

文件

wvga_lcc_app.h

C

```
void APP_UpdateLanguageTexts();
```

说明

在处于设置模式期间，APP_Tasks()会调用该函数，用于基于语言设置（英语、繁体中文和简体中文）绘制要显示的一组正确的设置文本

函数

void APP_UpdateLanguageTexts(void);

APP_UpdateMainMenu 函数

基于语言设置，更新要显示的主菜单文本

文件

wvga_lcc_app.h

C

```
void APP_UpdateMainMenu();
```

说明

在处于主菜单模式期间，APP_Tasks() 会调用该函数，用于基于语言设置（英语、繁体中文和简体中文）绘制要显示的一组正确的文本

函数

void APP_UpdateMainMenu(void);

APP_UpdateSlideShowTips 函数

基于语言设置，更新要显示的滑屏播放指示信息文本

文件

wvga_lcc_app.h

C

```
void APP_UpdateSlideShowTips();
```

说明

在处于滑屏播放模式期间，APP_Tasks()会调用该函数，用于基于语言设置（英语、繁体中文和简体中文）绘制要显示的一组正确的文本

函数

void APP_UpdateSlideShowTips(void);

数据类型和常量

APP_LANGUAGES 枚举

该应用程序支持的语言的枚举

文件

wvga_lcc_app.h

C

```
typedef enum {
  APP_LANG_ENGLISH,
  APP_LANG_CHINESE_TRAD,
  APP_LANG_CHINESE_SIMP
} APP_LANGUAGES;
```

说明

APP_LANGUAGES

该枚举用于帮助跟踪该应用程序中支持的三种语言：英语、繁体中文和简体中文。

MEB II 演示

本帮助文件介绍 SEGGER emWin 多媒体扩展板 II （MEB II）演示。

简介

SEGGER emWin MEB II 演示应用程序帮助

说明

本帮助文件介绍硬件要求，以及在 Microchip 演示和开发板上运行演示 SEGGER emWin 应用程序功能的 MEB II 相关固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

演示

本主题介绍如何运行本发行版中包含的 MEB II 演示应用程序。

gfx_camera

介绍 gfx_camera 演示和应用程序要求。

说明

gfx_camera 演示（`apps/meb_ii/gfx_camera/firmware/gfx_camera.X`）说明如何运用在 MEB II 上运行的 Omnivision 摄像头传感器。它利用 LCC 图形功能与 PIC32 的 DMA 功能将图形从摄像头传感器传到 LCD，而无需图形控制器。

应用程序过程

在运行演示之前，应用程序需要执行以下步骤：

1. 系统应已完成必需的设置初始化。
2. 应已通过调用 DRV_I2C_Initialize 初始化 I2C 驱动程序对象。
3. 应已通过调用 DRV_Timer_Initialize 初始化定时器驱动程序对象。
4. 应已通过调用 DRV_OC_Initialize 初始化输出控制驱动程序对象。
5. 应已通过调用 DRV_CAMERA_OVM7690_Initialize 初始化摄像头 OVM7690 驱动程序对象。
6. 通过调用 DRV_CAMERA_OVM7690_Open 打开摄像头 OVM7690 驱动程序客户端。
7. 通过调用 DRV_CAMERA_OVM7690_FrameBufferAddressSet 将图形帧缓冲区地址传递给摄像头 OVM7690 驱动程序。
8. 通过调用 DRV_CAMERA_OVM7690_FrameRectSet 设置帧矩形区域。
9. 通过调用 DRV_CAMERA_OVM7690_RegisterSet 设置其他摄像头设置，如：软复位、使能 pclk、使能 href、使能 vsync、输出颜色格式、反转 HREF 极性、送至 HREF 的门控时钟、像素时钟频率和欠采样模式。
10. 通过调用 DRV_CAMERA_OVM7690_Start 启动摄像头 OVM7690。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了图形库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_camera.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/meb_ii/gfx_camera`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
gfx_camera.X	<install-dir>/apps/meb_ii/gfx_camera/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包的多媒体扩展板 II （MEB II）上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

MEB II 上的跳线 J9 应设置到 （EBIWE - LCD_PCLK）。

运行演示

介绍如何编译并运行演示。

说明

使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 使用工具包中提供的 USB 电缆将入门工具包板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。
3. 编译、下载演示项目并在目标板上运行。
4. WQVGA LCD 上应出现一个图像，显示来自摄像头传感器的信息。

gfx_cdc_com_port_single

该演示应用程序会创建在主机个人计算机上枚举为单个 COM 端口的 GFX USB CDC 设备。该应用程序演示了 USB 设备和个人计算机主机之间的双向通信。该应用程序允许用户从 PC 主机 USB/ 串行程序中向 GFX UI 键盘编辑框显示中输入键盘数字和退格键。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了图形库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 gfx_cdc_com_port_single.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/meb_ii/gfx_cdc_com_port_single。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
gfx_cdc_com_port_single.X	<install-dir>/apps/meb_ii/gfx_cdc_com_port_single/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ 嵌入式连接 （EC）入门工具包的多媒体扩展板 II （MEB II）上运行演示。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包的多媒体扩展板 II （MEB II）上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行演示。

说明

该演示让设备对于主机来说就像是一个串行 （COM）端口。

1. 要运行该演示，需要先编译并对目标设备编程。在编译时，选择对应于所用入门工具包的 pic32mz_ec_sk_meb2 或 pic32mz_ef_sk_meb2 配置。将设备连接到主机。如果主机是一台个人计算机，并且这是第一次将该设备插入计算机中，则可能会要求您安装 .inf 文件。
2. 选择 “从列表或指定位置安装 （高级）” 选项。指向 <install_dir>/apps/usb/cdc_com_port_single/inf 目录。
3. 成功安装设备之后，打开一个终端程序，例如 HyperTerminal。选择适当的 COM 端口。在大多数计算机上，它将为 COM4 或更高编号的端口。
4. 连接设备之后，有两种方式可以运行该示例项目。在终端窗口中输入一个数字键值会导致设备在终端上回显被按下的键值；在所显示的文本框中输入时，会回显到 MEB II 扩展板显示屏上。

注： 一些终端程序（如 HyperTerminal）要求用户先单击断开连接按钮，之后才能从计算机中移除设备。如果不这样做，则为了重新连接设备，可能必须重新关闭并打开程序。

gfx_photo_frame

该演示应用程序说明 GFX、 USB 中间件和 SQI 的功能。在该演示中，使用片外 SQI 闪存 （SST26VF032）作为通过 USB CDC、XMODEM 协议存储图像的介质，然后通过 GFX 协议栈在 MEB II 显示屏上渲染这些图像。

该演示包含 4 个示例 WQVGA 对齐图像，它们被下载到 SQI 闪存上的不同位置，然后通过 COM 端口命令在屏幕上显示，并且还通过按钮渲染为滑屏。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了图形照片帧演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_photo_frame.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/meb_ii/gfx_photo_frame`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_photo_frame.X`	`<install-dir>/apps/meb_ii/gfx_photo_frame/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包的多媒体扩展板 II （MEB II）上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包和 MEB II。

MEB II 具有两种存储器选项：外部或内部。仅内部模式支持该演示。在运行演示之前，请确保 MEB II 上的 EBIWE 和 LCD_PCLK（J9）引脚闭合。

运行演示

介绍如何编译并运行演示。

说明

使用以下过程来运行演示：

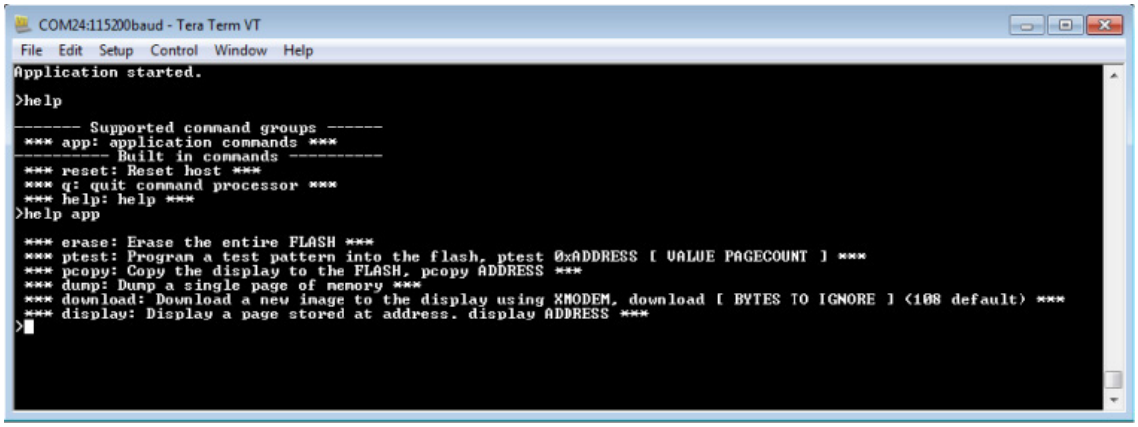
1. 将演示项目装入 MPLAB X IDE。
2. 将 MEB II 连接到 PIC32MZ EC 入门工具包。
3. 使用工具包中提供的 USB 电缆将入门工具包板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。此外，连接入门工具包背面的 micro-B USB 连接器。这会在 USB 总线上创建一个 USB CDC 设备。为了与软件进行通信，通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。

   注： 如果开发计算机无法自动识别 USB 驱动程序，请使用 `<install-dir>/apps/meb_ii/gfx_photo_frame/inf` 中提供的驱动程序支持文件来更新驱动程序。

4. 编译、下载演示项目并在目标板上运行。
5. 在 MEB II 显示屏变为空白时，启动控制台客户端 （操作系统本机客户端 /Tera Term 等），选择串行端口，并将串行端口设置设为 115200-N-1。

   注： 主要关注控制台客户端，按 Enter 键时控制台中应显示 “Application started”（应用程序已启动）字符串，指示应用程序已正确启动。

6. 在控制台客户端中再次按下 Enter 键可看到 “ >”提示符，指示演示已准备好接收命令。
7. 发送 “help”命令可查看演示支持的命令的列表。



8. 发送 “erase”命令可擦除整个 SQI 闪存。控制台将返回 “Erasing FLASH”（正在擦除闪存），接着返回 “Erase complete”（擦除完成），指示擦除操作完成。
9. 发送 “display 0x00000000”命令 （它会显示 SQI 闪存地址 0 处的帧）时，可以注意到 MEB II 显示屏变为白色，因为在已擦除状态下， SQI 闪存中全为 F 字符，而它会被装入帧缓冲区。
10. 发送 “ptest 0x00000000 0xf800 0x20”命令可向 SQI 闪存中装入测试模式，它会向闪存中的前 32 页写入 0x20 （红色）模式。在将测试模式装入 SQI 闪存后，控制台将返回 “Programming test pattern complete”（编程测试模式完成）。
11. 发送 “display 0x00000000”命令时，可以注意到显示屏上的前几行变为红色。
12. 发送 “erase”命令可擦除整个 SQI 闪存。
13. 要将随演示提供的图像下载到闪存中，请重复以下步骤：

- 发送 download [ignore bytes] 命令

注： 参数 ignore bytes 不是必需的。默认情况下，演示将使用以下值：108。已知的忽略值为 108、 124 和 188。该演示中的第 1 个、第 3 个和第 4 个图像使用 188。第 2 个图像将使用默认值，不需要指定忽略值。演示中的每个图像需要 0x3FC00 的闪存。

- 发送 download [ignore bytes] 命令。控制台将返回 “Start XMODEM download”（启动 XMODEM 下载），指示 XMODEM 已就绪
- 选择 *File （文件）*>*Transfer （传输）*> *XMODEM-Send （XMODEM 发送）*并使用传统校验和 XMODEM 传输，然后通过浏览至 `<install-dir>/apps/meb_ii/gfx_photo_frame` 选择要发送的图像并单击 **send** （发送）。这会将图像直接装入 GFX 帧缓冲区，并在显示屏上显示图像。

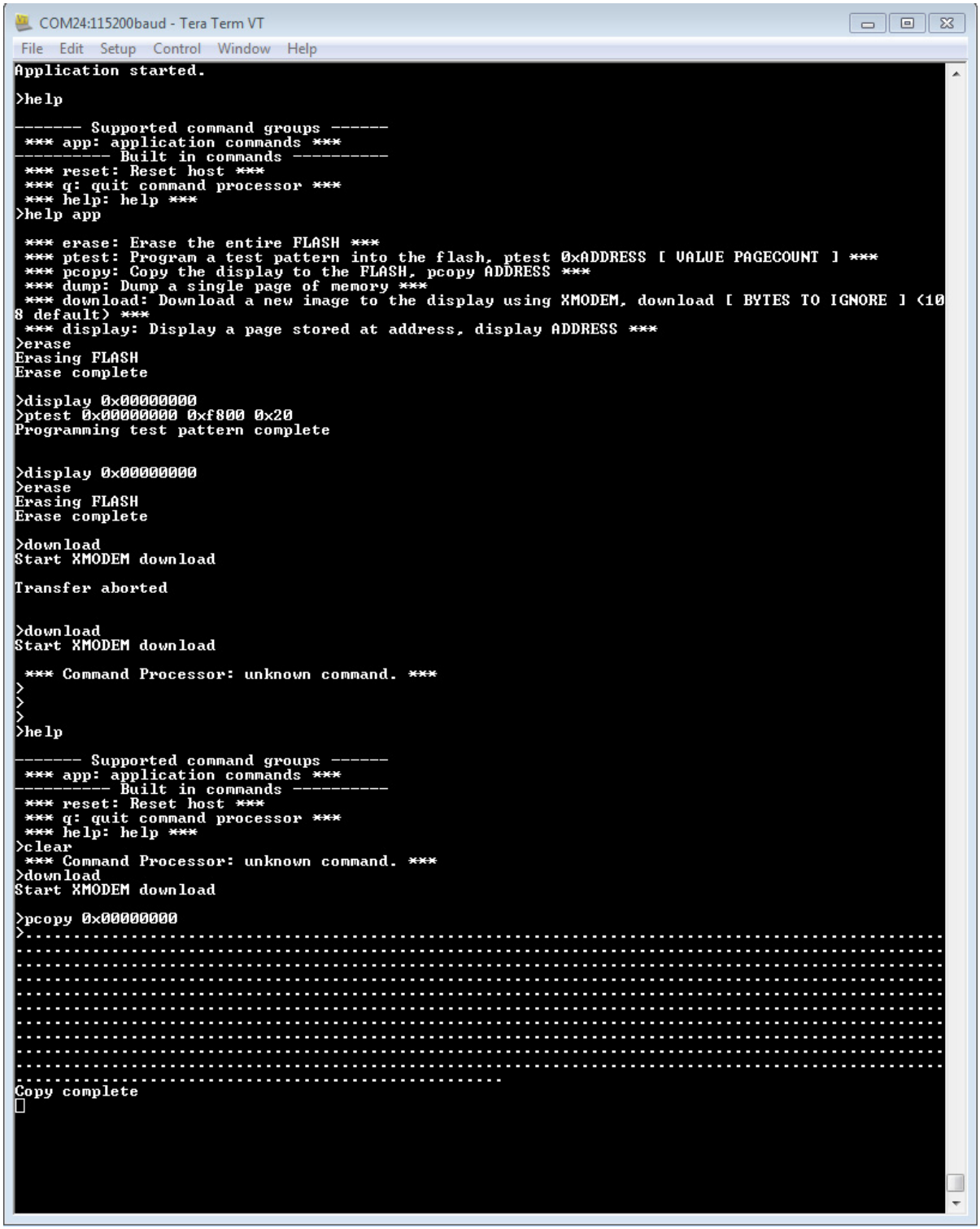
注： XMODEM 有时是不稳定的，在这种情况下，必须重复该步骤。

- 发送 “pcopy <ADDRESS>”命令将图像复制到 SQI 闪存的 <ADDRESS> 位置。控制台会返回以下信息来指示状态：

‘.....
.....
.....
.....
.....
.....
Copy complete'

注： 对于第1个图像，ADDRESS = 0x00000000，对于第2个图像，它为0x0003FC00，对于第3个图像，它为0x0007F800，对于第 4 个图像，它为 0x000BF400。

14. 将所有图像复制到 SQI 闪存中之后，发送 “display <ADDRESS>”命令可将存储的图像装入显示屏。
15. 通过按下 MEB II 上的 S1 开关，可以在滑屏播放模式下查看图像。
16. 关于前面步骤中介绍的命令的摘要，请参见下面的控制台窗口截图：



gfx_web_server_nvm_mpfs

GFX TCP/IP NVM MPFS Web 服务器演示（`apps\meb_ii\gfx_web_server_nvm_mpfs\firmware\pic32_eth_web_server.X`）说明如何运用在 PIC32 器件上运行的 HTTP Web 服务器。非易失性存储器（Non-Volatile Memory，NVM） Microchip 专有文件系统（Microchip Proprietary File System，MPFS） Web 服务器演示将网页存储在内部闪存中，并通过 MPFS API 访问。 Web 服务器的 IP 地址会显示在图形 UI 中。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了图形 Web 服务器 NVM MPFS 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `pic32_eth_web_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/meb_ii/gfx_web_server_nvm_mpfs`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`pic32_eth_web_server.X`	`<install-dir>/apps/meb_ii/gfx_web_server_nvm_mpfs/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ 嵌入式连接 （EC）入门工具包的多媒体扩展板 II （MEB II）上运行演示。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包的多媒体扩展板 II （MEB II）上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行演示。

说明

要查看演示应用程序托管的网页，可以打开 Web 浏览器，并通过在地址栏中输入 URL （如 http://mchpboard_e）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

注： TCP/IP 应用程序的 NetBIOS 名称在初始化 TCP/IP 协议栈时指定，它通常处于 tcpip_stack_init.c:: TCPIP_HOSTS_CONFIGURATION 结构的 hostName 成员中。必须使能 NetBIOS 服务， PIC32 演示才会响应 NetBIOS 查询。或者，也可以直接使用电路板的 IPv4 或 IPv6 地址 （如果使能了 IPv6），例如， http://192.168.1.131 或 http://[fdfe:dcba:9876:1:204:a3ff:fe12:128e]。

请使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 使用工具包中提供的 USB 电缆将入门工具包板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。

3. 使用工具包中提供的以太网接插线，将入门工具包上的 RJ-45 以太网端口连接到网络集线器或开发计算机上的以太网端口。
4. 编译、下载演示项目并在目标板上运行。
5. 演示应用程序托管了一个 HTTP 服务器。打开 Web 浏览器，并输入 `http://<ip address>`。IP 地址会显示在图形用户界面上。
   - *Real-time hardware control and Dynamic Variables* （实时硬件控制和动态变量） —— 在 Overview （概述）页面中，可以通过单击 LED 来翻转入门工具包上的 LED。按下电路板上的按钮时，可以看到网页上的按钮发生切换。动态变量可以在 HTTP 服务器上实时更新。

注： 由于与 GPIO 和以太网引脚上的功能复用相关的勘误项，演示的 LED 功能部分会受到一定限制。

- *Form Processing* （表单处理） —— 可以通过使用 GET 和 POST 方法来处理来自客户端的输入。
- *Authentication* （认证） —— 给出一个常用的受限访问功能的示例。
- *Cookies*—— 给出一个在客户端存储小文本字符串的示例。
- *File Uploads* （文件上传） —— 给出一个使用 POST 方法进行文件上传的示例。 HTTP 服务器可以接受用户定义的 MPFS/MPFS2 映像文件，使用它来供应网页。
- *Send E-mail* （发送电子邮件） —— 说明简单的 SMTP POST 方法。
- *Dynamic DNS* （动态 DNS） —— 运用动态 DNS 功能。
- *Network Configuration*（网络配置） —— 可以在 Network Configuration 页面中查看 PIC32 EC 入门工具包电路板的 MAC 地址、主机名和 IP 地址，并且可以更新一些配置。

segger_emwin

该演示使用 LCD 显示屏上的软件图形控制器来说明 SEGGER emWin 图形库的基本和高级功能。

说明

segger_emwin 演示能够将 SEGGER emWin 图形库支持的多项功能中的一部分显示为简单的图形用户界面 （GUI）演示。

SEGGER emWin 旨在为任何使用图形 LCD 工作的应用程序提供独立于处理器和 LCD 控制器的高效 GUI。在应用程序中演示的一些功能包括：显示 alpha 混合、精灵和圆形菜单操作的 GUI，列表和树窗口小部件功能，多图层和在图层上绘制，使用不同格式的图像，库提供的着色功能等。

目前，该演示包含了位于媒体扩展板 II （MEB II）上的 PIC32MZ 嵌入式连接 （EC）入门工具包和 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包，而媒体扩展板 II （MEB II）在 WQVGA LCD 显示屏上具有无控制器图形功能。

该演示需要使用内部存储器 （SRAM）。必须使用硬件跳线配置存储器设置，并且需要选择相应的 MHC 设置。

重要提示！ ☆	要设置内部存储器，必须在电路板上设置跳线。未配置该跳线设置会使显示屏无法工作，虽然软件可能仍会运行。关于相应跳线设置的详细信息，请参见配置硬件。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SEGGER emWin MEB II 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `segger_emwin.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/meb_ii/segger_emwin`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
segger_emwin.X	<install-dir>/apps/meb_ii/segger_emwin/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ EC 入门工具包的多媒体扩展板 II （MEB II）上运行演示。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	选择该 MPLAB X IDE 项目配置，可以在连接到 PIC32MZ EF 入门工具包的多媒体扩展板 II （MEB II）上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

MEB II 具有两种存储器设置选项：外部或内部；但是，该演示必须通过实现以下设置在内部模式下运行：

- EBIWE 和 LCD_PCLK （J9）必须闭合。该跳线位于 MEB II 背面。
- 确保所有 MHC 设置都定义为仅使用内部 SRAM。应用程序当前的内部 / 外部存储器设置可以通过选择 *Harmony Framework Configuration* > *Drivers* > *Graphics Controllers* > *LCC* > *Use LCC Driver* > *Memory Interface Mode* 来验证。

运行演示

介绍如何编译并运行演示。

说明

该演示说明图形库如何与低成本无控制器 （LCC）软件显示控制器接口。该演示会将 SEGGER emWin 图形库支持的多项功能中的一部分显示为简单的 GUI 演示。

使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 编译、下载并在目标板上运行演示项目 （见**注**）。
3. 它是一个只有 GUI 的演示；它不处理任何触摸输入。

注： 在编译时， pic32mz_ef_sk_mebii 配置会发出与浮点单元相关的警告。这是因为 emWin 不支持浮点，虽然 pic32mz_ek_sk 具有浮点单元。

演示屏幕

以下是演示各种图形功能的不同屏幕：

- **Radial menu** （圆形菜单） —— 从圆形菜单中选择图标。使用 emWin 移动支持来完成选择更改。
- **Bargraph demo** （条形图演示） —— 使用 alpha 混合来显示一个条形图
- **Antialiased text** （抗锯齿文本） —— 显示具有不同抗锯齿质量的抗锯齿文本。在不同背景 （2 bpp 和 4 bpp）上输出抗锯齿文本
- **Transparent dialog** （透明对话框） —— 使用 alpha 混合在运动背景上产生透明效果
- **Washing machine** （洗衣机） —— 显示一个洗衣机演示，会有一条蓝色海豚精灵在应用程序顶部移动
- **Iconview demo** （Iconview 演示） —— 使用 ICONVIEW 窗口小部件来显示基于图标的菜单，手持设备中经常会需要它。显示选择更改和图标文本对齐方式更改。

- **Treeview widget**（Treeview 窗口小部件）—— 显示定制的 TREEVIEW 窗口小部件。演示目录中的项和一些精灵构成的分层视图、显示 / 隐藏线条、移动光标、打开 / 关闭节点、更改选择模式，以及设置图像。
- **Listview widget**（Listview 窗口小部件）—— 说明如何使用 LISTVIEW 窗口小部件。演示如何更改顺序、使能排序、使用反向 / 正常排序顺序、移动行，以及对列表中行 / 列的各个元素进行着色。
- **Drawing a graph**（绘制图形）—— 使用 GRAPH 窗口小部件来可视化函数图（如心跳和正弦波）
- **High speed**（高速）—— 演示多图层裁剪和高度优化的驱动程序
- **Pixels speed**（像素速度）—— 以每秒像素数的方式演示像素速度
- **Bitmaps**（位图）—— 通过显示 Windows 目录的所有位图来演示 *.BMP 文件。基于调色板的位图，更改调色板，bmp、gif、jpeg，12、 16 和 24 bpp 格式， alpha 位图，更改颜色，灰度级位图。
- **Color bar**（色条）—— 显示含有梯度条形（黑色 > 彩色，白色 > 彩色）的色条。集成颜色管理，它可以确定任意逻辑颜色的优化颜色。

RTOS 演示

本主题介绍 RTOS 演示。

简介

RTOS 演示应用程序帮助

说明

该分发包包含一系列基于 RTOS 的固件项目，用于演示与在 PIC32 器件上运行的 RTOS 集成的 MPLAB Harmony 服务和协议栈的功能。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

要了解关于 MPLAB Harmony 协议栈和库的更多信息，请参见 Volume IV: MPLAB Harmony Framework Reference 中的相关文档。

源代码免责声明

OPENRTOS

MPLAB Harmony 中提供的 OPENRTOS 演示使用 OPENRTOS 评估许可证，它仅用于演示目的。希望在 OPENRTOS 上进行开发和生产活动的客户必须获取适当的许可证。关于获取适用于您所用器件的评估许可证的信息，请参见以下文档之一，它们位于 MPLAB Harmony 安装的 `third-party` 文件夹中：

- `OpenRTOS Click Thru Eval License PIC32MXxx.pdf`
- `OpenRTOS Click Thru Eval License PIC32MZxx.pdf`

Micriμm

所有 μC/OS-III 演示都在项目中添加了 `crt0.S` “C”运行时库启动文件。演示设置了链接器选项“do not link startup code”（不链接启动代码）。为使 μC/OS-III 可与 PIC32 器件正确配合工作，这是必需的，因为通用异常向量位于 `crt0.S` 中。 μC/OS-III 会改写该中断源 （通用异常处理程序），用于执行特定于操作系统的功能。

如果用户希望使用 μC/OS-III 和 PIC32 器件实现自己的应用程序，必须将 crt0.S 文件添加到到自己的项目中，并改写通用异常中断向量。关于其实现，请参见当前 RTOS 示例。

`crt0.S` 模板文件位于 MPLAB XC32 C/C++ 编译器安装目录中：`..\Microchip\xc32\<version>\pic32-libs\libpic32`。

重要提示！ 随 MPLAB Harmony 分发的 Micriμm μC/OS-II 和 μC/OS-III 源代码是用于免费短期评估、教育用途或私下研究的。如果计划或打算在商业应用程序 / 产品中使用 μC/OS-II 和 μC/OS-III，则需要联系 Micriμm，为在您的应用程序 / 产品中使用 μC/OS-II 和 μC/OS-III 获取适当的许可证。源代码是为方便您而提供的，用于帮助您体验 μC/OS-II 和 μC/OS-III。提供源代码并不意味着您可以不支付许可费而将它用于商业目的。关于源代码的知识不能用于开发类似产品。如果您不确定是否需要为应用程序获取许可证，请与 Micriμm 联系，并与销售代表讨论预期的用途（www.micrium.com）。

Express Logic ThreadX

ThreadX RTOS 的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 Express Logic ThreadX 网站：http://rtos.com/products/threadx/。

SEGGER embOS

SEGGER embOS RTOS 的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 SEGGER embOS 网站：https://www.segger.com/license-models.html。

Express Logic ThreadX 演示

basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

源代码免责声明

该 ThreadX RTOS 演示的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 Express Logic ThreadX 网站：http://rtos.com/products/threadx/。为使 ThreadX 能与适用的 MPLAB Harmony 演示配合工作，请将源代码安装到以下位置：`<install-dir>/third_party/rtos/ThreadX/`。

除非提供并在正确位置安装了源代码，否则将无法编译演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ThreadX 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `basic_threadx.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/threadx/basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`basic_threadx.X`	`<install-dir>/apps/rtos/threadx/basic`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mz_da_sk	pic32mz_da_sk	该配置在 PIC32MZ DA 入门工具包上运行。
pic32mx_sk	pic32mx_usb_sk2、 pic32mx_usb_sk3、pic32mx_eth_sk 和 pic32mx_eth_sk2	该配置在基于 PIC32MX 的入门工具包上运行：PIC32 以太网入门工具包、 PIC32 以太网入门工具包 II、 PIC32 USB 入门工具包 II 和 PIC32 USB 入门工具包 III。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包、 PIC32 以太网入门工具包 II、 PIC32 USB 入门工具包 II 和 PIC32 USB 入门工具包 III。

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 ThreadX 基本演示。

说明

当演示启动并在运行时， LED 会每 500 ms 翻转一次。该演示将显示 RTOS 正在使用所选硬件运行。

gfx

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

源代码免责声明

该 ThreadX RTOS 演示的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 Express Logic ThreadX 网站：http://rtos.com/products/threadx/。为使 ThreadX 能与适用的 MPLAB Harmony 演示配合工作，请将源代码安装到以下位置：`<install-dir>/third_party/rtos/ThreadX/`。

除非提供并在正确位置安装了源代码，否则将无法编译演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ThreadX 和 MPLAB Harmony 图形库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_threadx.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/threadx/gfx`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_threadx.X`	`<install-dir>/apps/rtos/threadx/gfx`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包和 MEB II 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 ThreadX 和 MPLAB Harmony 图形库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。LCD 屏幕颜色将每 500 ms 改变一次，并且用户 LED 将每 500 ms 翻转一次。

该应用程序中存在两个任务，并且该应用程序 / 系统中存在三个中断。

优先级最低的任务 LEDblinkTask 控制 LED 的翻转。优先级最高的任务 DisplayTask 会通过调用相应的 MPLAB Harmony 图形库函数来更改 LCD 屏幕的颜色。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序

gfx_usb

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

源代码免责声明

该 ThreadX RTOS 演示的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 Express Logic ThreadX 网站：http://rtos.com/products/threadx/。为使 ThreadX 能与适用的 MPLAB Harmony 演示配合工作，请将源代码安装到以下位置：`<install-dir>/third_party/rtos/ThreadX/`。

除非提供并在正确位置安装了源代码，否则将无法编译演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ThreadX 和 MPLAB Harmony 图形加 USB 库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_usb_threadx.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/threadx`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_usb_threadx.X`	`<install-dir>/apps/rtos/threadx/gfx_usb`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包和 MEB II 上运行。
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 ThreadX 和 MPLAB Harmony 图形加 USB 库。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。当 micro USB 电缆插入 PIC32MZ EC 入门工具包上的 J4 时，将会仿真一个 USB 键盘。按下按钮 SW3 时，将会发送键击 a-z 和 1-9。 LCD 屏幕颜色将每 500 ms 改变一次，并且用户 LED （D3）将每 500 ms 翻转一次。

该应用程序 / 系统中存在四个任务和五个中断。

SystemUSBDeviceTask 是优先级最高的任务，负责获取由应用程序排入队列的已准备好通过调用相应 MPLAB Harmony USB 协议栈函数经由 USB 发送的所有 USB 数据。 ApplicationUSBDeviceTask 是优先级次高的任务。当检测到 SW3 按钮被按下时，ApplicationUSBDeviceTask 会仿真键盘按键按下。 ApplicationDisplayTask 会通过调用相应的 MPLAB Harmony 图形库函数来更改整个 LCD 屏幕的颜色。 ApplicationLEDblinkTask 是优先级最低的任务，它会每 500 ms 翻转用户 LED （D3）一次。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。 Timer2 硬件中断用于调用 MPLAB Harmony 定时器驱动程序函数。这是 MPLAB Harmony USB 协议栈的硬件要求。 DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。USB 通用事件中断用于调用相应的 MPLAB Harmony USB 驱动程序函数，该函数会获取由应用程序排入队列的所有 USB 数据，并实际将它写出到 USB 硬件上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

usb

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

源代码免责声明

该 ThreadX RTOS 演示的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 Express Logic ThreadX 网站：http://rtos.com/products/threadx/。为使 ThreadX 能与适用的 MPLAB Harmony 演示配合工作，请将源代码安装到以下位置：`<install-dir>/third_party/rtos/ThreadX/`。

除非提供并在正确位置安装了源代码，否则将无法编译演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 ThreadX 和 MPLAB Harmony 图形加 USB 库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `usb_threadx.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/threadx/usb`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`usb_threadx.X`	`<install-dir>/apps/rtos/threadx/usb`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mx_usb_sk2	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 ThreadX 和 MPLAB Harmony USB 库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。当 micro USB 电缆插入 PIC32MZ EC 入门工具包上的 J4 时，将会仿真一个 USB 键盘。按下按钮 SW3 时，将会发送键击 a-z 和 1-9。用户 LED （D3）将每 500 ms 翻转一次。

该应用程序 / 系统中存在三个任务和四个中断。

SystemUSBDeviceTask 是优先级最高的任务，负责获取由应用程序排入队列的已准备好通过调用相应 MPLAB Harmony USB 协议栈函数经由 USB 发送的所有 USB 数据。 ApplicationUSBDeviceTask 是优先级次高的任务。当检测到 SW3 按钮被按下时，ApplicationUSBDeviceTask 会仿真键盘按键按下。 ApplicationLEDblinkTask 是优先级最低的任务，它会每 500 ms 翻转用户 LED （D3）一次。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。 Timer2 硬件中断用于调用 MPLAB Harmony 定时器驱动程序函数。USB 通用事件中断用于调用相应的 MPLAB Harmony USB 驱动程序函数，该函数会获取由应用程序排入队列的所有 USB 数据，并实际将它写出到 USB 硬件上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

FreeRTOS 演示

basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会使入门工具包上的 3 个用户 LED 发生闪烁，以显示 RTOS 线程正在运行并指示状态。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 FreeRTOS 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `basic_freertos.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/freertos`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`basic_freertos.X`	`<install-dir>/apps/rtos/freertos/basic`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_sk	pic32mx_usb_sk2、 pic32mx_usb_sk3、 pic32mx_eth_sk 和 pic32mx_eth_sk2	该配置在基于 PIC32MX 的入门工具包上运行：PIC32 以太网入门工具包、 PIC32 以太网入门工具包 II、 PIC32 USB 入门工具包 II 和 PIC32 USB 入门工具包 III。
pic32mz_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mz_da_sk	pic32mz_da_sk	该配置在 PIC32MZ DA 入门工具包上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包、 PIC32 以太网入门工具包 II、 PIC32 USB 入门工具包 II 和 PIC32 USB 入门工具包 III。

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 FreeRTOS 基本演示。

说明

请使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 使用工具包中提供的 USB 电缆将 PIC32MX 或 PIC32MZ 目标板上板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。
3. 编译、下载演示项目并在目标板上运行。

该演示应用程序具有以下功能：

- 应用程序会创建一个队列和四个任务。一个任务使用 FreeRTOS 队列向两个等待队列中数据的任务发送数据。（QueueReceiveTask2 优先级高于 QueueReceiveTask1 优先级）。
- QueueReceiveTask2 会先接收数据、翻转 LED，然后休眠指定的时间
- QueueReceiveTask1 从 QueueReceiveTask2 不处于运行状态开始接收接下来的数据
- QueueReceiveTask1 会接收数据、翻转 LED 并等待数据到达

cdc_com_port_dual

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

这个基于 RTOS 的演示应用程序会创建在 USB 主机个人计算机上枚举为两个串行端口的 USB CDC 设备。该应用程序演示 MPLAB Harmony USB 协议栈在实时操作系统 （该示例使用 FreeRTOS）中工作以及支持相同设备类的多个实例的能力。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了该演示应用程序的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_com_port_dual.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/freertos/cdc_com_port_dual`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_com_port_dual.X`	`<install-dir>/apps/rtos/freertos/cdc_com_port_dual/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该配置可以在 PIC32 USB 入门工具包 II 上以中断模式和动态操作方式运行演示应用程序。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上以中断模式和动态操作方式运行演示应用程序。

pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上以中断模式和动态操作方式运行演示应用程序。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

移除跳线 JP1。

运行演示

介绍如何编译并运行 cdc_com_port_dual 演示。

说明

关于用于运行该演示的 cdc_com_port_dual 演示应用程序的裸机 （非 RTOS）版本，请参见运行演示主题。

该演示应用程序包含 6 个任务。这些任务的说明如下：

- FrameworkTasks 任务是在 SYS_Tasks 函数中创建的。该任务会调用 USB 设备层任务函数（USB_DEVICE_Tasks）。该任务的优先级被设计为在与其他任务的优先级相比时最低。因此，该任务在所有其他任务处于阻塞状态或未准备好运行时运行。
- APP_USB_DEVICE_Open 任务是在 APP_Tasks 函数中创建的。该任务会创建应用程序所需的所有信号量和消息队列。它会创建 4 个任务，这 4 个任务实现应用程序逻辑。它会尝试打开设备层，然后基于 xSemaphoreBlockUsbConfigure 信号量发生阻塞。在 USB 设备层事件处理程序中，当主机配置了设备时，会给出 xSemaphoreBlockUsbConfigure。然后，该任务会恢复 4 个应用程序逻辑任务并挂起自身。
- APP_CDC1Read 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会调度对 CDC1 实例的读操作，然后基于 CDC1 实例 xSemaphoreCDCReadComplete 信号量发生阻塞。在 CDC 应用程序事件处理程序中，在遇到读完成事件时，会给出该信号量。然后，该任务会将它从主机处接收到的数据发布到 CDC 1 实例消息队列。
- APP_CDC2Read 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会调度对 CDC2 实例的读操作，然后基于 CDC2 实例 xSemaphoreCDCReadComplete 信号量发生阻塞。在 CDC 应用程序事件处理程序中，在遇到读完成事件时，会给出该信号量。然后，该任务会将它从主机处接收到的数据发布到 CDC 1 实例消息队列。
- APP_CDC1Write 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会基于 CDC 2 消息队列发生阻塞。当 APP_CDC2Read 任务将消息发布到该队列时， APP_CDC1Write 会准备好运行，并将数据 （从队列中接收）写入 CDC 1。然后，这些数据会被传输给主机。
- APP_CDC2Write 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会基于 CDC 1 消息队列发生阻塞。当 APP_CDC1Read 任务将消息发布到该队列时， APP_CDC2Write 会准备好运行，并将数据 （从队列中接收）写入 CDC 2。然后，这些数据会被传输给主机。

cdc_msd_basic

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了该演示应用程序的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_msd_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/freertos/cdc_msd_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_msd_basic.X`	`<install-dir>/apps/rtos/freertos/cdc_msd_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该配置可以在 PIC32 USB 入门工具包 II 上以中断模式和动态操作方式运行演示应用程序。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

如果连接的 USB 设备由总线供电，则 JP2 应就位。如果连接的 USB 设备是自供电的，则应移除它。

PIC32MZ 嵌入式连接 （EC）入门工具包

移除跳线 JP1。

运行演示

介绍如何编译并运行 cdc_msd_basic 演示。

说明

该 USB 主机演示应用程序说明如何运用所连接的复合 USB 设备上的 CDC 和 MSD 接口。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对器件编程。
3. 按照指示来设置并运行 cdc_serial_emulator_msd USB 设备演示。
4. 将 Explorer 16 开发板 （运行 cdc_serial_emulator_msd 演示）上的 UART （P1）端口通过市售串行转 USB 转换器连接到 USB 主机个人计算机。
5. 在 USB 主机个人计算机上启动终端程序，并选择串行转 USB 转换器作为通信端口。选择波特率 9600、无奇偶校验、1 个停止位和无流量控制。
6. 将 cdc_serial_emulator_msd 演示设置的 USB PICtail Plus 子板上的 mini-B 连接器连接到入门工具包上的 A 型 USB 主机连接器。
7. 终端仿真程序上会立即显示一个提示符 （DATA:）。
8. 输入少于 12 个字符的字符串，并按下 <Enter> 键。此处输入的字符串会被存储在 MSD 设备上名为 `file.txt` 的文件中。
9. 可以重复步骤 8。在提示符处输入的数据会被追加到该文件中。
10. 从主机上拔下 USB 设备，并将它连接到个人计算机主机，以检查海量存储设备的内容。其中应包含一个名为 `file.txt` 的文件，该文件应包含先前在终端程序提示符处输入的数据。

任务

该 USB 主机演示应用程序包含 4 个任务。这些任务的说明如下：

- FrameworkTasks 任务是在 SYS_Tasks 函数中创建的。该任务会调用 USB 主机层任务函数（USB_HOST_Tasks）。该任务的优先级被设计为在与其他任务的优先级相比时最低。因此，该任务在所有其他任务处于阻塞状态或未准备好运行时运行。
- APP_USB_HOST_Open 任务是在 APP_Tasks 函数中创建的。该任务会创建应用程序所需的所有信号量和消息队列。它会创建 2 个任务，这 2 个任务实现应用程序逻辑。它会尝试打开主机层，然后使能 USB 主机操作。然后，该任务会恢复 2 个应用程序逻辑任务并挂起自身。
- APP_USBHostCDCTask 任务是在 APP_USB_HOST_Open 任务中创建的。它会基于 xSemaphoreUSBCDCAttach 信号量发生阻塞。在应用程序 CDC 类驱动程序事件处理程序中，当 CDC 设备被枚举时，会给出该信号量。然后，该任务会打印一个提示符，并调度从所连接 USB 设备的 CDC 接口进行读取。当数据可用时，它会发布 xSemaphoreCDCReadComplete 信号量。
- APP_USBHostMSDTask 任务是在 APP_USB_HOST_Open 任务中创建的。它会基于 xSemaphoreUSBMSDAttach 信号量发生阻塞。在应用程序 MSD 事件处理程序中，当 MSD 设备被枚举时，会给出该信号量。然后，该任务会挂载所连接的存储介质，然后基于 xSemaphoreCDCReadComplete 信号量发生阻塞。当 CDC 主机接收到数据时， APP_USBHostCDCTask 任务会给出 xSemaphoreCDCReadComplete 信号量。然后， APP_USBHostMSDTask 任务会在所挂载的驱动器上打开一个文件，并在该文件中追加数据 （从 CDC 接收）。它会关闭该文件，然后再基于 xSemaphoreCDCReadComplete 信号量追加数据。

gfx

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 FreeRTOS 和 MPLAB Harmony 图形演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_freertos.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/freertos`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_freertos.X`	`<install-dir>/apps/rtos/freertos/gfx`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包和 MEB II 上运行。
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 FreeRTOS RTOS 与图形演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。LCD 屏幕颜色将每 500 ms 改变一次，并且用户 LED 将每 500 ms 翻转一次。

该应用程序中存在两个任务，并且该应用程序 / 系统中存在三个中断。

优先级最低的任务 LEDblinkTask 控制 LED 的翻转。优先级最高的任务 DisplayTask 会通过调用相应的 MPLAB Harmony 图形库函数来更改 LCD 屏幕的颜色。

RTOS 使用 Timer1 硬件中断来作为 RTOS 时钟节拍源。 DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。 RTOS 会调用软件中断 0，使用它来处理任务现场切换程序。

tcpip_client_server

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

TCP/IP 客户端服务器应用程序 tcpip_client_server 演示如何在 RTOS 环境中使用 TCP/IP 协议栈来运行多个 TCP 和 UDP 服务器和客户端。该演示还会运行 HTTP Web 服务器，使用非易失性存储器 （NVM） Microchip 专有文件系统 （MPFS）在内部 PIC32 闪存中存储网页。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 FreeRTOS 和 MPLAB Harmony TCP/IP 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `tcpip_client_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/freertos`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`tcpip_client_server.X`	`<install-dir>/apps/rtos/freertos/tcpip_client_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包和入门工具包 I/O 扩展板上运行。
pic32mx_eth_sk	pic32mx_eth_sk	该配置在 PIC32 以太网入门工具包和入门工具包 I/O 扩展板上运行。

配置硬件

介绍如何配置支持的硬件。

说明

连接到入门工具包 I/O 扩展板的 PIC32MZ EC 入门工具包

该配置不需要任何跳线设置。

该演示会大量使用 UART。为使用 UART 输出，需要将一个 RS-232 电平转换器连接到入门工具包 I/O 扩展板上的 UART2。从 J11 中，将 U2TX （48）连接到电平转换器 TX，将 U2RX （46）连接到电平转换器 RX。使用任意 +5V 和 GND 来构成完整的 RS-232 电平转换器线路。波特率为 115.2 kBaud， N81。

连接到入门工具包 I/O 扩展板的 PIC32 以太网入门工具包

该配置不需要任何跳线设置。

该演示会大量使用 UART。为使用 UART 输出，需要将一个 RS-232 电平转换器连接到入门工具包 I/O 扩展板上的 UART2。从 J11 中，将 U2TX （48）连接到电平转换器 TX，将 U2RX （46）连接到电平转换器 RX。使用任意 +5V 和 GND 来构成完整的 RS-232 电平转换器线路。波特率为 115.2 kBaud， N81。

运行演示

介绍如何编译并运行 FreeRTOS RTOS 与 TCP/IP 演示。

说明

要运行该演示应用程序，需要为电路板加电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。

该演示仅在以太网接口上运行 IPv4。要查看演示应用程序托管的网页，可以打开 Web 浏览器，并通过在地址栏中输入 URL （如 http://mchpboard_e）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

注：

1. TCP/IP 应用程序的 NetBIOS 名称在初始化 TCP/IP 协议栈时指定，它通常处于 tcpip_stack_init.c:: TCPIP_HOSTS_CONFIGURATION 结构的 hostName 成员中。必须使能 NetBIOS 服务， PIC32 演示才会响应 NetBIOS 查询 （默认情况下，会为该演示使能 NetBIOS 服务）。或者，也可以直接使用电路板的 IPv4，例如，http://192.168.1.131。
2. IPv4 地址可以通过在 PC 侧运行 TCP/IP Discovery 应用程序获取。这要求在编译协议栈时使能 TCP/IP Announce 模块 （默认情况下，该应用程序中会使能 Announce 模块）。

高级特性

要使用该演示的高级特性，必须为应用程序使能系统控制台。串行控制台是首选，默认情况下会为演示使能它。或者，也可以使用 MHC 将应用程序重新配置为使用 USB 控制台。

默认情况下，PIC32MZ 配置会使能 Telnet。Telnet 连接也可以用于传送应用程序需要的命令。在 PIC32MX 配置中，由于存储器资源有限， Telnet 模块是不使能的。

PIC32MZ 配置是运行该演示的首选配置，因为 PIC32MX 版本可能在运行时存储器不足。

TCP/IP 任务

该应用程序中存在 4 个 TCP/IP 任务，它们演示如何在多线程系统中使用 IPv4 TCP 和 UDP 套接字。其中每个任务实现 （并向系统命令处理器注册）特定的命令。这些命令用于打开相应的套接字，以及启动与远程主机的通信。在所有主机上，必须先打开服务器套接字，然后让客户端套接字与它们连接。此外，还有一些用于配置每个套接字 / 通信通道的命令。

遵循该应用程序中的模型，可以为任意任务添加额外的命令，用于实现所需的操作。

其中每个通信通道可以同时或依次运用。为了该演示的目的，应至少同时打开两个不同的通信通道，以反映多线程行为。

以下给出了一种常见情形，列出了运行该演示需要的控制台命令和步骤：

1. 启动 app1——TCP 服务器：
   - 通过发出控制台命令来启动在端口 9760 上监听的 PIC32 TCP 服务器套接字：`topen_s1<CR>`
   - 在客户端一侧 （PC 等）打开通过 TCP 使用端口 9760 连接 PIC32 服务器的客户端。使用任意网络工具 （netcat 等）或特殊应用程序和脚本 （TCL 和 Python 等）来发送和接收数据文件。
2. 启动 app2——TCP 客户端：
   - 在远程主机侧 （PC 等）打开在端口 9761 上监听传入连接的 TCP 服务器。使用任意网络工具 （netcat 等）或特殊应用程序和脚本 （TCL 和 Python 等）来接收 （和发送）数据文件。
   - 设置供服务器连接的 PIC32 客户端一侧的地址和端口，例如：`tsrv4_c1 192.168.100.101 9761<CR>`
   - 通过发出控制台命令启动 PIC32 TCP 客户端套接字：`topen_c1<CR>`
3. 启动 app3——UDP 服务器：
   - 通过发出控制台命令来启动在端口 32323 上监听的 PIC32 UDP 服务器套接字：`uopen_s1<CR>`
   - 在客户端一侧 （PC 等）打开通过 UDP 使用端口 32323 连接 PIC32 服务器的客户端。使用任意网络工具 （netcat 等）或特殊应用程序和脚本 （TCL 和 Python 等）来发送和接收数据文件。
4. 启动 app4——UDP 客户端：
   - 在远程主机一侧 （PC 等）打开在端口 32324 上监听传入连接的 UDP 服务器。使用任意网络工具 （netcat 等）或特殊应用程序和脚本 （TCL 和 Python 等）来接收 （和发送）数据文件。
   - 设置供服务器连接的 PIC32 客户端一侧的地址和端口，例如：`usrv4_c1 192.168.100.101 32324<CR>`
   - 通过发出控制台命令启动 PIC32 UDP 客户端套接字：`uopen_c1<CR>`
5. 现在， TCP/IP 任务已在运行，您可以在运行时检查进度情况。以下命令会给出接收和发送统计数据，显示每个任务传输的数据量：
   - `tstat_s1<CR>`
   - `tstat_c1<CR>`
   - `ustat_s1<CR>`
   - `ustat_c1<CR>`
6. 数据传输完成后，关闭 TCP/IP 套接字 （如果远程方尚未关闭）：
   - `tclose_s1<CR>`
   - `tclose_c1<CR>`
   - `uclose_s1<CR>`
   - `uclose_c1<CR>`

TCP/IP 任务说明和命令

app1.c::TCP 服务器

该任务使用默认情况下在端口 9760 上监听的 TCP 服务器套接字来实现简单的中继服务器。在该 TCP 端口上接收到的所有消息都会被中继回送到发起套接字。下表列出并说明了可用的命令

命令	说明
`topen_s1`	打开监听 TCP 服务器套接字。
`tmsg_s1`	使用该套接字向远程客户端发送短消息。
`tabort_s1`	向远程客户端发送中止 /RST 并停止通信。
`tclose_s1`	关闭套接字并停止通信。
`ttxsize_s1`	设置服务器套接字的发送缓冲区大小。缓冲区越大，套接字使用的存储器就越多，传输效率就越高。默认值为 2048 字节。
`trxsize_s1`	设置服务器套接字的接收缓冲区大小。缓冲区越大，套接字使用的存储器就越多，传输效率就越高。默认值为 2048 字节。
`tdisplay_s1`	使能显示系统控制台本地接收到的消息。

tstat_s1	显示 / 清除当前连接的当前发送和接收统计数据。

app2.c::TCP 客户端

该任务使用 TCP 客户端套接字来连接默认情况下在端口 9761 上监听的远程服务器。可以选择在本地显示通过套接字接收到的所有消息。客户端可能会向服务器发送消息。下表列出并说明了可用的命令。

命令	说明
topen_c1	打开 TCP 客户端套接字。
tmsg_c1	向远程服务器发送消息。
tabort_c1	向服务器发送中止 /RST 并停止通信。
tclose_c1	关闭套接字并停止通信。
tsrv4_c1	设置服务器 IPv4 地址和端口。默认值为 192.168.100.101 和 9761。
tasync_c1	使能向服务器连续发送消息。
tdisplay_c1	使能显示系统控制台本地接收到的消息。
tstat_c1	显示 / 清除当前连接的当前发送和接收统计数据。

app3.c::UDP 服务器

该任务使用默认情况下在端口 32323 上监听的 UDP 服务器套接字来实现简单的中继服务器。在该 UDP 端口上接收到的所有消息都会被中继回送到发起套接字。下表列出并说明了可用的命令。

命令	说明
uopen_s1	打开监听 UDP 服务器套接字。
uclose_s1	关闭套接字并停止通信。
ustnet_s1	选择 UDP 套接字的严格网络选项（允许来自任意网络的传入连接，或仅允许来自启动第一次连接的网络的传入连接）。
ustport_s1	选择 UDP 套接字的严格端口选项（允许来自任意主机端口的传入连接，或仅允许来自启动第一次连接时所用端口的传入连接）。
ustadd_s1	选择 UDP 套接字的严格地址选项（允许来自任意主机地址的传入连接，或仅允许来自启动第一次连接时所用地址的传入连接）。
udisplay_s1	使能显示系统控制台本地接收到的消息。
ustat_s1	显示 / 清除当前连接的当前发送和接收统计数据。
utxsize_s1	设置服务器套接字的发送缓冲区大小。缓冲区越大，套接字使用的存储器就越多，传输效率就越高。默认值为 1024 字节。请注意，对于以太网网络，该值不应大于 1460（以避免出现大于链路 MTU 的数据包）。

app4.c::UDP 客户端

该任务使用 UDP 客户端套接字来连接默认情况下在端口 32324 上监听的远程服务器。可以选择在本地显示通过套接字接收到的所有消息。客户端可能会向服务器发送消息。下表列出并说明了可用的命令。

命令	说明
uopen_c1	打开 UDP 客户端套接字。
umsg_c1	向远程服务器发送消息。
uclose_c	关闭套接字并停止通信。
usrv4_c1	设置服务器 IPv4 地址和端口。默认值为 192.168.100.101 和 32324。
uasync_c1	使能向服务器连续发送消息。
udisplay_c1	使能显示系统控制台本地接收到的消息。
ustat_c1	显示 / 清除当前连接的当前发送和接收统计数据。

utxsize_c1	设置服务器套接字的发送缓冲区大小。缓冲区越大，套接字使用的存储器就越多，传输效率就越高。默认值为 1024 字节。请注意，对于以太网网络，该值不应大于 1460（以避免出现大于链路 MTU 的数据包）。

Micrium uC_OS_II 演示

basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会使入门工具包上的用户 LED 发生闪烁，以显示 RTOS 线程正在运行并指示状态。

法律免责声明

该演示的源代码不是免费分发的。要获取相应的许可协议，请访问 Micriμm 网站：http://www.micrium.com。 Micriμm μC/OS-II 源代码必须安装在以下位置：`<install_dir>/third_party/rtos/MicriumOSII/Software`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Micriμm μC/OS-II 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `basic_ucos_II.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/uC_OS_II`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
basic_ucos_II.X	<install-dir>/apps/rtos/uC_OS_II/basic

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。
pic32mz_ec_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mz_da_sk	pic32mz_da_sk	该配置在 PIC32MZ DA 入门工具包上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 Micriμm μC/OS-II 基本演示。

说明

在该演示中，存在一个用户任务和一个硬件中断。用户任务 LEDBlinkTask 负责翻转用户 LED，以显示 RTOS 和应用程序已启动并正在运行。 RTOS 使用该硬件中断来运行 RTOS 时钟节拍。内部的内核定时器用作该硬件中断的中断源。

当演示启动并在运行时， LED 会每 500 ms 翻转一次。该演示将显示 RTOS 正在使用所选硬件运行。

Micrium uC/OS-III 演示

basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会使入门工具包上的用户 LED 发生闪烁，以显示 RTOS 线程正在运行并指示状态。

法律免责声明

该演示的源代码不是免费分发的。要获取相应的许可协议，请访问 Micriμm 网站：http://www.micrium.com。Micriμm μC/OS-III 源代码必须安装在以下位置：`<install_dir>/third_party/rtos/MicriumOSIII/Software`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Micriμm μC/OS-III 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `basic_ucos_III.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/uC_OS_III`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`basic_ucos_III.X`	`<install-dir>/apps/rtos/uC_OS_III/basic`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。
pic32mz_ec_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mz_da_sk	pic32mz_da_sk	该配置在 PIC32MZ DA 入门工具包上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 Micriμm μC/OS-III 基本演示。

说明

在该演示中，存在一个用户任务和一个硬件中断。用户任务 LEDBlinkTask 负责翻转用户 LED，以显示 RTOS 和应用程序已启动并正在运行。 RTOS 使用该硬件中断来运行 RTOS 时钟节拍。内部的内核定时器用作该硬件中断的中断源。

当演示启动并在运行时， LED 会每 500 ms 翻转一次。该演示将显示 RTOS 正在使用所选硬件运行。

gfx

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

法律免责声明

该演示的源代码不是免费分发的。要获取相应的许可协议，请访问 Micriμm 网站：http://www.micrium.com。 Micriμm μC/OS-III 源代码必须安装在以下位置：`<install_dir>/third_party/rtos/MicriumOSIII/Software`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Micriμm μC/OS-III 和 MPLAB Harmony 图形库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_ucosIII.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/uC_OS_III/gfx`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_ucosIII.X`	`<install-dir>/apps/rtos/uC_OS_III/gfx`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包和 MEB II 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 Micriμm μC/OS-III 和 MPLAB Harmony 图形库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。LCD 屏幕颜色将每 500 ms 改变一次，并且用户 LED 将每 500 ms 翻转一次。

该应用程序中存在两个任务，并且该应用程序 / 系统中存在三个中断。

优先级最低的任务 LEDblinkTask 控制 LED 的翻转。优先级最高的任务 DisplayTask 会通过调用相应的 MPLAB Harmony 图形库函数来更改 LCD 屏幕的颜色。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

gfx_usb

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

法律免责声明

该演示的源代码不是免费分发的。要获取相应的许可协议，请访问 Micriμm 网站：http://www.micrium.com。 Micriμm μC/OS-III 源代码必须安装在以下位置：`<install_dir>/third_party/rtos/MicriumOSIII/Software`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了MPLAB X IDE项目名称和位置，并且列出和说明了Micriμm μC/OS-III和MPLAB Harmony图形加USB库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_usb_ucos_III.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/uC_OS_III`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_usb_ucos_III.X`	`<install-dir>/apps/rtos/uC_OS_III/gfx_usb`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包和 MEB II 上运行。
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 Micriμm μC/OS-III 和 MPLAB Harmony 图形加 USB 库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。当 micro USB 电缆插入入门工具包时，将会仿真一个 USB 键盘。按下按钮 SW3 时，将会发送键击 a-z 和 1-9。 LCD 屏幕颜色将每 500 ms 改变一次，并且用户 LED （D3）将每 500 ms 翻转一次。

该应用程序 / 系统中存在四个任务和五个中断。

SystemUSBDeviceTask 是优先级最高的任务，负责获取由应用程序排入队列的已准备好通过调用相应 MPLAB Harmony USB 协议栈函数经由 USB 发送的所有 USB 数据。 ApplicationUSBDeviceTask 是优先级次高的任务。当检测到 SW3 按钮被按下时，ApplicationUSBDeviceTask 会仿真键盘按键按下。 ApplicationDisplayTask 会通过调用相应的 MPLAB Harmony 图形库函数来更改整个 LCD 屏幕的颜色。 ApplicationLEDblinkTask 是优先级最低的任务，它会每 500 ms 翻转用户 LED （D3）一次。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。 Timer2 硬件中断用于调用 MPLAB Harmony 定时器驱动程序函数。这是 MPLAB Harmony USB 协议栈的硬件要求。 DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。USB 通用事件中断用于调用相应的 MPLAB Harmony USB 驱动程序函数，该函数会获取由应用程序排入队列的所有 USB 数据，并实际将它写出到 USB 硬件上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

usb

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

法律免责声明

该演示的源代码不是免费分发的。要获取相应的许可协议，请访问 Micriμm 网站：http://www.micrium.com。 Micriμm μC/OS-III 源代码必须安装在以下位置：`<install_dir>/third_party/rtos/MicriumOSIII/Software`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了MPLAB X IDE项目名称和位置，并且列出和说明了Micriμm μC/OS-III和MPLAB Harmony图形加USB库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `usb_ucos_III.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/uC_OS_III/usb`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`usb_ucos_III.X`	`<install-dir>/apps/rtos/uC_OS_III/usb`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mx_usb_sk2	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 Micriμm μC/OS-III 和 MPLAB Harmony USB 库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。当 micro USB 电缆插入入门工具包时，将会仿真一个 USB 键盘。按下按钮 SW3 时，将会发送键击 a-z 和 1-9。用户 LED （D3）将每 500 ms 翻转一次。

该应用程序 / 系统中存在三个任务和四个中断。

SystemUSBDeviceTask 是优先级最高的任务，负责获取由应用程序排入队列的已准备好通过调用相应 MPLAB Harmony USB 协议栈函数经由 USB 发送的所有 USB 数据。 ApplicationUSBDeviceTask 是优先级次高的任务。当检测到 SW3 按钮被按下时，ApplicationUSBDeviceTask 会仿真键盘按键按下。 ApplicationLEDblinkTask 是优先级最低的任务，它会每 500 ms 翻转用户 LED （D3）一次。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。 Timer2 硬件中断用于调用 MPLAB Harmony 定时器驱动程序函数。USB 通用事件中断用于调用相应的 MPLAB Harmony USB 驱动程序函数，该函数会获取由应用程序排入队列的所有 USB 数据，并实际将它写出到 USB 硬件上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

OPENRTOS 演示

basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会使入门工具包上的 3 个用户 LED 发生闪烁，以显示 RTOS 线程正在运行并指示状态。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 OPENRTOS 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `basic_openrtos.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/openrtos/basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
basic_openrtos.X	<install-dir>/apps/rtos/openrtos/basic

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_da_sk	pic32mz_da_sk	该配置在 PIC32MZ DA 入门工具包上运行。
pic32mx_sk	pic32mx_usb_sk2、 pic32mx_usb_sk3、 pic32mx_eth_sk 和 pic32mx_eth_sk2	该配置在基于 PIC32MX 的入门工具包上运行：PIC32 以太网入门工具包、PIC32 以太网入门工具包 II、PIC32 USB 入门工具包 II 和 PIC32 USB 入门工具包 III。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包、 PIC32 以太网入门工具包 II、 PIC32 USB 入门工具包 II 和 PIC32 USB 入门工具包 III。

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 OPENRTOS 基本演示。

说明

当演示启动并在运行时， LED 会每 500 ms 翻转一次。该演示将显示 RTOS 正在使用所选硬件运行。该演示会使入门工具包上的 3 个用户 LED 发生闪烁，以显示 RTOS 线程正在运行并指示状态。

cdc_com_port_dual

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

这个基于 RTOS 的演示应用程序会创建在 USB 主机个人计算机上枚举为两个串行端口的 USB CDC 设备。该应用程序演示 MPLAB Harmony USB 协议栈在实时操作系统 （该示例使用 OPENRTOS）中工作以及支持相同设备类的多个实例的能力。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了该演示应用程序的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 cdc_com_port_dual.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/rtos/openrtos/cdc_com_port_dual。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
cdc_com_port_dual.X	<install-dir>/apps/rtos/openrtos/cdc_com_port_dual/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该配置可以在 PIC32 USB 入门工具包 II 上以中断模式和动态操作方式运行演示应用程序。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上以中断模式和动态操作方式运行演示应用程序。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该配置可以在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上以中断模式和动态操作方式运行演示应用程序。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

移除跳线 JP1。

运行演示

介绍如何编译并运行 cdc_com_port_dual 演示。

说明

关于用于运行该演示的 cdc_com_port_dual 演示应用程序的裸机 （非 RTOS）版本，请参见运行演示主题。

该演示应用程序包含 6 个任务。这些任务的说明如下：

- FrameworkTasks 任务是在 SYS_Tasks 函数中创建的。该任务会调用 USB 设备层任务函数（USB_DEVICE_Tasks）。该任务的优先级被设计为在与其他任务的优先级相比时最低。因此，该任务在所有其他任务处于阻塞状态或未准备好运行时运行。
- APP_USB_DEVICE_Open 任务是在 APP_Tasks 函数中创建的。该任务会创建应用程序所需的所有信号量和消息队列。它会创建 4 个任务，这 4 个任务实现应用程序逻辑。它会尝试打开设备层，然后基于 xSemaphoreBlockUsbConfigure 信号量发生阻塞。在 USB 设备层事件处理程序中，当主机配置了设备时，会给出 xSemaphoreBlockUsbConfigure。然后，该任务会恢复 4 个应用程序逻辑任务并挂起自身。
- APP_CDC1Read 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会调度对 CDC1 实例的读操作，然后基于 CDC1 实例 xSemaphoreCDCReadComplete 信号量发生阻塞。在 CDC 应用程序事件处理程序中，在遇到读完成事件时，会给出该信号量。然后，该任务会将它从主机处接收到的数据发布到 CDC 1 实例消息队列。
- APP_CDC2Read 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会调度对 CDC2 实例的读操作，然后基于 CDC2 实例 xSemaphoreCDCReadComplete 信号量发生阻塞。在 CDC 应用程序事件处理程序中，在遇到读完成事件时，会给出该信号量。然后，该任务会将它从主机处接收到的数据发布到 CDC 1 实例消息队列。

- APP_CDC1Write 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会基于 CDC 2 消息队列发生阻塞。当 APP_CDC2Read 任务将消息发布到该队列时， APP_CDC1Write 会准备好运行，并将数据 （从队列中接收）写入 CDC 1。然后，这些数据会被传输给主机。
- APP_CDC2Write 任务是在 APP_USB_DEVICE_Open 任务中创建的。它会基于 CDC 1 消息队列发生阻塞。当 APP_CDC1Read 任务将消息发布到该队列时， APP_CDC2Write 会准备好运行，并将数据 （从队列中接收）写入 CDC 2。然后，这些数据会被传输给主机。

cdc_msd_basic

演示 RTOS 应用程序中对于复合 USB 设备的主机支持。

说明

该演示应用程序会创建一个演示复合 USB 设备操作的 USB 主机应用程序。该主机应用程序会枚举所连接的复合设备上的 CDC 和 MSD 接口，然后在一个应用程序中操作它们。该演示应用程序使用 RTOS 来创建负责管理主机应用程序的 CDC 和 MSD 方面的线程。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了该演示应用程序的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_msd_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/openrtos/cdc_msd_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_msd_basic.X`	`<install-dir>/apps/rtos/openrtos/cdc_msd_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该配置可以在 PIC32 USB 入门工具包 II 上以中断模式和动态操作方式运行演示应用程序。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

如果连接的 USB 设备由总线供电，则 JP2 应就位。如果连接的 USB 设备是自供电的，则应移除它。

运行演示

介绍如何编译并运行 cdc_msd_basic 演示。

说明

该 USB 主机演示应用程序说明如何运用所连接的复合 USB 设备上的 CDC 和 MSD 接口。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对设备编程。
3. 按照指示来设置并运行 cdc_serial_emulator_msd USB 设备演示。
4. 将 Explorer 16 开发板 （运行 cdc_serial_emulator_msd 演示）上的 UART （P1）端口通过市售串行转 USB 转换器连接到 USB 主机个人计算机。
5. 在 USB 主机个人计算机上启动终端程序，并选择串行转 USB 转换器作为通信端口。选择波特率 9600、无奇偶校验、1 个停止位和无流量控制。
6. 将 cdc_serial_emulator_msd 演示设置的 USB PICtail Plus 子板上的 mini-B 连接器连接到入门工具包上的 A 型 USB 主机连接器。
7. 终端仿真程序上会立即显示一个提示符 （DATA:）。
8. 输入少于 12 个字符的字符串，并按下 <Enter> 键。此处输入的字符串会被存储在 MSD 设备上名为 `file.txt` 的文件中。
9. 可以重复步骤 8。在提示符处输入的数据会被追加到该文件中。
10. 从主机上拔下 USB 设备，并将它连接到个人计算机主机，以检查海量存储设备的内容。其中应包含一个名为 `file.txt` 的文件，该文件应包含先前在终端程序提示符处输入的数据。

任务

该 USB 主机演示应用程序包含 4 个任务。这些任务的说明如下：

- FrameworkTasks 任务是在 SYS_Tasks 函数中创建的。该任务会调用 USB 主机层任务函数（USB_HOST_Tasks）。该任务的优先级被设计为在与其他任务的优先级相比时最低。因此，该任务在所有其他任务处于阻塞状态或未准备好运行时运行。
- APP_USB_HOST_Open 任务是在 APP_Tasks 函数中创建的。该任务会创建应用程序所需的所有信号量和消息队列。它会创建 2 个任务，这 2 个任务实现应用程序逻辑。它会尝试打开主机层，然后使能 USB 主机操作。然后，该任务会恢复 2 个应用程序逻辑任务并挂起自身。
- APP_USBHostCDCTask 任务是在 APP_USB_HOST_Open 任务中创建的。它会基于 xSemaphoreUSBCDCAttach 信号量发生阻塞。在应用程序 CDC 类驱动程序事件处理程序中，当 CDC 设备被枚举时，会给出该信号量。然后，该任务会打印一个提示符，并调度从所连接 USB 设备的 CDC 接口进行读取。当数据可用时，它会发布 xSemaphoreCDCReadComplete 信号量。
- APP_USBHostMSDTask 任务是在 APP_USB_HOST_Open 任务中创建的。它会基于 xSemaphoreUSBMSDAttach 信号量发生阻塞。在应用程序 MSD 事件处理程序中，当 MSD 设备被枚举时，会给出该信号量。然后，该任务会挂载所连接的存储介质，然后基于 xSemaphoreCDCReadComplete 信号量发生阻塞。当 CDC 主机接收到数据时， APP_USBHostCDCTask 任务会给出 xSemaphoreCDCReadComplete 信号量。然后， APP_USBHostMSDTask 任务会在所挂载的驱动器上打开一个文件，并在该文件中追加数据 （从 CDC 接收）。它会关闭该文件，然后再基于 xSemaphoreCDCReadComplete 信号量追加数据。

gfx

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 OPENRTOS 和 MPLAB Harmony 图形库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_openrtos.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/openrtos`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_openrtos.X`	`<install-dir>/apps/rtos/openrtos/gfx`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

连接到 MEB II 的 PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

连接到 MEB 的 PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 OPENRTOS 和 MPLAB Harmony 图形库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。LCD 屏幕颜色将每 500 ms 改变一次，并且用户 LED 将每 500 ms 翻转一次。

该应用程序中存在两个任务，并且该应用程序 / 系统中存在三个中断。

优先级最低的任务 LEDblinkTask 控制 LED 的翻转。优先级最高的任务 DisplayTask 会通过调用相应的 MPLAB Harmony 图形库函数来更改 LCD 屏幕的颜色。

RTOS 使用 Timer1 硬件中断来作为 RTOS 时钟节拍源。 DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。 RTOS 会调用软件中断 0，使用它来处理任务现场切换程序。

SEGGER embOS 演示

basic

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

该演示会使入门工具包上的 3 个用户 LED 发生闪烁，以显示 RTOS 线程正在运行并指示状态。

法律免责声明

该 SEGGER embOS RTOS 演示的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 SEGGER embOS 网站：https://www.segger.com/license-models.html。 SEGGER embOS 源代码必须安装在以下位置：`<install-dir>/third_party/rtos/embOS`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SEGGER embOS 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 basic_embos.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/rtos/embos/basic。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
basic_embos.X	<install-dir>/apps/rtos/embos/basic/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。
pic32mz_ec_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mz_da_sk	pic32mz_da_sk	该配置在 PIC32MZ DA 入门工具包上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SEGGER embOS 基本演示。

说明

在该演示中，存在一个用户任务和一个硬件中断。用户任务负责翻转用户 LED，以显示 RTOS 和应用程序已启动并正在运行。 RTOS 使用该硬件中断来运行 RTOS 时钟节拍。内部的内核定时器用作该硬件中断的中断源。

当演示启动并在运行时， LED 会每 500 ms 翻转一次。该演示将显示 RTOS 正在使用所选硬件运行。

gfx

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

法律免责声明

该 SEGGER embOS RTOS 演示的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 SEGGER embOS 网站：https://www.segger.com/license-models.html。 SEGGER embOS 源代码必须安装在以下位置：`<install-dir>/third_party/rtos/embOS`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SEGGER embOS 和 MPLAB Harmony 图形库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_embos.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/embos/gfx`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_embos.X`	`<install-dir>/apps/rtos/embos/gfx/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包和 MEB II 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SEGGER embOS 和 MPLAB Harmony 图形库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。LCD 屏幕颜色将每 500 ms 改变一次，并且用户 LED 将每 500 ms 翻转一次。

该应用程序中存在一个任务，并且该应用程序 / 系统中存在三个中断。

用户任务控制用户 LED 的翻转，并通过调用相应的 MPLAB Harmony 图形库函数来更改 LCD 屏幕颜色。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

gfx_usb

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

法律免责声明

该 SEGGER embOS RTOS 演示的源代码不是免费分发的。要获取源代码和相应的许可协议，请访问 SEGGER embOS 网站：https://www.segger.com/license-models.html。 SEGGER embOS 源代码必须安装在以下位置：`<install-dir>/third_party/rtos/embOS`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了MPLAB X IDE项目名称和位置，并且列出和说明了SEGGER embOS和MPLAB Harmony图形加USB库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `gfx_usb_embos.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/embos/gfx_usb`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`gfx_usb_embos.X`	`<install-dir>/apps/rtos/embos/gfx_usb/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	该配置在 PIC32MZ EC 入门工具包和 MEB II 上运行。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	该配置在 PIC32MZ EF 入门工具包和 MEB II 上运行。
pic32mx_usb_sk2_meb	pic32mx_usb_sk2+meb	该配置在 PIC32 USB 入门工具包 II 和 MEB 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包和 MEB II

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II 和 MEB

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SEGGER embOS 和 MPLAB Harmony 图形加 USB 库。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。当 micro USB 电缆插入入门工具包时，将会仿真一个 USB 键盘。按下按钮SW3时，将会发送键击a-z和1-9。LCD屏幕颜色将每500 ms改变一次，并且用户LED（MEB II上的D3）将每500 ms翻转一次。

该应用程序 / 系统中存在四个任务和五个中断。

SystemUSBDeviceTask 是优先级最高的任务，负责获取由应用程序排入队列的已准备好通过调用相应 MPLAB Harmony USB 协议栈函数经由 USB 发送的所有 USB 数据。 ApplicationUSBDeviceTask 是优先级次高的任务。当检测到 SW3 按钮被按下时，ApplicationUSBDeviceTask 会仿真键盘按键按下。ApplicationDisplayTask 会通过调用相应的 MPLAB Harmony 图形库函数来更改整个 LCD 屏幕的颜色。 ApplicationLEDblinkTask 是优先级最低的任务，它会每 500 ms 翻转一次用户 LED （MEB II 上的 D3）。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。 Timer2 硬件中断用于调用 MPLAB Harmony 定时器驱动程序函数。这是 MPLAB Harmony USB 协议栈的硬件要求。 DMA 通道 1 中断用于调用相应的 MPLAB Harmony 图形驱动程序函数，该函数会获取由应用程序排入队列的所有图形工作，并实际将它写出到 LCD 显示屏上。USB 通用事件中断用于调用相应的 MPLAB Harmony USB 驱动程序函数，该函数会获取由应用程序排入队列的所有 USB 数据，并实际将它写出到 USB 硬件上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

usb

本部分介绍受支持的演示板，说明了如何配置硬件 （如需要）以及如何运行演示。

说明

法律免责声明

该 SEGGER embOS RTOS 演示的源代码不是免费分发的。要获取源代码和相应的

许可协议，请访问 SEGGER embOS 网站：https://www.segger.com/license-models.html。 SEGGER embOS 源代码必须安装在以下位置：`<install-dir>/third_party/rtos/embOS`，以便适用的 MPLAB Harmony 演示可以工作。

编译应用程序

本部分指明了MPLAB X IDE项目名称和位置，并且列出和说明了SEGGER embOS和MPLAB Harmony图形加USB库演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `usb_embos.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/rtos/embos/usb`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
usb_embos.X	<install-dir>/apps/rtos/embos/usb/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk	pic32mz_ec_sk	该配置在 PIC32MZ EC 入门工具包上运行。
pic32mz_ef_sk	pic32mz_ef_sk	该配置在 PIC32MZ EF 入门工具包上运行。
pic32mx_usb_sk2	pic32mx_usb_sk2	该配置在 PIC32 USB 入门工具包 II 上运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 II

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 SEGGER embOS 和 MPLAB Harmony USB 库演示。

说明

为电路板供电，使用 MPLAB X IDE 打开演示，然后编译并装入演示。当 micro USB 电缆插入入门工具包时，将会仿真一个 USB 键盘。按下按钮 SW3 时，将会发送键击 a-z 和 1-9。用户 LED 将每 500 ms 翻转一次。

该应用程序 / 系统中存在三个任务和四个中断。

SystemUSBDeviceTask 是优先级最高的任务，负责获取由应用程序排入队列的已准备好通过调用相应 MPLAB Harmony USB 协议栈函数经由 USB 发送的所有 USB 数据。 ApplicationUSBDeviceTask 是优先级次高的任务。当检测到 SW3 按钮被按下时，ApplicationUSBDeviceTask 会仿真键盘按键按下。 ApplicationLEDblinkTask 是优先级最低的任务，它会每 500 ms 翻转一次用户 LED。

RTOS 使用 CPU 内核定时器硬件中断来作为 RTOS 时钟节拍源。 Timer2 硬件中断用于调用 MPLAB Harmony 定时器驱动程序函数。USB 通用事件中断用于调用相应的 MPLAB Harmony USB 驱动程序函数，该函数会获取由应用程序排入队列的所有 USB 数据，并实际将它写出到 USB 硬件上。 RTOS 会调用 SYSCALL 通用异常，使用它来处理任务现场切换程序。

TCP/IP 演示

本部分介绍 TCP/IP 演示。

简介

TCP/IP 和 Wi-Fi® 演示应用程序帮助

说明

TCP/IP 演示

本部分介绍 Microchip 的 TCP/IP 演示项目，包含了关于演示硬件兼容性的信息，并且还介绍了如何配置和运行演示。

Wi-Fi 演示

该分发包包含一系列基于 Wi-Fi 的固件项目，用于演示在 PIC32 器件上运行的 MPLAB Harmony Wi-Fi 服务和 TCP/IP 协议栈的功能。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

关于演示支持信息，请参见 Wi-Fi 演示配置矩阵。

Wi-Fi 演示配置矩阵

提供 Wi-Fi 演示支持信息。

说明

以下矩阵提供关于 Wi-Fi 演示的信息。

Wi-Fi Demonstation Matrix	Demonstration Projects					
	pic32_wifi_web_server	pic32_eth_wifi_web_server	wifi_easy_configuration	wifi_g_demo	wifi_wolfssl_tcp_client	wifi_wolfssl_tcp_server
Wi-Fi PICtail Module						
MRF24WG0MA Wi-Fi G PICTail/PICtail Plus Daughter Board	Yes	Yes	Yes	Yes	Yes	Yes
MRF24WN0MA Wi-Fi PICtail/PICtail Plus Daughter Board	Yes	Yes	Yes	N/A	N/A	N/A
BSP						
Wi-Fi G Demo Board	N/A	N/A	N/A	Yes	N/A	N/A
chipKIT WF32	Yes	N/A	N/A	N/A	N/A	N/A
PIC32MX795F512L PIM with Explorer 16 Development Board	Yes	N/A	N/A	N/A	N/A	N/A
PIC32MX Ethernet Starter Kit	N/A	Yes	Yes	N/A	Yes	Yes
PIC32MZ EC Starter Kit	N/A	Yes	Yes	N/A	Yes	Yes
PIC32MZ EC Starter Kit with MEB II	N/A	Yes	Yes	N/A	N/A	N/A
PIC32MZ EF Starter Kit	N/A	Yes	Yes	N/A	N/A	N/A
PIC32MZ EF Starter Kit with MEB II	N/A	Yes	Yes	N/A	N/A	N/A
Project Configuration						
chipkit_wf32	Yes	N/A	N/A	N/A	N/A	N/A
pic32mx795_pim+e16	Yes	N/A	Yes	N/A	N/A	N/A
pic32mx795_pim+e16+11n+freertos	Yes	N/A	Yes	N/A	N/A	N/A
pic32mx795_pim+e16+freertos	Yes	N/A	Yes	N/A	N/A	N/A
pic32mx_eth_sk+ioexp	N/A	Yes	Yes	N/A	Yes	Yes
pic32mx_eth_sk+ioexp+11n+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mx_eth_sk+ioexp+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ec_sk+ioexp	N/A	Yes	Yes	N/A	Yes	Yes
pic32mz_ec_sk+ioexp+11n+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ec_sk+ioexp+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ec_sk+meb2	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ec_sk+meb2+11n+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ec_sk+meb2+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ef_sk+ioexp	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ef_sk+ioexp+11n+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ef_sk+ioexp+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ef_sk+meb2	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ef_sk+meb2+11n+freertos	N/A	Yes	Yes	N/A	N/A	N/A
pic32mz_ef_sk+meb2+freertos	N/A	Yes	Yes	N/A	N/A	N/A
wifi_g_db	N/A	N/A	N/A	Yes	N/A	N/A
Console Command Interface						
UART Console	Yes	Optional	Optional	N/A	Optional	Optional
USB CDC Console	N/A	Yes	Yes	N/A	Yes	Yes
RTOS						
Non-RTOS	Yes	Yes	Yes	Yes	Yes	Yes
FreeRTOS	Yes	Yes	Yes	N/A	N/A	N/A
Device Configuration						
PIC32MX695F512L	Yes	N/A	N/A	N/A	N/A	N/A
PIC32MX695F512H	N/A	N/A	N/A	Yes	N/A	N/A
PIC32MX795F512L	Yes	Yes	Yes	N/A	Yes	Yes
PIC32MZ2048ECH144	N/A	Yes	Yes	N/A	Yes	Yes
PIC32MZ2048EFM144	N/A	Yes	Yes	N/A	N/A	N/A
Duo-Port Ethernet and Wi-Fi						
PIC32MX Ethernet Starter Kit	N/A	Yes	Yes	N/A	N/A	N/A
PIC32MZ EC Start Kit	N/A	Yes	Yes	N/A	N/A	N/A
PIC32MZ EC Starter Kit with MEB II - rework on MEB II board is Required	N/A	Yes	Yes	N/A	N/A	N/A
PIC32MZ EF Start Kit	N/A	Yes	Yes	N/A	N/A	N/A
PIC32MZ EF Starter Kit MEB II - rework on MEB II board is Required	N/A	Yes	Yes	N/A	N/A	N/A
Single-Port Wi-Fi or Wi-Fi Only						
PIC32MX706F512L PIM with Explorer 16 Development Board	Yes	N/A	Yes	N/A	N/A	N/A
Starter Kit I/O Expansion						
Add on-board UART/Serial 9-pin Connector (see Configuring the Hardware for details)	N/A	Optional	Optional	N/A	Optional	Optional
On-board Jumper: J10/pin 56 to J10/pin 37 Required for MRF24WG PICtail (see Configuring the Hardware for details)	N/A	Required	Required	N/A	Required	Required
On-board Jumper: J10/pin 12 to J10/pin 8 Required for MRF24WN PICtail (see Configuring the Hardware for details)	N/A	Required	Required	N/A	N/A	N/A
168-pin to 132-pin Starter Kit Adapter for PIC32MZ EC or PIC32MZ EF Starter Kit	N/A	Required	Required	N/A	N/A	N/A

Wi-Fi 控制台命令

本部分介绍可用于 Wi-Fi Web 服务器和 EasyConfig 演示的演示支持命令。

说明

注： 请参见以下文档，它们可以从 Microchip 网站下载：

- Microchip MRF24W Getting Started Guide for MRF24WB0MA/A, MRF24WG0MA/B for MLA v5 （DS52108）
- Microchip Wi-Fi® G Demo Board User's Guide （DS50002147）

请确保访问网站获取文档的最新版本。

Web 服务器和 EasyConfig 演示都支持 Wi-FI 控制台命令，这使用户可以通过 Wi-Fi 设置进行控制。rftest 除外，以下所有命令都可用于 MRF24WG 和 MRF24WN Wi-Fi 驱动程序。 rftest 命令只能用于 MRF24WN Wi-Fi 驱动程序。

命令：eraseconf

参数	说明
无。	擦除在存储器中保存的 Wi-Fi 配置的 Wi-Fi 控制台命令。

命令：iwconfig

参数	说明
[ssid <name>]	name：指定 SSID 的名称 （1 至 32 个 ASCII 个字符）。
[mode <idle | managed>]	idle：从当前配置断开。 managed：在基础结构模式下连接当前设置的 SSID。
[power <enable | disable>]	enable：使能所有节能功能 （PS_POLL）。将会通过唤醒来检查所有类型的通信 （单播、多播和广播）。 disable：禁止所有节能功能。将总是处于活动供电状态。

[security <mode>]	mode：open/wep40/wep104/wpa/wpa2/pin/pbc。例如：
	`iwconfig security open`
	`iwconfig security wep40 <key>`
	`iwconfig security wep104 <key>`
	`iwconfig security wpa <key>`
	`iwconfig security wpa2 <key>`
	`iwconfig security pin <pin>`
	`iwconfig security pbc`

[scan]	启动 Wi-Fi 扫描。
[scanget <scan_index>]	scan_index：在完成扫描之后取回扫描结果 （1 - n）。

命令：mac

参数	说明
无。	取回 MRF24WN 模块的 MAC 地址的 Wi-Fi 控制台命令。

命令：readconf

参数	说明
无。	读取在存储器中保存的 Wi-Fi 配置的 Wi-Fi 控制台命令。

命令：rftest *（仅用于 MRF24WN Wi-Fi 驱动程序）*

参数	说明
[rate < packet_count packet_size channel header_type >]	执行监管测试的 Wi-Fi 控制台命令。 rate （以 Mbps 为单位）： • 0 = 1 • 1 = 2 • 2 = 5.5 • 3 = 11 • 4 = 6 • 5 = 9 • 6 = 12 • 7 = 18 • 8 = 24 • 9 = 36 • 10 = 48 • 11 = 54 • 12 = 6.5 • 13 = 13 • 14 = 19.5 • 15 = 26 • 16 = 39 • 17 = 52 • 18 = 58.5 packet_count：发送数量 （1 至 14）。 packet_size：有效载荷大小 （0 至 1400）。 channel：1 至 14。 header_type：0—— 信标帧； 1——QoS 数据帧； 2 至 4—— 地址数据帧。

命令：saveconf

参数	说明
无。	将 Wi-Fi 配置保存到存储器中的 Wi-Fi 控制台命令。

演示

TCP/IP 协议栈库演示应用程序的说明。

说明

PHY 驱动程序支持

所有属于分发包一部分且使用 Microchip 参考开发板的 PIC32MX 和 PIC32MZ 项目都预先配置了特定的 PHY 驱动程序。在开发板支持不同 PHY 子板时，可以更改默认 PHY。要对特定开发板使用不同的 PHY，必须完成以下操作：

1. 使用 MHC，将项目配置为使用正确的 PHY，并确保对于特定 PHY 子板使用正确的 PHY 地址和配置标志。 PHY 板的 MII/RMII 和 I/O 配置标志应与项目配置熔丝位匹配。
2. 重新生成项目，并确保为您正在使用的配置选择新的 PHY 驱动程序。

或者，也可以手动设置您的项目，如下：

- 项目应选择对应于选定配置所用 PHY 子板 （即 LAN8720、 LAN8740 和 LAN9303 等）的 PHY 驱动程序
- 修改 TCPIP_EMAC_PHY_ADDRESS 和 TCPIP_EMAC_PHY_CONFIG_FLAGS，以使用正确的 PHY 地址 （例如，两个 SMSC PHY 子板的 PHY 地址通常为零）和配置标志 （MII/RMII 和 I/O 引脚配置等）
- 或直接更新 tcpip_stack_init.c::tcpipMACPIC32INTInitData 结构，以使用正确的 PHY 地址和配置标志

- 确保正确选择配置熔丝位，使之与您使用的硬件和 PHY 板匹配
- 重新编译项目

berkeley_tcp_client

该配置演示如何创建互联网客户端，该客户端使用 Berkeley API 来创建到 Web 服务器的 TCP/IP 连接。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Berkeley TCP 客户端演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `berkeley_tcp_client.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/berkeley_tcp_client`。

警告

在使用 Microchip Harmony 配置器 （Microchip Harmony Configurator， MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`berkeley_tcp_client.X`	`<install-dir>/apps/tcpip/berkeley_tcp_client/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 Berkeley TCP 客户端。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 Berkeley TCP 客户端。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 Berkeley TCP 客户端。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 Berkeley TCP 客户端。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

使用 RJ45 连接器在路由器或交换机和 PIC32 以太网入门工具包之间建立连接。确保路由器或交换机连接到互联网。

该演示中只有一个可通过串行端口使用的命令：

`openurl <url>`——<url> 参数必须是完整形式的 URL ；例如， http://www.microchip.com/

在输入这一个命令之后，演示会进行 DNS 查询，然后打开到所请求 URL 的连接，并执行简单的 HTTP PUT 命令。结果会被发送到串行端口。

berkeley_tcp_server

该配置演示如何创建互联网服务器，该服务器使用 Berkeley API 在端口 9764 上创建 TCP/IP 回送服务器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Berkeley TCP 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `berkeley_tcp_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/berkeley_tcp_server`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`berkeley_tcp_server.X`	`<install-dir>/apps/tcpip/berkeley_tcp_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 Berkeley TCP 服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 Berkeley TCP 服务器。

pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 Berkeley TCP 服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 Berkeley TCP 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

该演示不会通过串行端口提供任何附加的功能；但是，可以检查当前的 IP。在通过 DHCP 向演示分配有效 IP 之后，它会立即准备好在 9760 上接受 TCP/IP 连接。该演示会回送它在连接上接收到的所有数据。

可以将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。这会在 USB 总线上创建一个 USB CDC 设备。为了与软件进行通信，通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。

berkeley_udp_client

该配置演示如何创建互联网客户端，该客户端使用 Berkeley API 来创建到指定端口的 UDP/IP 连接。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Berkeley UDP 客户端演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `berkeley_udp_client.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/berkeley_udp_client`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
berkeley_udp_client.X	<install-dir>/apps/tcpip/berkeley_udp_client/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 Berkeley UDP 客户端。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 Berkeley UDP 客户端。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 Berkeley UDP 客户端。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 Berkeley UDP 客户端。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

可以通过控制台使用 3 个顺序命令：

setudppacketoptions <hostname> <port> <message>—— 该命令指定将 UDP 数据包发送到何处，以及消息中的内容

getudppacketoptions—— 该命令会显示当前选项

sendudppacket—— 该命令会发送一个 UDP 数据包

在输入 sendudppacket 命令之后，该演示会通过进行 DNS 查询来查找主机名，并向该主机发送一个 UDP 数据包。

UDP 服务器将在通过命令 setudppacketoptions 配置的 UDP 端口上接收到输出消息。

berkeley_udp_relay

该应用程序演示如何使用多个套接字来同时进行发送和接收。该应用程序具有 3 个不同的子功能：

- UDP 中继，它在一个套接字上接受 UDP 数据包，并在不同套接字上发送数据包
- UDP 中继客户端，它会产生与 UDP 中继服务器兼容的 UDP 通信数据
- UDP 中继服务器，它会接收并通过检查通信数据来对数据包进行计数，并报告是否有任何数据包被丢弃

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Berkeley UDP 中继演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `berkeley_udp_relay.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/berkeley_udp_relay`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`berkeley_udp_relay.X`	`<install-dir>/apps/tcpip/berkeley_udp_relay/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 Berkeley UDP 中继。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 Berkeley UDP 中继。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 Berkeley UDP 中继。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 Berkeley UDP 中继。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。

在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

此外，使用 RJ45 连接器在路由器或交换机和 PIC32 以太网入门工具包之间建立连接。确保路由器或交换机连接到互联网。

该应用程序演示简单的 UDP 数据包中继。此外，还提供了在同一设备上生成数据包和接收数据包的功能。 IPv6 未经过测试。

演示命令

该演示中具有几个可通过控制台端口使用的不同命令：

常规应用程序命令：

- current—— 显示当前配置
- start—— 启动数据包中继服务
- stop—— 停止数据包中继服务
- reportinterval <seconds>—— 设置向控制台进行报告之间的时间间隔

中继服务配置：

- relayhost <host name>—— 设置要将数据包中继到的主机
- relayport <port number>—— 设置要将数据包中继到的端口
- ipv4port <port number>—— 设置中继服务器监听要进行中继的数据包的 IPv4 端口
- ipv6port <port number>—— 设置中继服务器监听要进行中继的数据包的 IPv6 端口

中继客户端配置和命令：

- relayclienthost <host name>—— 设置要将数据包发送到的主机
- relayclientport <port number>—— 设置要将数据包发送到的端口
- relayclientiter <number>—— 要生成的数据包数量
- relayclientstart—— 启动中继客户端。该命令必须在常规应用程序启动后使用。在调用 start 之后，并且在中继或中继服务器接收到第一个数据包之后，会向控制台发送定期更新，其中包含关于已接收到的数据包数量和字节数量的信息。

berkeley_udp_server

该配置演示如何创建互联网服务器，该服务器使用 Berkeley API 在端口 9764 上创建 UDP/IP 回送服务器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Berkeley UDP 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `berkeley_udp_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/berkeley_udp_server`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`berkeley_udp_server.X`	`<install-dir>/apps/tcpip/berkeley_udp_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 Berkeley UDP 服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 Berkeley UDP 服务器。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 Berkeley UDP 服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 Berkeley UDP 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

该演示不会通过串行端口提供任何附加的功能；但是，可以检查当前的 IP。在通过 DHCP 向演示分配有效 IP 之后，它会立即准备好在 9760 上接受 UDP/IP 连接。该演示会回送它在连接上接收到的所有数据。

可以将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。这会在 USB 总线上创建一个 USB CDC 设备。为了与软件进行通信，通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。

snmpv3_nvm_mpfs

SNMPv3 NVM MPFS 演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SNMPv3 NVM MPFS 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `snmpv2_nvm_mpfs.X` 项目，然后选择所需的配置。

非易失性存储器（NVM）Microchip专有文件系统（MPFS）将snmp.bib文件和其他网页文件存储在内部闪存中，并通过MPFS API访问。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/snmpv3_nvm_mpfs`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`snmpv3_nvm_mpfs.X`	`<install-dir>/apps/tcpip/snmpv3_nvm_mpfs/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上，在中断模式和动态操作下演示 SNMPv3 NVM MPFS。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上，在中断模式和动态操作下演示 SNMPv3 NVM MPFS。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上，在中断模式和动态操作下演示 SNMPv3 NVM MPFS。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 使用所用工具包中提供的 USB 电缆将入门工具包板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。
3. 使用工具包中提供的以太网接插线，将入门工具包电路板上的 RJ-45 以太网端口连接到网络集线器或开发计算机上的以太网端口。

4. 编译、下载演示项目并在目标板上运行。
5. 演示应用程序托管了一个 SNMP 和 SNMPv3 服务器。
6. 通过运行 tcpip_discoverer 来获得电路板的 IPv4 和 IPv6 地址。
7. 打开 SNMP 管理器 （建议使用 iREASONING SNMP 管理器）并配置 IPv4 或 IPv6 地址。

注： 关于使用 iREASONING SNMP 管理器来使用和配置应用程序的完整详细信息，请参见第三方帮助中的 iREASONING Networks MIB Browser （iREASONING 网络 MIB 浏览器）部分。

8. 将 USB 电缆连接到 PIC32 以太网入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。您可以使用控制台界面观察 IP 地址详细信息和查询协议栈。

SNMP MIB 浏览器

有几种 SNMP MIB 浏览器可供使用。用户也可以安装特定于其应用程序的定制 MIB 浏览器。

SNMP Get、 GetNext、 GetBulk、 Set 请求和响应按 SNMP v1/v2/v3 版本的预期方式工作。

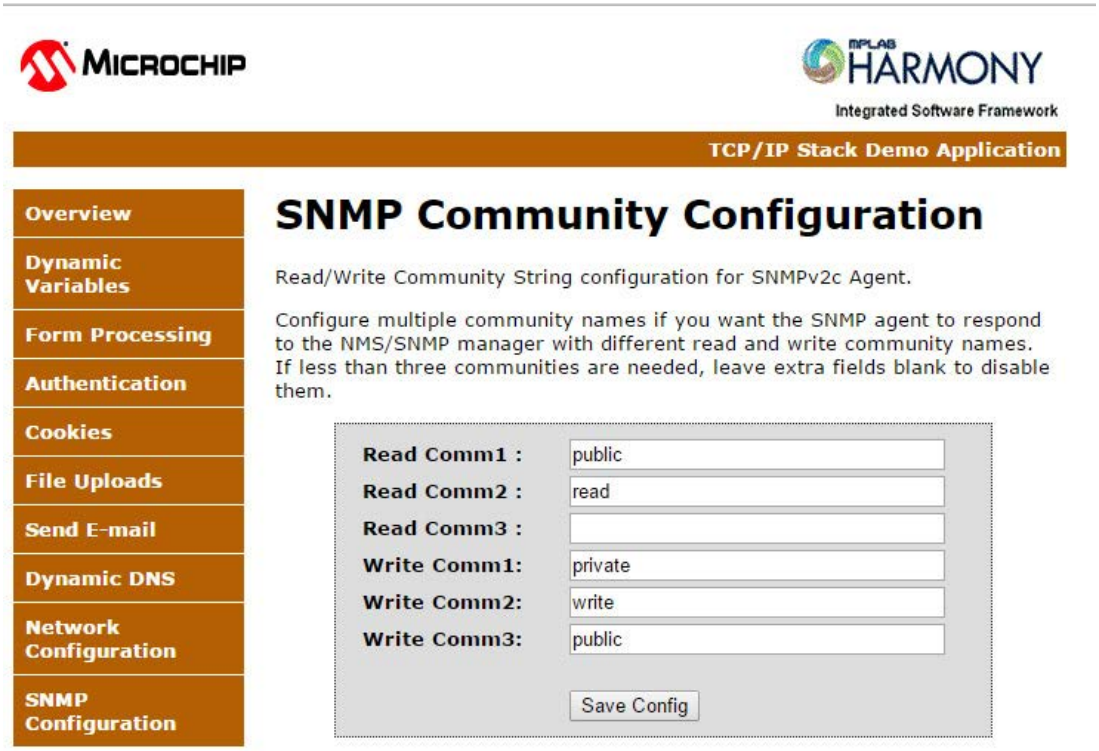
- 对于 SNMP v2c，代理配置为具有 3 个读共同体（“public”、“read”和“ ”）和 3 个写共同体（“private”、“write”和“public”）。
- 对于 SNMP v3，代理按照下表配置：

类型	用户 1	用户 2	用户 3
USM 用户	microchip	SnmpAdmin	root
安全级别	auth 和 priv	auth 和 no priv	no auth 和 no priv
认证算法	MD5	SHA1	N/A
认证密码	auth12345	ChandlerUS	N/A
隐私算法	AES	N/A	N/A
隐私密码	priv12345	N/A	N/A

Microchip SNMP 协议栈支持陷阱版本 1 和陷阱版本 2。该演示的陷阱输出为多 varbind SNMPv3 陷阱版本 2。用户可能需要根据选择的 SNMP 浏览器来配置陷阱接收器。

SNMP v2c 共同体的 HTTP 配置

可以通过 SNMP Configuration （SNMP 配置）网页动态地配置读共同体名和写共同体名。使用 http://mchpboard_e/mpfsupload或 http://< 电路板 IP 地址 >（对于 IPv6，它应为 http://<Ipv6 地址 >:80/index.html）访问网页，然后通过导航栏访问 SNMP Configuration 网页。使用 “admin”作为用户名，“microchip”作为密码。



警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

snmpv3_sdcard_fatfs

SNMPv3 SD 卡 FAT 文件系统演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 SNMPv3 SD 卡 FAT FS 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `snmpv3_sdcard_fatfs.X` 项目，然后选择所需的配置。

SD 卡 FAT FS 将 snmp.bib 文件和其他网页存储在外部 SD 卡中，并通过 FAT FS API 访问。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/snmpv3_sdcard_fatfs`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`snmpv3_sdcard_fatfs.X`	`<install-dir>/apps/tcpip/snmpv3_sdcard_fatfs/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32_eth_sk	pic32mx_eth_sk	演示在使用入门工具包 I/O 扩展板与用于 SD 和 MMC 卡的 PICtail 子板的 PIC32 以太网入门工具包上，通过 FAT 文件系统访问 microSD 卡上的 SNMP 文件。该演示在中断模式和动态操作下运行。
pic32mz_ec_sk	pic32mz_ec_sk+meb2	演示在 PIC32MZ EC 入门工具包和 MEB II 板组合上，通过 FAT 文件系统访问 microSD 卡上的 SNMP 文件。该演示在中断模式和动态操作下运行。
pic32mz_ef_sk	pic32mz_ef_sk+meb2	演示在 PIC32MZ EF 入门工具包和 MEB II 板组合上，通过 FAT 文件系统访问 microSD 卡上的 SNMP 文件。该演示在中断模式和动态操作下运行。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

1. 将入门工具包连接到 MEB II 上的应用电路板连接器。
2. 确保 microSD 卡已格式化，并已装入 `snmp.bib` 文件和在 `<install-dir>/apps/tcpip/web_server_sdcard_fatfs/firmware/src/web_pages` 文件夹中提供的网页。
3. 将含有网页的 microSD 卡插入 MEB II 上的 microSD 卡插槽 （J8）。

PIC32 以太网入门工具包与入门工具包扩展板

1. 将 PIC32 以太网入门工具包连接到 I/O 扩展板。
2. 确保 SD 卡已格式化，并已装入 `snmp.bib` 文件和在 `<install-dir>apps/tcpip/snmpv3_sdcard_fatfs/firmware/src/web_pages` 文件夹中提供的网页。
3. 将含有 snmp.bib 文件和网页的 SD 卡插入用于 SD 和 MMC 卡的 PICtail 子板，然后将该子板插入 PIC32 I/O 扩展板的 SPI1 插槽（J4—— 插槽从 1 开始计数）。

注： PICtail 子板上的 SD 卡应正对 PIC32 以太网入门工具包。

运行演示

本部分介绍如何编译并运行演示。

说明

请参见 snmpv3_nvm_mpfs 配置的运行演示部分，因为该过程与该配置相同。

注： 确保按配置硬件所述方式插入含有 snmp.bib 文件和网页的 SD 卡。

tcpip_tcp_client

该配置演示如何创建互联网客户端，该客户端使用 MPLAB Harmony TCP API 来创建到 Web 服务器的 TCP/IP 连接。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP TCP 客户端演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `tcpip_tcp_client.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/tcpip_tcp_client`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`tcpip_tcp_client.X`	`<install-dir>/apps/tcpip/tcpip_tcp_client/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 TCP/IP TCP 客户端。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 TCP/IP TCP 客户端。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 TCP/IP TCP 客户端。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 TCP/IP TCP 客户端。
pic32mx_eth_sk2_enc28j60	pic32mx_eth_sk2	在连接到 10 Mbps 以太网 PICtail Plus 子板和入门工具包 I/O 扩展板的 PIC32 以太网入门工具包 II 上，使用 ENC28J60 驱动程序库来演示 TCP/IP TCP 客户端。
pic32mx_eth_sk2_encx24j600	pic32mx_eth_sk2	在连接到快速 100 Mbps 以太网 PICtail Plus 子板和入门工具包 I/O 扩展板的 PIC32 以太网入门工具包 II 上，使用 ENCx24J600 驱动程序库来演示 TCP/IP TCP 客户端。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

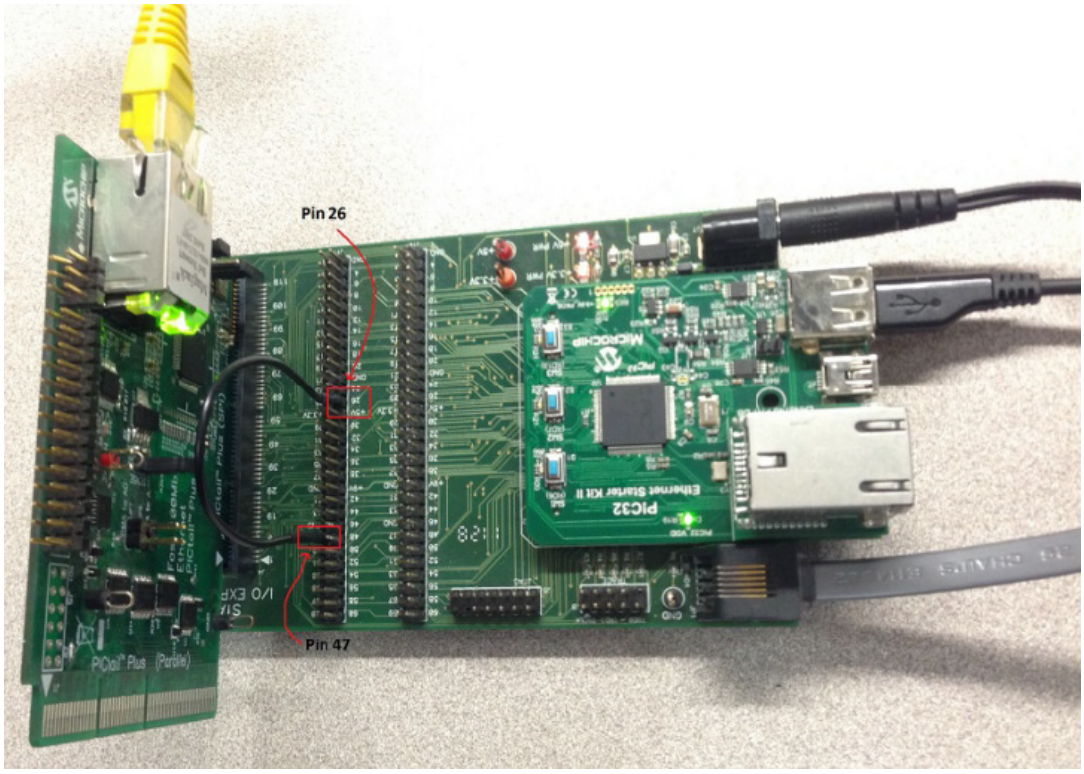
不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II、快速 100 Mbps 以太网 PICtail Plus 子板和入门工具包 I/O 扩展板

将快速 100 Mbps 以太网 PICtail Plus 子板连接到入门工具包 I/O 扩展板板上的 J4，方法是将 J2 PICtail Plus （SPI）卡的边缘滑入连接器顶部，从而使两个电路板上的白色箭头对齐。插入 PICtail 子板，以便它使用 SPI1。J11 上的引脚 26 和 47 需要通过跳线连接，使 CS 线由 PIC32 控制。 PIC32 以太网入门工具包 II 连接到入门工具包 I/O 扩展板上的 J1。更多详细信息请参见下图。



PIC32 以太网入门工具包 II、以太网 PICtail Plus 子板和入门工具包 I/O 扩展板

将 10 Mbps 以太网 PICtail Plus 子板连接到入门工具包 I/O 扩展板板上的 J4，方法是将 J2 PICtail Plus （SPI）卡的边缘滑入连接器顶部。插入 PICtail 子板，以便它使用 SPI1。J11 上的引脚 26 和 47 需要通过跳线连接，使 CS 线由 PIC32 控制。PIC32 以太网入门工具包 II 连接到入门工具包 I/O 扩展板上的 J1。

参考前面的图示，使用该配置的以太网 PICtail Plus 子板替换快速 100 Mbps 以太网 PICtail Plus 子板。 PICtail 的 RJ45 应正对入门工具包 I/O 扩展板的边缘连接器。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

该演示中只有一个可通过串行端口使用的命令：

`openurl <url>`——<url> 参数必须是完整形式的 URL ；例如， http://www.microchip.com/

在输入这一个命令之后，演示会进行 DNS 查询，然后打开到所请求 URL 的连接，并执行简单的 HTTP PUT 命令。结果会被发送到串行端口。

tcpip_tcp_client_server

该配置演示如何创建使用 MPLAB Harmony TCP API 的互联网客户端和互联网服务器。该演示说明了 TCP/IP 环回是如何工作的，并且它是 TCP/IP 客户端和 TCP/IP 服务器应用程序的组合。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP TCP 客户端服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `tcpip_tcp_client_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为`<install-dir>/apps/tcpip/tcpip_tcp_client_server`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`tcpip_tcp_client_server.X`	`<install-dir>/apps/tcpip/tcpip_tcp_client_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 TCP/IP TCP 客户端服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 TCP/IP TCP 客户端服务器。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 TCP/IP TCP 客户端服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 TCP/IP TCP 客户端服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

该演示中只有一个可通过串行端口使用的命令：

`openurl <url>`——<url> 参数必须是完整形式的 URL ； 例如， http://www.microchip.com/

在输入这一个命令之后，演示会进行 DNS 查询，然后打开到所请求 URL 的连接，并执行简单的 HTTP PUT 命令。结果会被发送到串行端口。

tcpip_tcp_server

该配置演示如何创建互联网服务器，该服务器使用 MPLAB Harmony TCP API 在端口 9764 上创建 TCP/IP 回送服务器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP TCP 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `tcpip_tcp_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/tcpip_tcp_server`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`tcpip_tcp_server.X`	`<install-dir>/apps/tcpip/tcpip_tcp_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 TCP/IP TCP 服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 TCP/IP TCP 服务器。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 TCP/IP TCP 服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 TCP/IP TCP 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

该演示不会通过串行端口提供任何附加的功能；但是，可以检查当前的 IP。在通过 DHCP 向演示分配有效 IP 之后，它会立即准备好在 9764 上接受 TCP/IP 连接。该演示会回送它在连接上接收到的所有数据。

可以将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。这会在 USB 总线上创建一个 USB CDC 设备。为了与软件进行通信，通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。

tcpip_udp_client

该配置演示如何创建互联网客户端，该客户端使用 MPLAB Harmony UDP API 来创建到指定端口的 UDP/IP 连接。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP UDP 客户端演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `tcpip_udp_client.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/tcpip_udp_client`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`tcpip_udp_client.X`	`<install-dir>/apps/tcpip/tcpip_udp_client/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 TCP/IP UDP 客户端。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 TCP/IP UDP 客户端。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 TCP/IP UDP 客户端。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 TCP/IP UDP 客户端。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

可以通过控制台使用 3 个命令：

- `setudppacketoptions <hostname> <port> <message>`—— 该命令指定将 UDP 数据包发送到何处，以及消息中的内容
- `getudppacketoptions`—— 该命令会显示当前选项
- `sendudppacket`—— 该命令会发送一个 UDP 数据包

在输入 sendudppacket 命令之后，该演示会通过进行 DNS 查询来查找主机名，并向该主机发送一个 UDP 数据包。

tcpip_udp_client_server

该配置演示如何创建使用 MPLAB Harmony UDP API 的互联网客户端和互联网服务器。该演示说明了 UDP/IP 环回是如何工作的，并且它是 TCP/IP UDP 客户端和 TCP/IP UDP 服务器应用程序的组合。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP UDP 客户端服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `tcpip_udp_client_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为<install-dir>/apps/tcpip/tcpip_udp_client_server。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 system_init.c 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
tcpip_udp_client_server.X	<install-dir>/apps/tcpip/tcpip_udp_client_server/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 TCP/IP UDP 客户端服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 TCP/IP UDP 客户端服务器。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 TCP/IP UDP 客户端服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 TCP/IP UDP 客户端服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

可以通过控制台使用 3 个命令：

- setudppacketoptions <hostname> <port> <message>—— 该命令指定将 UDP 数据包发送到何处，以及消息中的内容

- `getudppacketoptions`—— 该命令会显示当前选项
- `sendudppacket`—— 该命令会发送一个 UDP 数据包

在输入 sendudppacket 命令之后，该演示会通过进行 DNS 查询来查找主机名，并向该主机发送一个 UDP 数据包。

tcpip_udp_server

该配置演示如何创建互联网服务器，该服务器使用 MPLAB Harmony UDP API 在端口 9764 上创建 UDP/IP 回送服务器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP UDP 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `tcpip_udp_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/tcpip_udp_server`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`tcpip_udp_server.X`	`<install-dir>/apps/tcpip/tcpip_udp_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 TCP/IP UDP 服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 TCP/IP UDP 服务器。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 TCP/IP UDP 服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 TCP/IP UDP 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

该演示不会通过串行端口提供任何附加的功能；但是，可以检查当前的 IP。在通过 DHCP 向演示分配有效 IP 之后，它会立即准备好在 9764 上接受 UDP/IP 连接。该演示会回送它在连接上接收到的所有数据。

可以将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。这会在 USB 总线上创建一个 USB CDC 设备。为了与软件进行通信，通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。

web_net_server_nvm_mpfs

Web 网络服务器非易失性存储器 （NVM） MPFS TCP/IP 演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP Web 网络服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `pic32_eth_web_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/web_net_server_nvm_mpfs`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`pic32_eth_web_server.X`	`<install-dir>/apps/tcpip/web_net_server_nvm_mpfs/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。

pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

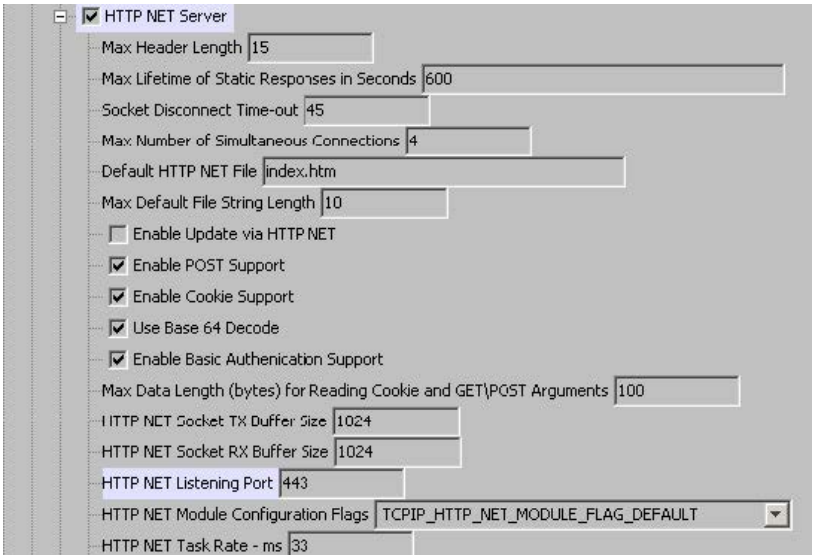
配置 MHC

提供关于演示的 MHC 配置的信息。

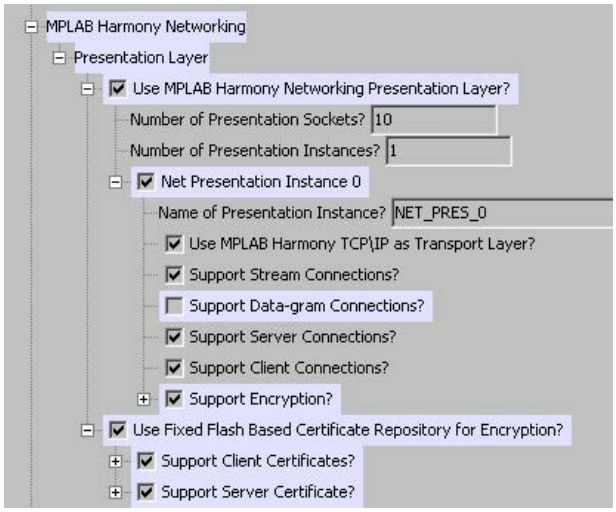
说明

MHC 配置：

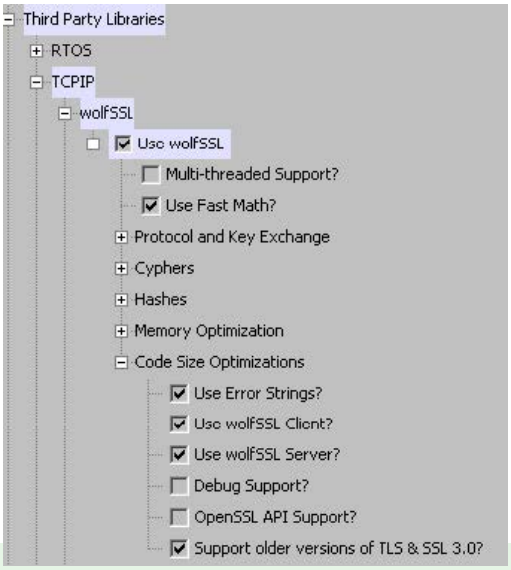
1. 从 *Harmony Framework Configuration* > *TCPIP Stack （TCPIP 协议栈）*中选择 **HTTP NET Server** （HTTP 网络服务器）。
2. 保留默认设置。 HTTP 服务器监听端口应已设置为 443，用于进行加密连接，如下图所示；但是，如果需要未加密的连接，则将该值更改为 80：



3. 使能 MPLAB Harmony 联网表示层，如下所示：
   - 确保使用 TCP/IP 协议栈作为传输层
   - 选择 **Support Stream Connections?** （支持流连接？）（用于支持 TCP）
   - 选择 **Support Server Connections?** （支持服务器连接？）（用于支持 HTTPS）
   - 选择 **Support Client Connections?** （支持客户端连接？）（用于支持加密 SMTP）
   - 选择 **Support Client Certificate?** （支持客户端证书？）和 **Support Server Certificate?** （支持服务器证书？）（根据需要）



4. 使能 *Third Party Libraries （第三方库）> TCPIP > wolfSSL > Use wolfSSL （使用 wolfSSL）*。确保根据 HTTP 和 SMTP 选择使能 wolfSSL 客户端和 wolfSSL 服务器，如下图所示：



运行演示

本部分介绍如何编译并运行 PIC32 以太网 Web 网络服务器演示。

说明

要查看演示应用程序托管的网页，可以打开 Web 浏览器，并通过在地址栏中输入 URL （如 http://mchpboard_e）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

注：

1. TCP/IP 应用程序的 NetBIOS 名称在初始化 TCP/IP 协议栈时指定，它通常处于 tcpip_stack_init.c:: TCPIP_HOSTS_CONFIGURATION 结构的 hostName 成员中。必须使能 NetBIOS 服务， PIC32 演示才会响应 NetBIOS 查询。或者，也可以直接使用电路板的 IPv4 或 IPv6 地址 （如果使能了 IPv6），例如，http://192.168.1.131 或 http://[fdfe:dcba:9876:1:204:a3ff:fe12:128e]。
2. IPv4 和 IPv6 地址可以通过在 PC 侧运行 TCP/IP Discovery 应用程序获取。这要求在编译协议栈时使能 TCP/IP Announce 模块。

演示过程

请使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 使用所用工具包中提供的 USB 电缆将入门工具包板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。

3. 使用工具包中提供的以太网接插线，将入门工具包电路板上的 RJ-45 以太网端口连接到网络集线器或开发计算机上的以太网端口。
4. 编译、下载演示项目并在目标板上运行。
5. 演示应用程序托管了一个 HTTP 服务器。打开 Web 浏览器，并通过在地址栏中输入 URL（如 http://mchpboard_e）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

该演示应用程序具有以下功能：

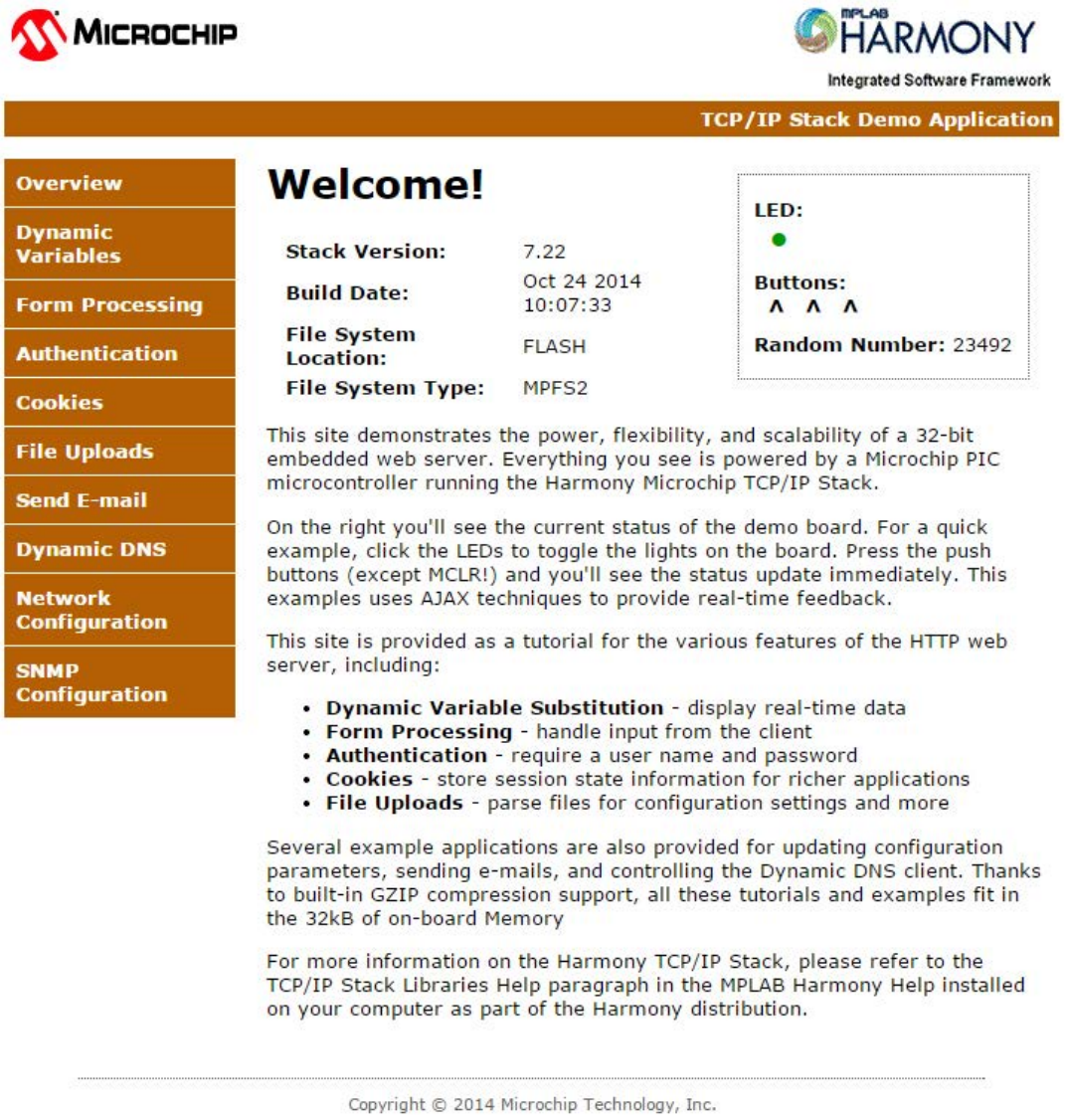
- *Real-time Hardware Control and Dynamic Variables*—— 在 Overview 页面中，可以通过单击 LED 来翻转 PIC32 以太网入门工具包电路板上的 LED。按下 PIC32 以太网入门工具包电路板上的按钮，可以使网页上的按钮发生翻转。动态变量可以在 HTTP 服务器上实时更新。

注： 由于与不同受支持硬件上的 GPIO 和以太网引脚上的功能复用相关的问题，演示的 LED 功能部分会受到一定限制。

- *Form Processing*—— 可以通过使用 GET 和 POST 方法来处理来自客户端的输入（该功能控制板载 LED，只能在 Explorer 16 开发板上工作）。
- *Authentication*—— 给出一个常用的受限访问功能的示例。
- *Cookies*—— 给出一个在客户端存储小文本字符串的示例。
- *File Uploads*—— 给出一个使用 POST 方法进行文件上传的示例。HTTP 服务器可以接受用户定义的 MPFS/MPFS2 映像文件，使用它来供应网页。
- *Send E-mail*—— 说明简单的 SMTP POST 方法。
- *Dynamic DNS*—— 运用动态 DNS 功能。
- *Network Configuration*—— 可以在 Network Configuration 页面中查看 PIC32 以太网入门工具包电路板的 MAC 地址、主机名和 IP 地址，并且可以更新一些配置。
- *MPFS Upload* （MPFS 上传） —— 可以使用该功能将一组新的网页上传到 Web 服务器，该功能可以通过 http://mchpboard_e/mpfsupload 访问。

注：

1. MPFS 映像的位置固定在闪存页的起始处 （与页边界对齐）。在该演示中， MPFS 上传的大小限制为 64K ； 可以增大该大小，方法是将 NVM_MEDIA_SIZE 更改为所需大小 （受可用大小限制）并使用以下链接器命令改写 EBASE 地址：
   - --defsym=_ebase_address=0x9D0xxxx （其中， xxxx = 9D000000+NVM_MEDIA_SIZE）
2. 在编译项目时，必须使能 MPFS 上传功能。



web_server_nvm_mpfs

Web 服务器非易失性存储器 （NVM） MPFS TCP/IP 演示。

pic32_eth_web_server

本部分介绍开始使用 PIC32 以太网 Web 服务器演示应用程序所需要的步骤。

说明

该演示说明如何运用在 PIC32 器件上运行的 HTTP Web 服务器。非易失性存储器（NVM）Microchip 专有文件系统（MPFS）Web 服务器演示将网页存储在内部闪存中，并通过 MPFS API 访问。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP Web 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `pic32_eth_web_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/web_server_nvm_mpfs`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`pic32_eth_web_server.X`	`<install-dir>/apps/tcpip/web_server_nvm_mpfs/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示通过 Microchip 专有文件系统在内部闪存上托管的 Web 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行 PIC32 以太网 Web 服务器演示。

说明

要查看演示应用程序托管的网页，可以打开 Web 浏览器，并通过在地址栏中输入 URL （如 http://mchpboard_e）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

注：

1. TCP/IP 应用程序的 NetBIOS 名称在初始化 TCP/IP 协议栈时指定，它通常处于 tcpip_stack_init.c:: TCPIP_HOSTS_CONFIGURATION 结构的 hostName 成员中。必须使能 NetBIOS 服务， PIC32 演示才会响应 NetBIOS 查询。或者，也可以直接使用电路板的 IPv4 或 IPv6 地址 （如果使能了 IPv6），例如，http://192.168.1.131 或 http://[fdfe:dcba:9876:1:204:a3ff:fe12:128e]。
2. IPv4 和 IPv6 地址可以通过在 PC 侧运行 TCP/IP Discovery 应用程序获取。这要求在编译协议栈时使能 TCP/IP Announce 模块。

演示过程

请使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 使用所用工具包中提供的 USB 电缆将入门工具包板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。
3. 使用工具包中提供的以太网接插线，将入门工具包电路板上的 RJ-45 以太网端口连接到网络集线器或开发计算机上的以太网端口。
4. 编译、下载演示项目并在目标板上运行。
5. 演示应用程序托管了一个 HTTP 服务器。打开 Web 浏览器，并通过在地址栏中输入 URL（如 http://mchpboard_e）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

该演示应用程序具有以下功能：

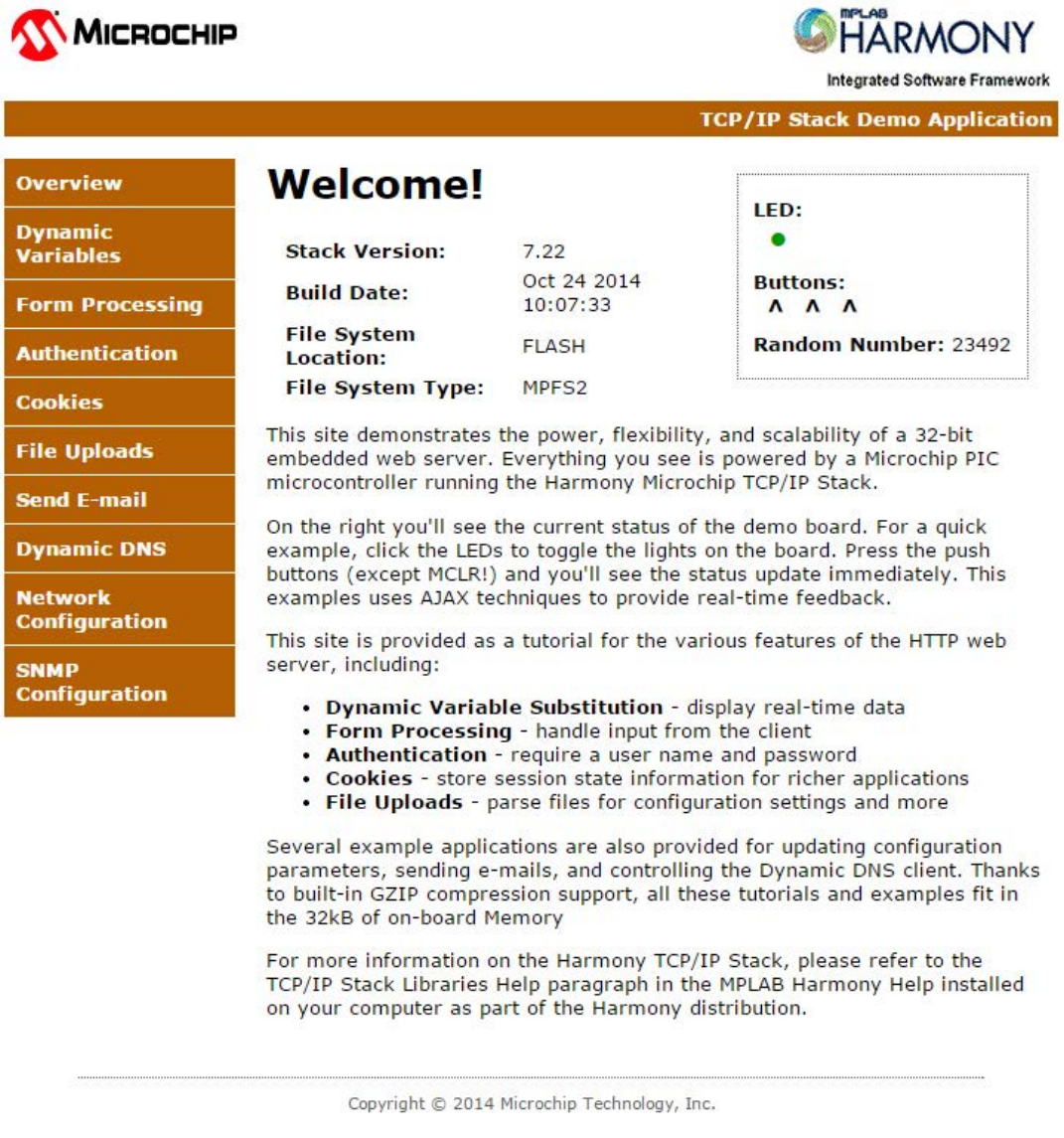
- *Real-time Hardware Control and Dynamic Variables*—— 在 Overview 页面中，可以通过单击 LED 来翻转 PIC32 以太网入门工具包电路板上的 LED。按下 PIC32 以太网入门工具包电路板上的按钮，可以使网页上的按钮发生翻转。动态变量可以在 HTTP 服务器上实时更新。

注： 由于与不同受支持硬件上的 GPIO 和以太网引脚上的功能复用相关的问题，演示的 LED 功能部分会受到一定限制。

- *Form Processing*—— 可以通过使用 GET 和 POST 方法来处理来自客户端的输入（该功能控制板载 LED，只能在 Explorer 16 开发板上工作）。
- *Authentication*—— 给出一个常用的受限访问功能的示例。
- *Cookies*—— 给出一个在客户端存储小文本字符串的示例。
- *File Uploads*—— 给出一个使用 POST 方法进行文件上传的示例。HTTP 服务器可以接受用户定义的 MPFS/MPFS2 映像文件，使用它来供应网页。
- *Send E-mail*—— 说明简单的 SMTP POST 方法。
- *Dynamic DNS*—— 运用动态 DNS 功能。
- *Network Configuration*—— 可以在 Network Configuration 页面中查看 PIC32 以太网入门工具包电路板的 MAC 地址、主机名和 IP 地址，并且可以更新一些配置。
- *MPFS Upload*—— 可以使用该功能将一组新的网页上传到 Web 服务器，该功能可以通过 http://mchpboard_e/mpfsupload 访问。

注：

1. MPFS 映像的位置固定在闪存页的起始处 （与页边界对齐）。在该演示中， MPFS 上传的大小限制为 64K ；可以增大该大小，方法是将 NVM_MEDIA_SIZE 更改为所需大小 （受可用大小限制）并使用以下链接器命令改写 EBASE 地址：
   - --defsym=_ebase_address=0x9D0xxxx （其中， xxxx = 9D000000+NVM_MEDIA_SIZE）
2. 在编译项目时，必须使能 MPFS 上传功能。



pic32_eth_wifi_web_server

本部分介绍开始使用 PIC32 以太网 Wi-Fi Web 服务器演示应用程序所需要的步骤。

说明

Wi-Fi Web 服务器演示 （`apps\tcpip\web_server_nvm_mpfs\firmware`）说明如何运用在 PIC32 器件上运行的 HTTP Web 服务器。非易失性存储器（NVM）Microchip 专有文件系统（MPFS）Web 服务器演示将网页存储在内部闪存中，并通过 MPFS API 访问。

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

注：

1. 由于硬件限制，以太网和 Wi-Fi Web 服务器演示 （pic32_eth_wifi_web_server）中的 pic32mz_ec_sk+meb2 和 pic32mz_ef_sk+meb2 配置仅通过 Wi-Fi 来运用 Web 服务器。配置硬件主题中介绍了变通方法。
2. 对于 pic32mz_ec_sk+meb2 或 pic32mz_ef_sk+meb2 配置，可以安装 MRF24WG Wi-Fi 模块和 / 或 MRF24WN Wi-Fi PICtail，但只有一个 Wi-Fi 模块会处于工作状态，这取决于 MHC 选择的 MRF24W。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 PIC32 以太网 Wi-Fi Web 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `pic32_eth_wifi_web_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/web_server_nvm_mpfs`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`pic32_eth_wifi_web_server.X`	`<install-dir>/apps/tcpip/web_server_nvm_mpfs/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk+ioexp	pic32mx_eth_sk	在连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32 以太网入门工具包上运行的演示。
pic32mz_ec_sk+ioexp	pic32mz_ec_sk	在连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的演示。
pic32mz_ec_sk+meb2	pic32mz_ec_sk+meb2	在连接到多媒体扩展板 II（MEB II）与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的演示。
pic32mx_eth_sk+ioexp+freertos	pic32mx_eth_sk	在连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32 以太网入门工具包上运行的 FreeRTOS 版本的演示。
pic32mx_eth_sk+ioexp+11n+freertos	pic32mx_eth_sk	在连接到入门工具包 I/O 扩展板与 MRF24WN PICtail 子板的 PIC32 以太网入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ec_sk+ioexp+freertos	pic32mz_ec_sk	在连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ec_sk+ioexp+11n+freertos	pic32mz_ec_sk	在连接到入门工具包 I/O 扩展板与 MRF24WN PICtail 子板的 PIC32MZ EC 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ec_sk+meb2+freertos	pic32mz_ec_sk+meb2	在连接到多媒体扩展板 II（MEB II）与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ec_sk+meb2+11n+freertos	pic32mz_ec_sk+meb2	在连接到多媒体扩展板 II（MEB II）与 MRF24WN PICtail 子板的 PIC32MZ EC 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ef_sk+ioexp	pic32mz_ef_sk	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的演示。

pic32mz_ef_sk+ioexp+freertos	pic32mz_ef_sk	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ef_sk+meb2	pic32mz_ef_sk+meb2	在连接到 MEB II 与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的演示。
pic32mz_ef_sk+meb2+freertos	pic32mz_ef_sk+meb2	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ef_sk+ioexp+11n+freertos	pic32mz_ef_sk	在连接到 I/O 扩展板与 MRF24WN PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ef_sk+meb2+11n+freertos	pic32mz_ef_sk	在连接到 MEB II 与 MRF24WN PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。

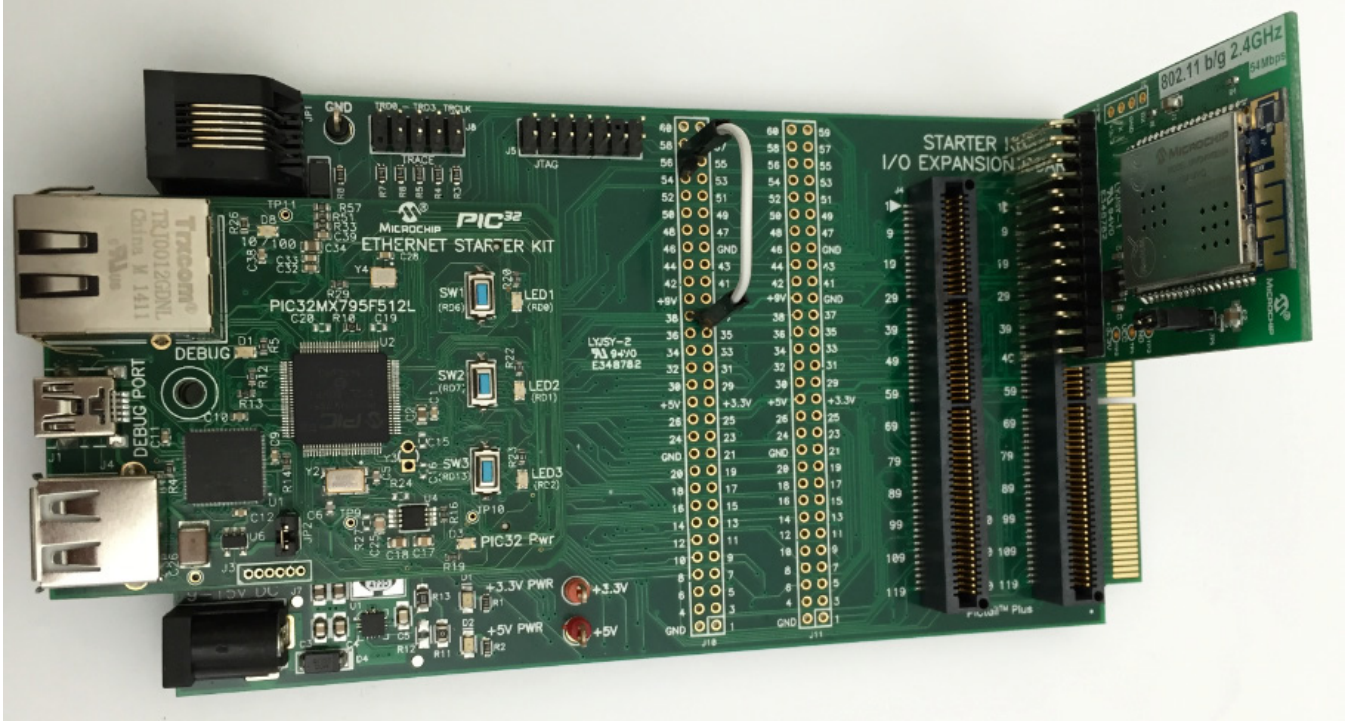
配置硬件

介绍如何配置支持的硬件。

说明

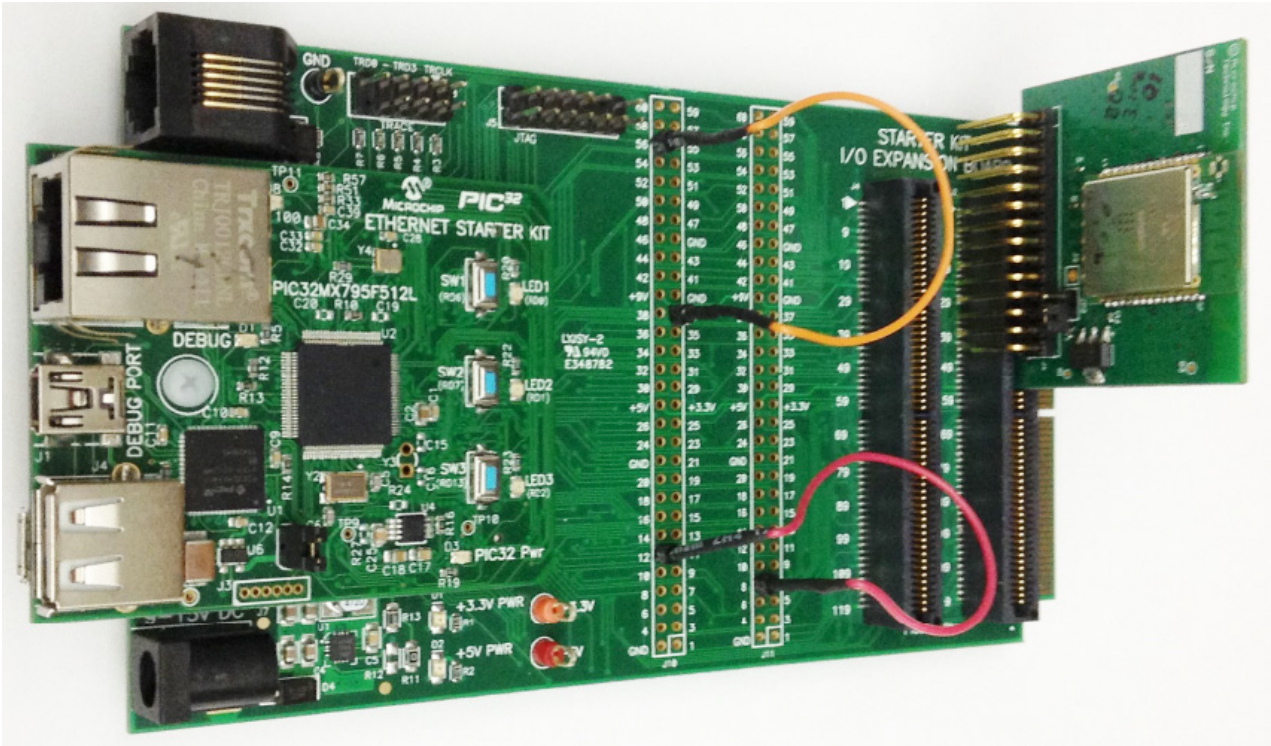
带 PIC32MX795F512L 的 PIC32 以太网入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板配置入门工具包 I/O 扩展板跳线改装，如下图所示：

板载跳线：J10/ 引脚 56 连接至 J10/ 引脚 37 （白色跳线电缆）



带 PIC32MX795F512L 的 PIC32 以太网入门工具包、 MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板配置入门工具包 I/O 扩展板跳线改装，如下图所示：

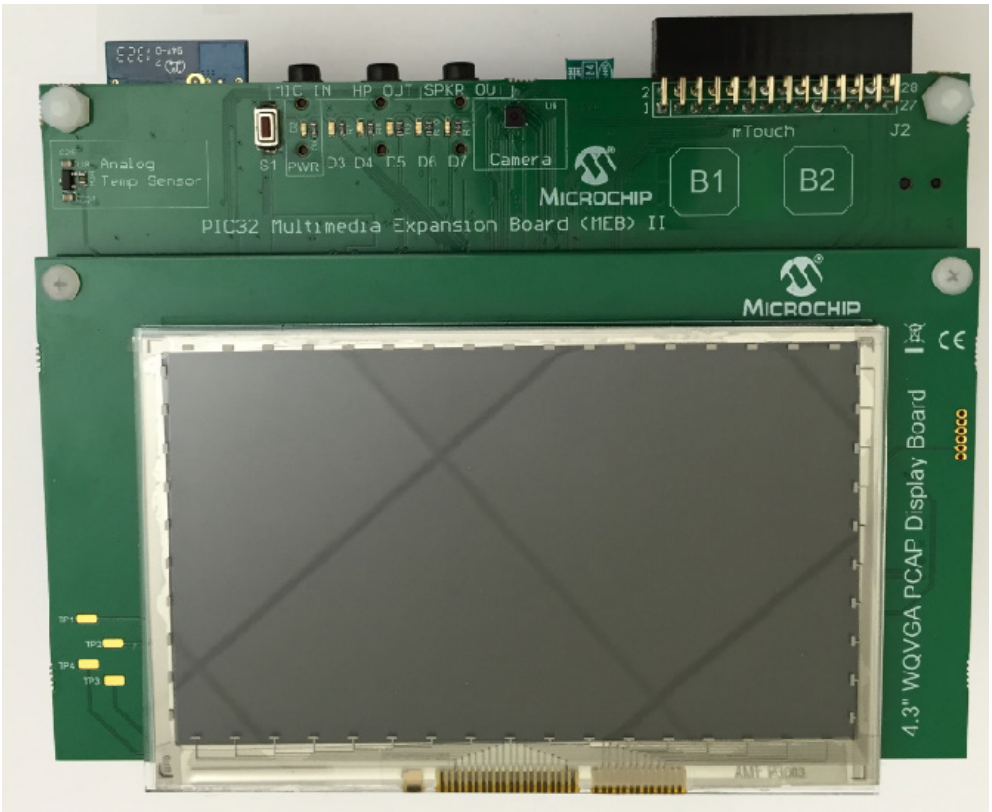
板载跳线：J10/ 引脚 12 连接至 J11/ 引脚 8 （红色跳线电缆）， J10/ 引脚 56 连接至 J10/ 引脚 37 （橙色跳线电缆）。



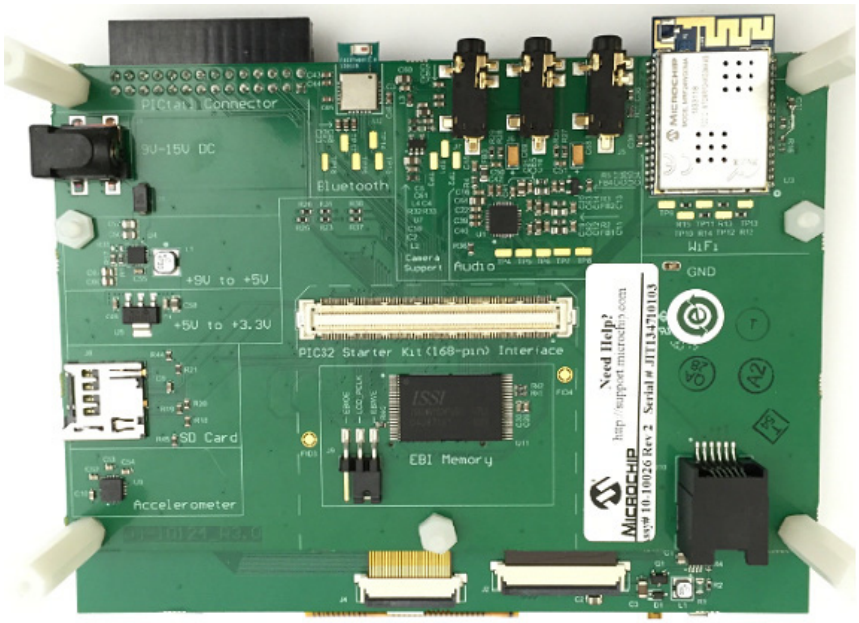
MRF24WG0MA Wi-Fi 模块、 4.3" WQVGA PCAP 显示板和多媒体扩展板 II （MEB II）

配置硬件，如以下各图中所示：

正面配置

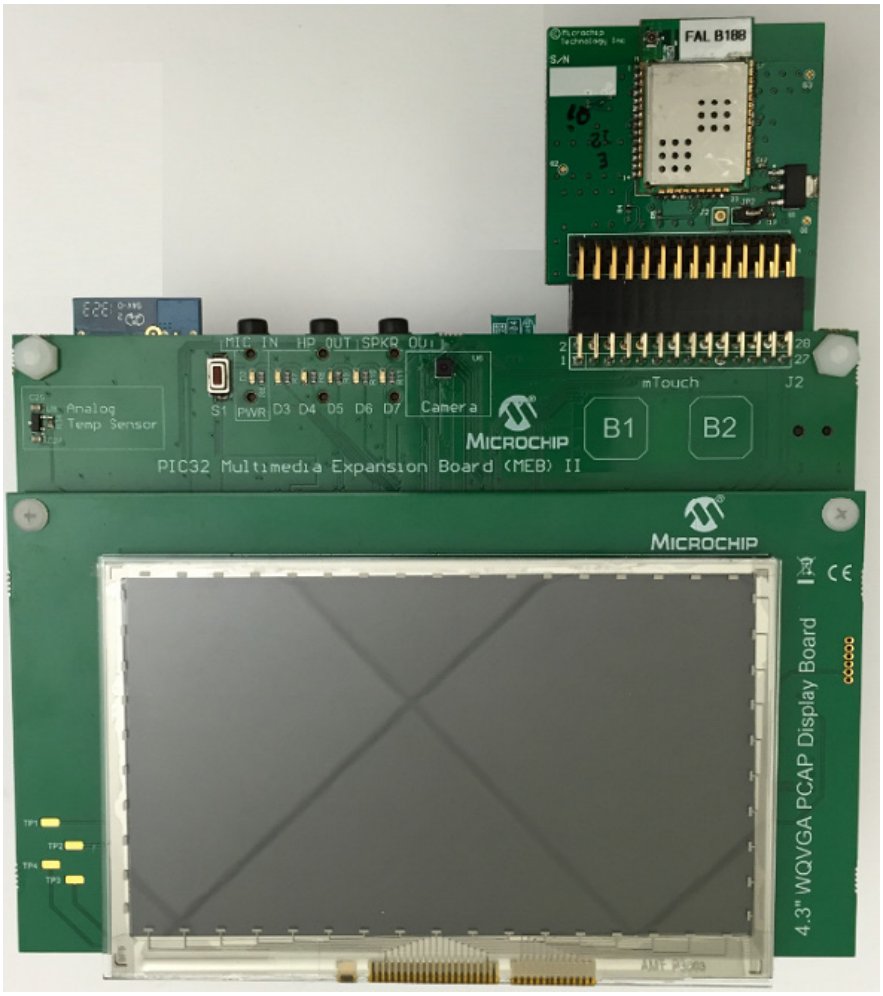


背面配置

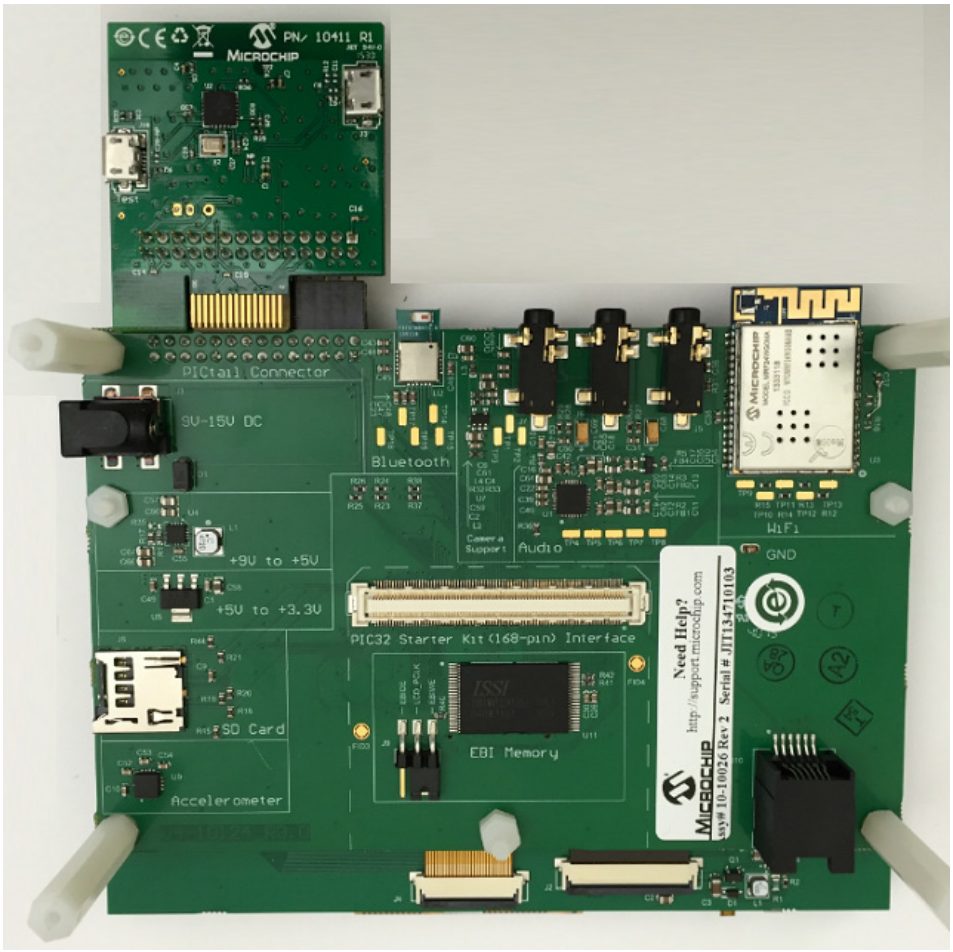


MRF24WG0MA Wi-Fi 模块、MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板、4.3" WQVGA PCAP 显示板和多媒体扩展板 II（MEB II）配置硬件，如以下各图中所示：

正面配置

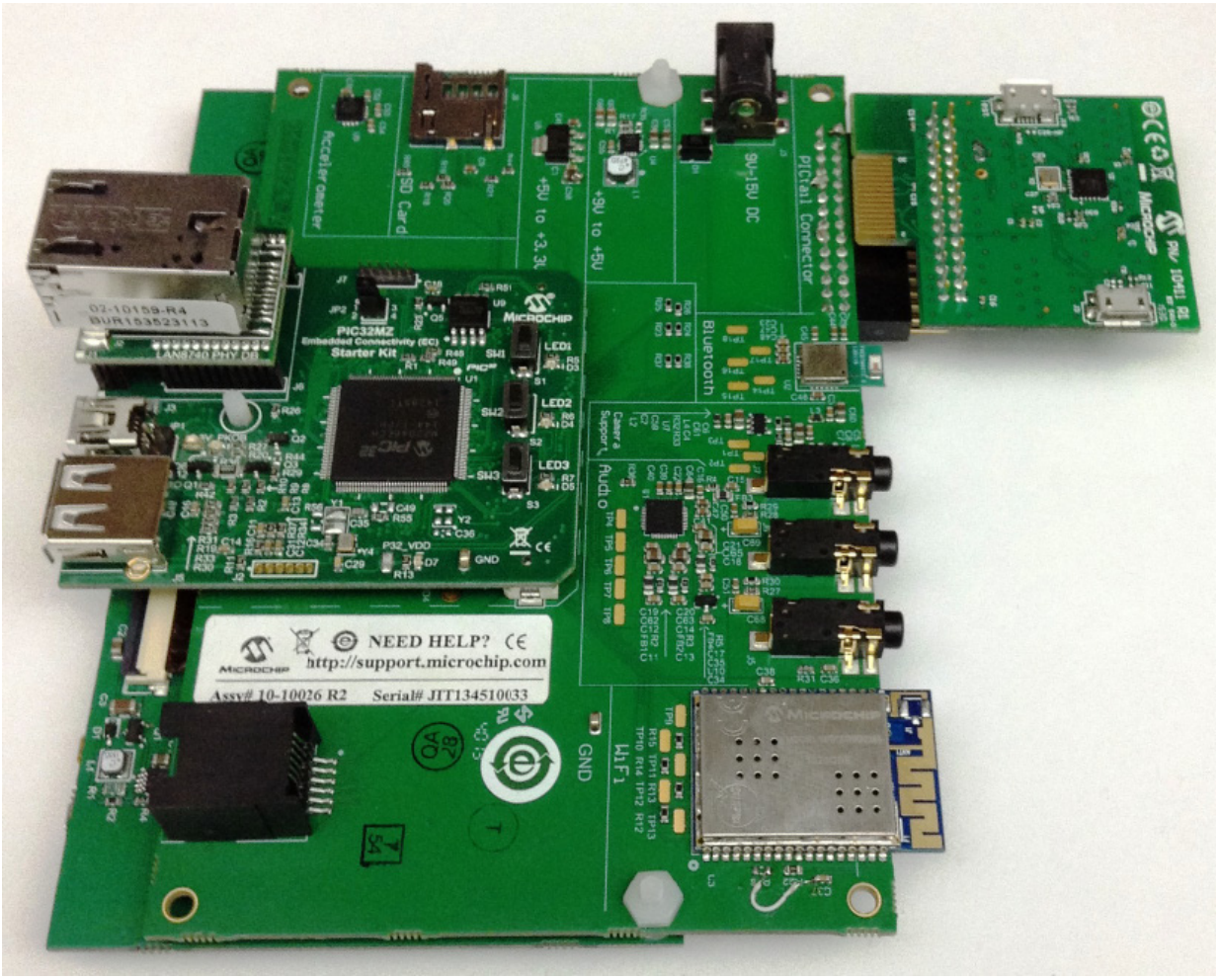


背面配置



带 PIC32MZ2048ECH144 的 PIC32MZ EC 入门工具包、MRF24WG0MA Wi-Fi G 模块、MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板、 4.3" WQVGA PCAP 显示板和多媒体扩展板 II （MEB II）

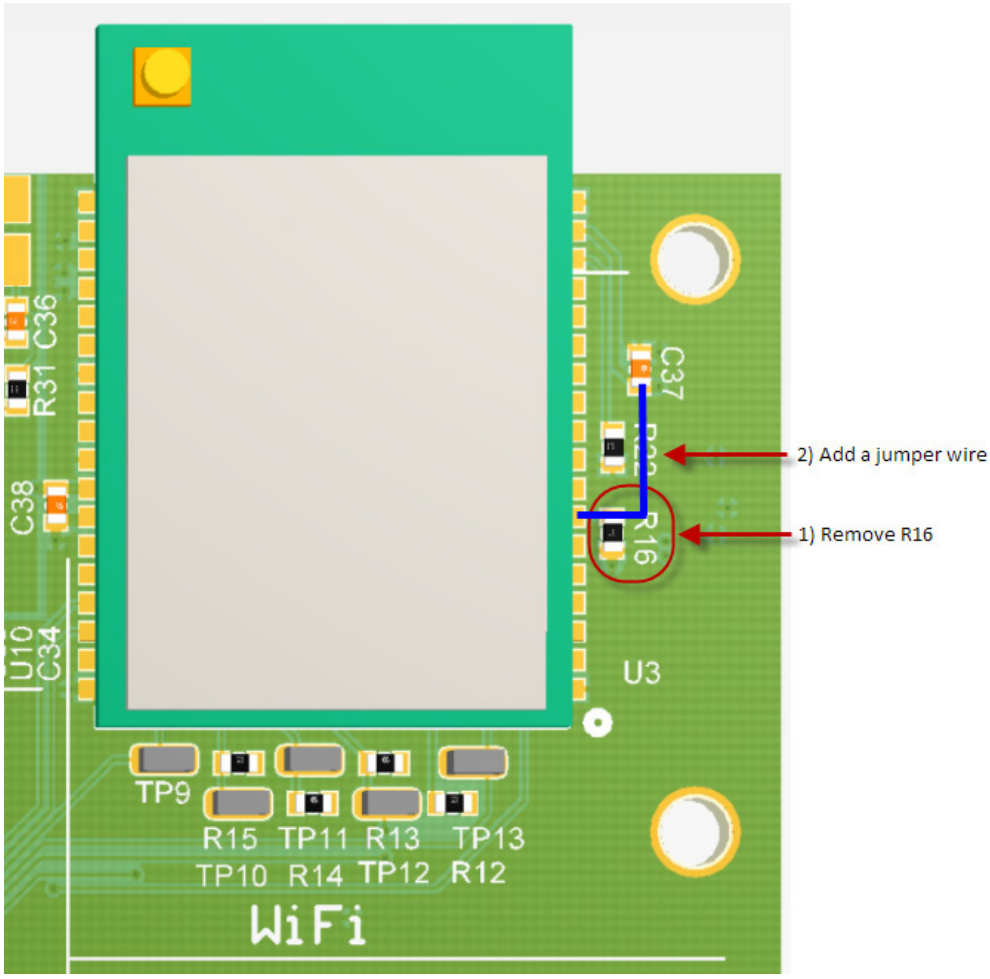
注： PIC32MZ EC 入门工具包 （如下图所示）也可与带 PIC32MZ2048EFM144 的 PIC32MZ EF 入门工具包互换。



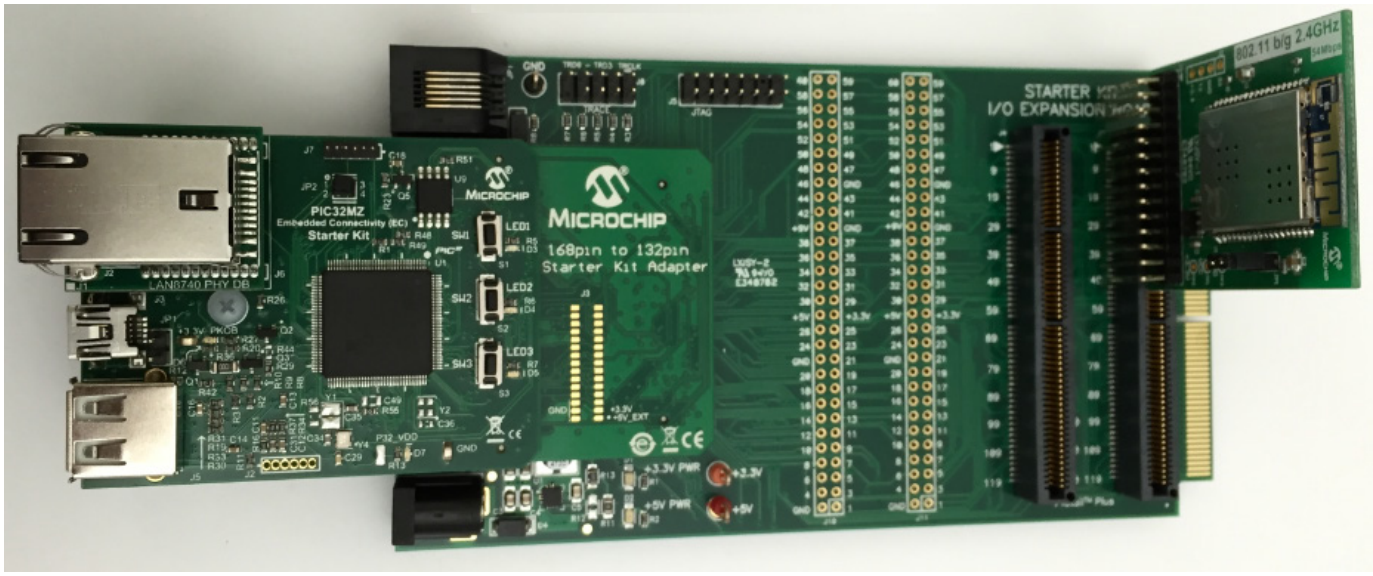
PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和多媒体扩展板 II （MEB II）

由于硬件限制，以太网和 Wi-Fi Web 服务器演示中的 pic32mz_ec_sk+meb2 配置仅通过 Wi-Fi 来运用 Web 服务器。要使能以太网，请使用以下硬件变通方法，并在项目中添加以太网接口，以同时使用以太网和 Wi-Fi 连接运行 Web 服务器。详情请参见下图：

1. 移除 R16，它位于 MEB II 的 Wi-Fi 模块附近。
2. 添加一条从 U3 的引脚 7 连接至 C37 （VCC）下方焊盘的跳线。



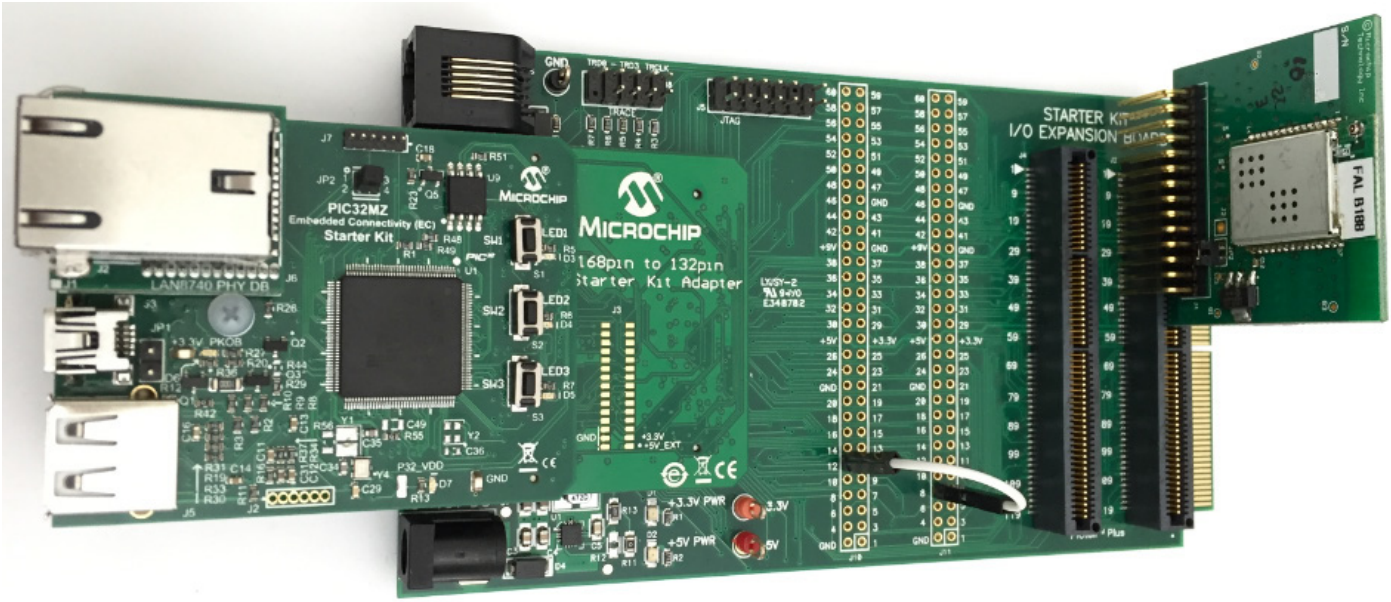
带 PIC32MX2048ECH144 的 PIC32MZ EC 入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板配置硬件，如下图所示：



PIC32MZ EC 入门工具包、 MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

注： PIC32MZ EC 入门工具包 （如下图所示）也可与带 PIC32MZ2048EFM144 的 PIC32MZ EF 入门工具包互换。配置硬件，如下图所示：

板载跳线：J10/ 引脚 12 连接至 J11/ 引脚 8 （白色跳线电缆）



运行演示

本部分介绍如何编译并运行 PIC32 以太网 Wi-Fi Web 服务器演示。

说明

要查看演示应用程序托管的网页，可以打开 Web 浏览器，并通过在地址栏中输入 URL（如 http://mchpboard_w）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

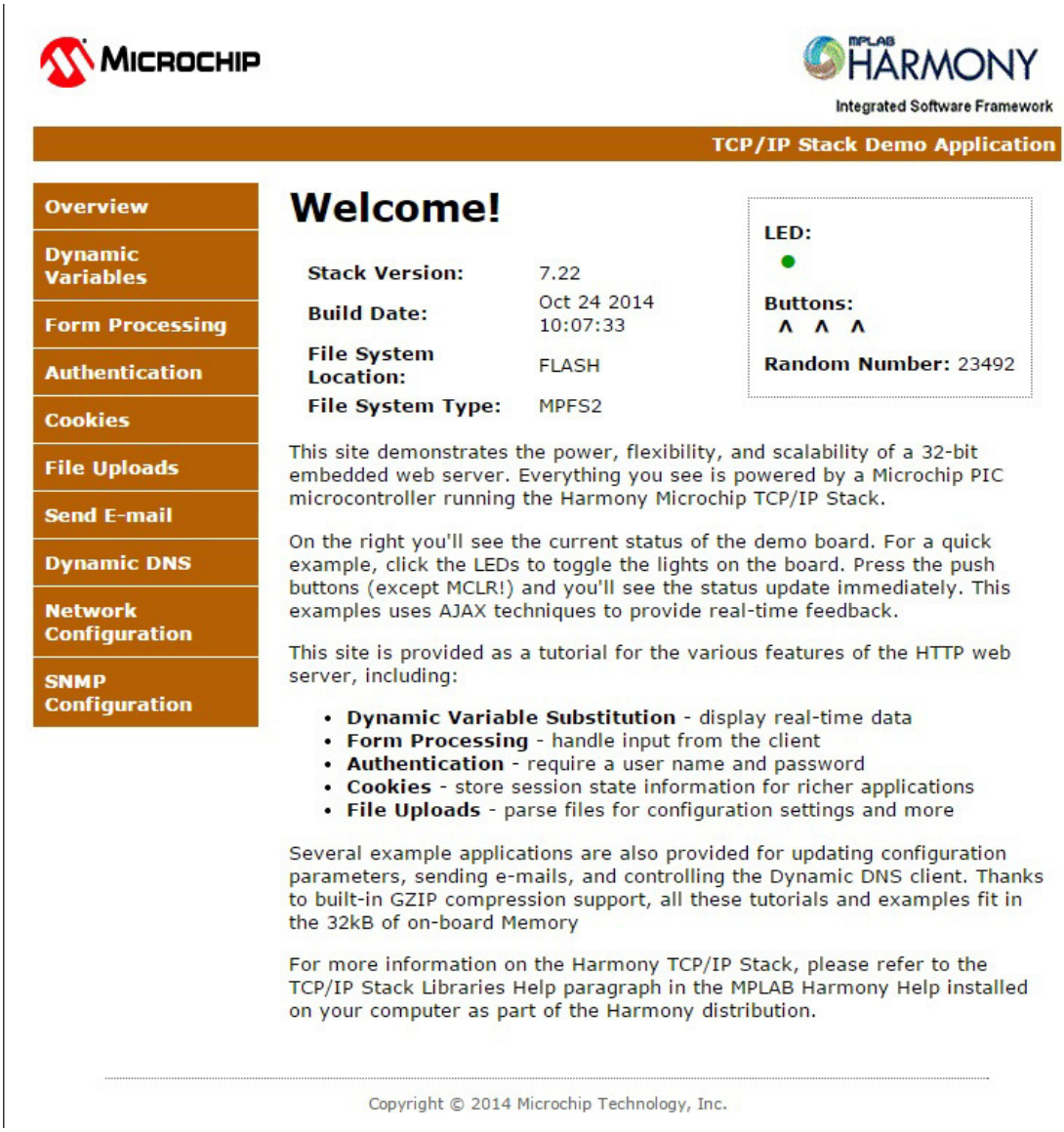
该演示在两种接口（以太网和 Wi-Fi）上运行。可以打开第二个 Web 浏览器选项卡，指向另一个接口（http://mchpboard_e），使两个接口同时运行。

注：

1. TCP/IP 应用程序的 NetBIOS 名称在初始化 TCP/IP 协议栈时指定，它通常处于 tcpip_stack_init.c:: TCPIP_HOSTS_CONFIGURATION 结构的 hostName 成员中。必须使能 NetBIOS 服务， PIC32 演示才会响应 NetBIOS 查询。或者，也可以直接使用电路板的 IPv4 或 IPv6 地址 （如果使能了 IPv6），例如，http://192.168.1.131 或 http://[fdfe:dcba:9876:1:204:a3ff:fe12:128e]。
2. IPv4 和 IPv6 地址可以通过在 PC 侧运行 TCP/IP Discovery 应用程序获取。这要求在编译协议栈时使能 TCP/IP Announce 模块。

请参见 pic32_eth_web_server 配置运行演示部分中的**演示过程**，因为该过程与该配置相同。

注： 关于支持通过 Wi-Fi 设置进行控制的命令的信息，请参见 Wi-Fi 控制台命令。



pic32_wifi_web_server

本部分介绍开始使用 PIC32 Wi-Fi Web 服务器演示应用程序所需要的步骤。

说明

Wi-Fi Web 服务器演示 （`<install-dir>\apps\tcpip\web_server_nvm_mpfs\firmware`）说明如何运用在 PIC32 器件上运行的 HTTP Web 服务器。非易失性存储器（NVM） Microchip 专有文件系统（MPFS） Web 服务器演示将网页存储在内部闪存中，并通过 MPFS API 访问。

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 PIC32 Wi-Fi Web 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `pic32_wifi_web_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/web_server_nvm_mpfs`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
pic32_wifi_web_server.X	<install-dir>/apps/tcpip/web_server_nvm_mpfs/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim+e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板及 MRF24WN PICtail 子板上运行的演示。
pic32mx795_pim+e16+freertos	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板及 MRF24WN PICtail 子板上运行的 FreeRTOS 版本的演示。
pic32mx795_pim+e16+11n+freertos	pic32mx795_pim+e16	在 PIC32MX795F512L PIM 和 Explorer 16 开发板及 MRF24WN PICtail 子板上运行的 FreeRTOS 版本的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM 和 Explorer 16 开发板

- 将 MRF24WG PICtail 连接到最接近 MCU 的 PICtail Plus 插槽。该模块应正对电路板，并一直插到连接器引脚 1 一侧的末端。
- 控制台输出使用电路板上的 DB9 连接器，设置为使用 115200 波特率， 8 位数据，无奇偶校验， 1 个停止位，无流量控制。

运行演示

本部分介绍如何编译并运行 PIC32 Wi-Fi Web 服务器演示。

说明

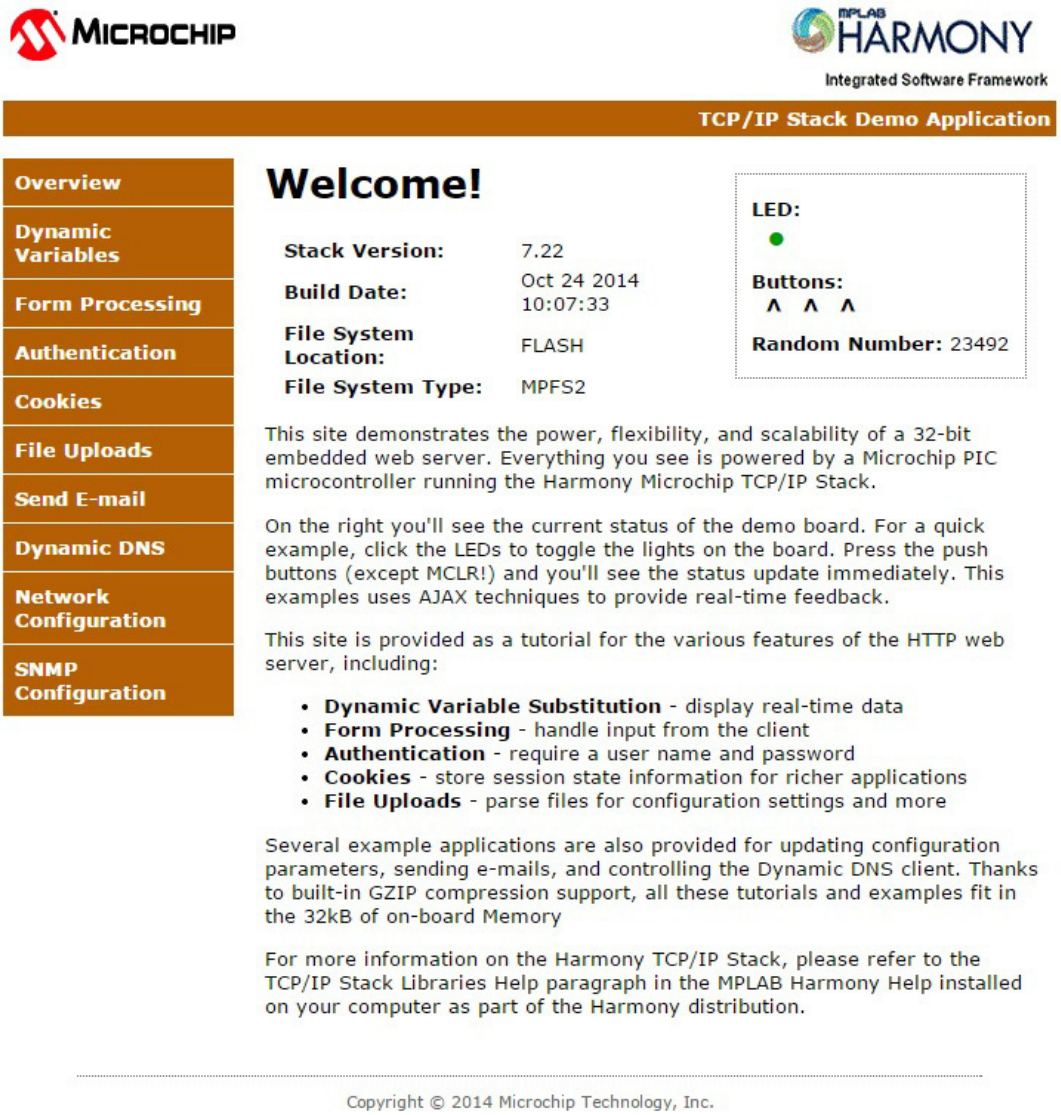
要查看演示应用程序托管的网页，可以打开 Web 浏览器，并通过在地址栏中输入 URL（如 http://mchpboard_w）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

注：

1. TCP/IP 应用程序的 NetBIOS 名称在初始化 TCP/IP 协议栈时指定，它通常处于 tcpip_stack_init.c:: TCPIP_HOSTS_CONFIGURATION 结构的 hostName 成员中。必须使能 NetBIOS 服务， PIC32 演示才会响应 NetBIOS 查询。或者，也可以直接使用电路板的 IPv4 或 IPv6 地址 （如果使能了 IPv6），例如，http://192.168.1.131 或 http://[fdfe:dcba:9876:1:204:a3ff:fe12:128e]。
2. IPv4 和 IPv6 地址可以通过在 PC 侧运行 TCP/IP Discovery 应用程序获取。这要求在编译协议栈时使能 TCP/IP Announce 模块。

请参见 pic32_eth_wifi_web_server 配置运行演示部分中的**演示过程**，因为该过程与该配置相同。

注： 关于支持通过 Wi-Fi 设置进行控制的命令的信息，请参见 Wi-Fi 控制台命令。



web_server_sdcard_fatfs

Web 服务器 SD 卡 FAT 文件系统 TCP/IP 演示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 Web 服务器 SD 卡 FAT FS 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `pic32_eth_web_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/web_server_sdcard_fatfs`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
pic32_eth_web_server.X	<install-dir>/apps/tcpip/web_server_sdcard_fatfs/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk_sd_mmc_pictail	pic32mx_eth_sk	在 PIC32 以太网入门工具包和用于 SD 和 MMC 的 PICtail 子板上，通过 FAT 文件系统演示在 microSD 卡上托管的 Web 服务器。
pic32mx_eth_sk2_sd_mmc_pictail	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 和用于 SD 和 MMC 的 PICtail 子板上，通过 FAT 文件系统演示在 microSD 卡上托管的 Web 服务器。
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	在 PIC32MZ EC 入门工具包和 MEB II 组合上，通过 FAT 文件系统演示在 microSD 卡上托管的 Web 服务器。
pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	在 PIC32MZ EF 入门工具包和 MEB II 组合上，通过 FAT 文件系统演示在 microSD 卡上托管的 Web 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包与入门工具包 I/O 扩展板和用于 SD 和 MMC 的 PICtail 子板

- 将 PIC32 以太网入门工具包插入入门工具包 I/O 扩展板上的应用电路板连接器 （J1）
- 将用于 SD 和 MMC 的 PICtail 子板插入入门工具包 I/O 扩展板上的 PICtail 连接器 （J4）
- 确保 SD 卡已格式化，并已装入在 apps/tcpip/web_server_sdcard_fatfs/firmware/src/web_pages2_sdcard 文件夹中提供的网页。
- 将含有网页的 SD 卡插入用于 SD 和 MMC 的 PICtail 子板上的 SD 卡插槽 （J1），然后将该子板插入 PICtail 连接器
- 将 USB 电缆连接到 PIC32 以太网入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。您可以使用控制台界面观察 IP 地址详细信息和查询协议栈。

PIC32 以太网入门工具包 II 与入门工具包 I/O 扩展板和用于 SD 和 MMC 的 PICtail 子板

- 将 PIC32 以太网入门工具包 II 插入入门工具包 I/O 扩展板上的应用电路板连接器 （J1）
- 将用于 SD 和 MMC 的 PICtail 子板插入入门工具包 I/O 扩展板上的 PICtail 连接器 （J4）
- 确保 SD 卡已格式化，并已装入在 apps/tcpip/web_server_sdcard_fatfs/firmware/src/web_pages2_sdcard 文件夹中提供的网页。
- 将含有网页的 SD 卡插入用于 SD 和 MMC 的 PICtail 子板上的 SD 卡插槽 （J1），然后将该子板插入 PICtail 连接器
- 将 USB 电缆连接到 PIC32 以太网入门工具包 II 背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。您可以使用控制台界面观察 IP 地址详细信息和查询协议栈。

PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

- 将所需的入门工具包插入 MEB II 上的应用电路板连接器
- 确保 microSD 卡已格式化，并已装入在 apps/tcpip/web_server_sdcard_fatfs/firmware/src/web_pages2_sdcard 目录中提供的网页。
- 将含有网页的 microSD 卡插入 MEB II 上的 microSD 卡插槽 （J8）
- 将 USB 电缆连接到 PIC32MZ EC 入门工具包背面的 micro-B USB 连接器
- 在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。您可以使用控制台界面观察 IP 地址详细信息和查询协议栈。

运行演示

本部分介绍如何编译并运行 TCP/IP SD 卡 FAT FS Web 服务器演示。

说明

要查看演示应用程序托管的网页，可以打开 Web 浏览器，并通过在地址栏中输入 URL （如 http://mchpboard_e）然后按 **Enter**，使浏览器定向到运行 HTTP 服务器的电路板。

注：

1. TCP/IP 应用程序的 NetBIOS 名称在初始化 TCP/IP 协议栈时指定，它通常处于 tcpip_stack_init.c:: TCPIP_HOSTS_CONFIGURATION 结构的 hostName 成员中。必须使能 NetBIOS 服务， PIC32 演示才会响应 NetBIOS 查询。或者，也可以直接使用电路板的 IPv4 或 IPv6 地址 （如果使能了 IPv6），例如，http://192.168.1.131 或 http://[fdfe:dcba:9876:1:204:a3ff:fe12:128e]。
2. IPv4 和 IPv6 地址可以通过在 PC 侧运行 TCP/IP Discovery 应用程序获取。这要求在编译协议栈时使能 TCP/IP Announce 模块。

请参见 web_server_nvm_mpfs 演示 pic32_eth_web_server 配置的运行演示部分中的**演示过程**，因为该过程与该配置相同。

wifi_easy_configuration

Wi-Fi 简易配置 TCP/IP 演示。

说明

该演示说明如何将没有键盘或显示屏的 MRF24WG Wi-Fi 设备连接到无线网络。

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 PIC32 Wi-Fi EasyConf 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wifi_easy_configuration.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为`<install-dir>/apps/tcpip/wifi_easy_configuration`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wifi_easy_configuration.X`	`<install-dir>/apps/tcpip/wifi_easy_configuration/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk_ioexp	pic32mx_eth_sk	在使用 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的演示。

pic32mx_eth_sk_ioexp+freertos	pic32mx_eth_sk	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mx795_pim+e16	pic32mx795_pim+e16	在 PIC32MX795F512L PIM （Explorer 16 开发板上）及 MRF24WG PICtail 子板上运行的演示。
pic32mx795_pim+e16+freertos	pic32mx795_pim+e16	在 PIC32MX795F512L PIM （Explorer 16 开发板上）及 MRF24WG PICtail 子板上运行的 FreeRTOS 版本的演示。
pic32mz_ec_sk_ioexp	pic32mz_ec_sk	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的演示。
pic32mz_ec_sk_ioexp+freertos	pic32mz_ec_sk	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ec_sk+meb2	pic32mz_ec_sk+meb2	在连接到 MEB II 与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的演示。
pic32mz_ec_sk+meb2+freertos	pic32mz_ec_sk+meb2	在连接到 MEB II 与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mx795_pim+e16+11n+freertos	pic32mx795_pim+e16	在 PIC32MX795F512L PIM （Explorer 16 开发板上）及 MRF24WN PICtail 子板上运行的 FreeRTOS 版本的演示。
pic32mx_eth_sk_ioexp+11n+freertos	pic32mx_eth_sk	在连接到入门工具包 I/O 扩展板与 MRF24WN PICtail 子板的 PIC32 以太网入门工具包上运行 FreeRTOS 的演示。
pic32mz_ec_sk_ioexp+11n+freertos	pic32mz_ec_sk	在连接到入门工具包I/O扩展板与MRF24WN PICtail子板的PIC32MZ EC 入门工具包上运行 FreeRTOS 的演示。
pic32mz_ec_sk_meb2+11n+freertos	pic32mz_ec_sk+meb2	在 PIC32MZ EC 入门工具包和 MEB II 上运行 FreeRTOS 的演示。
pic32mz_ef_sk+ioexp	pic32mz_ef_sk	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的演示。
pic32mz_ef_sk+ioexp+freertos	pic32mz_ef_sk	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ef_sk+meb2	pic32mz_ef_sk+meb2	在连接到 MEB II 与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的演示。
pic32mz_ef_sk+meb2+freertos	pic32mz_ef_sk+meb2	在连接到 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ef_sk+ioexp+11n+freertos	pic32mz_ef_sk	在连接到 I/O 扩展板与 MRF24WN PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。
pic32mz_ef_sk+meb2+11n+freertos	pic32mz_ef_sk	在连接到 MEB II 与 MRF24WN PICtail 子板的 PIC32MZ EF 入门工具包上运行的 FreeRTOS 版本的演示。

配置硬件

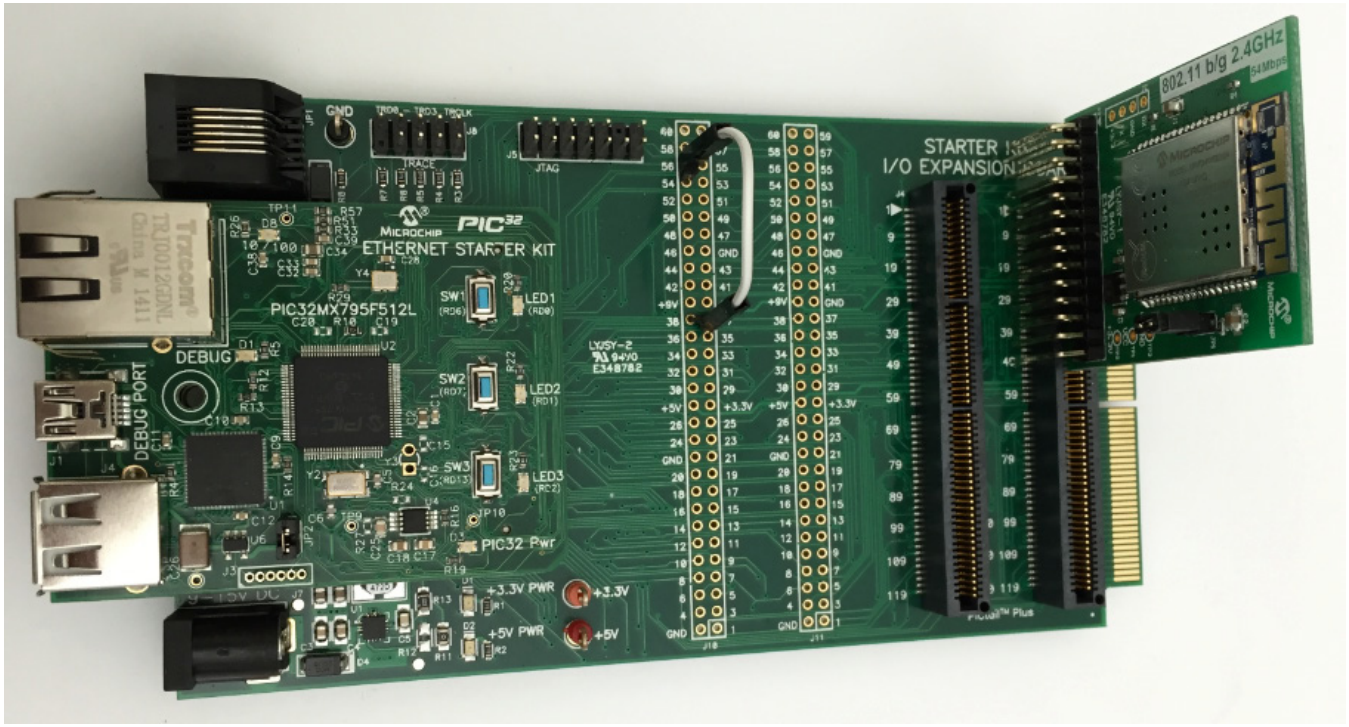
介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

配置入门工具包 I/O 扩展板跳线改装，如下图所示：

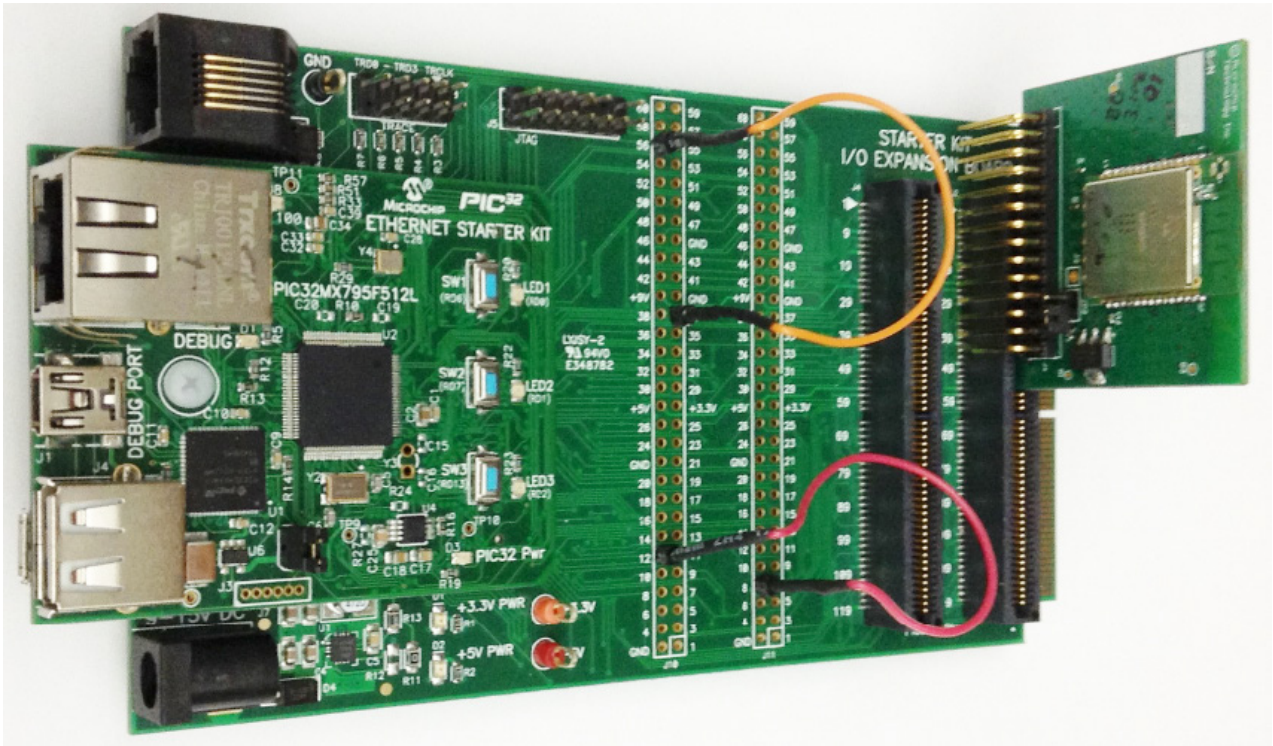
板载跳线：J10/ 引脚 56 连接至 J10/ 引脚 37 （白色跳线电缆）



PIC32 以太网入门工具包、 MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板。

配置入门工具包 I/O 扩展板跳线改装，如下图所示：

板载跳线：J10/ 引脚 12 连接至 J11/ 引脚 8 （红色跳线电缆）， J10/ 引脚 56 连接至 J10/ 引脚 37 （橙色跳线电缆）。



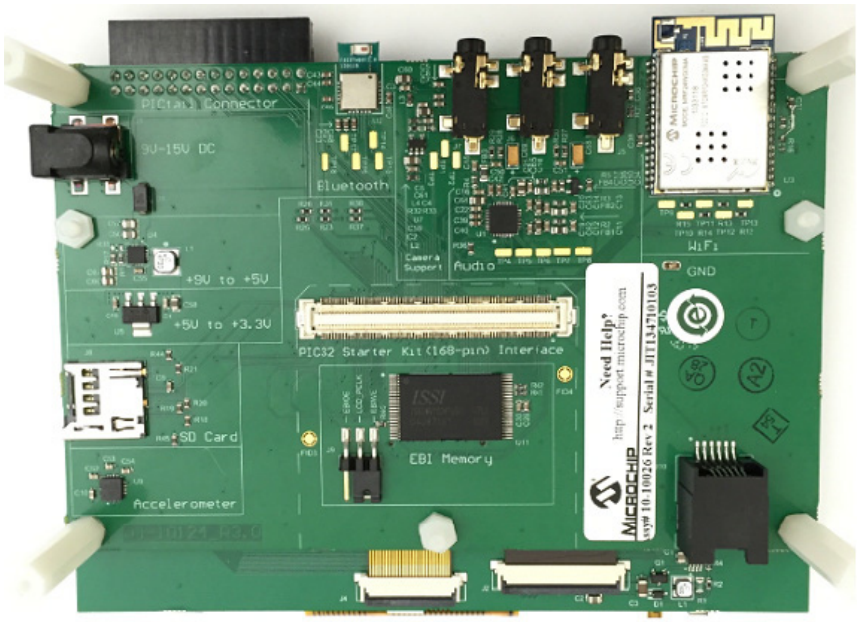
MRF24WG0MA Wi-Fi G 模块、 4.3" WQVGA PCAP 显示板和多媒体扩展板 II （MEB II）

配置硬件，如以下各图中所示：

正面配置

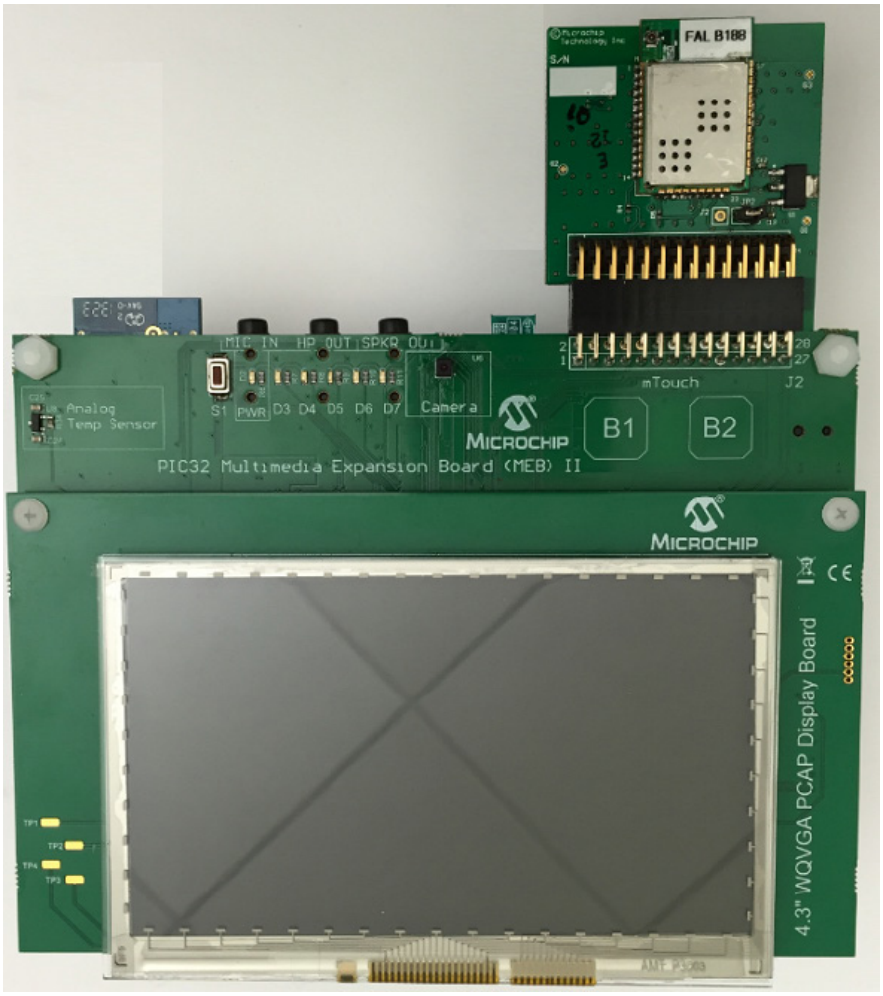


背面配置

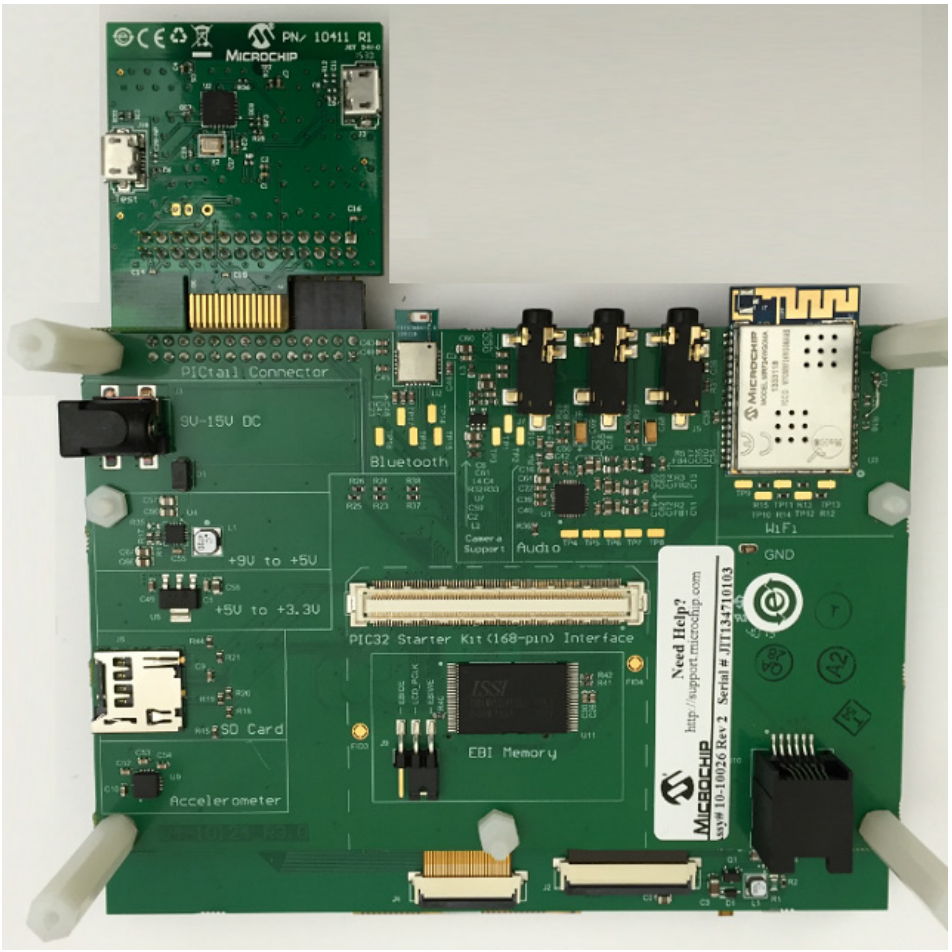


MRF24WF0MA Wi-Fi 模块、 MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板、 4.3" PCAP 显示板和多媒体扩展板 II （MEB II）配置硬件，如以下各图中所示：

正面配置

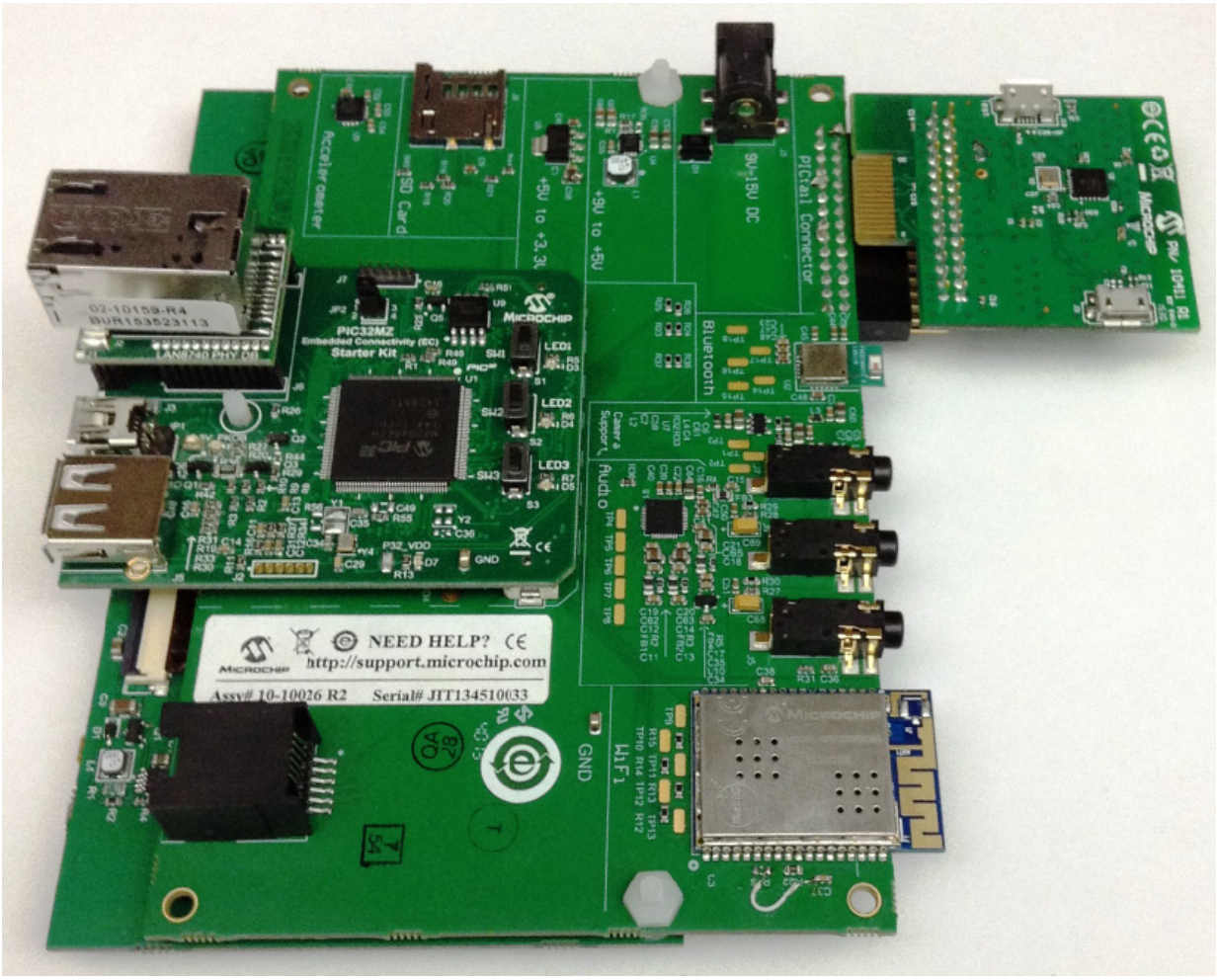

FAL B188
MIC IN HP OUT SPKR OUT
S1 PWR D3 D4 D5 D6 D7 Camera
mTouch
J2
Analog
Temp Sensor
MICROCHIP
B1
B2
PIC32 Multimedia Expansion Board (MEB) II
MICROCHIP
4.3" WQVGA PCAP Display Board

背面配置



带 PIC32MZ2048ECH144 的 PIC32MZ EC 入门工具包、MRF24WG0MA Wi-Fi 模块、MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板、4.3" WQVGA PCAP 显示板和多媒体扩展板 II （MEB II）

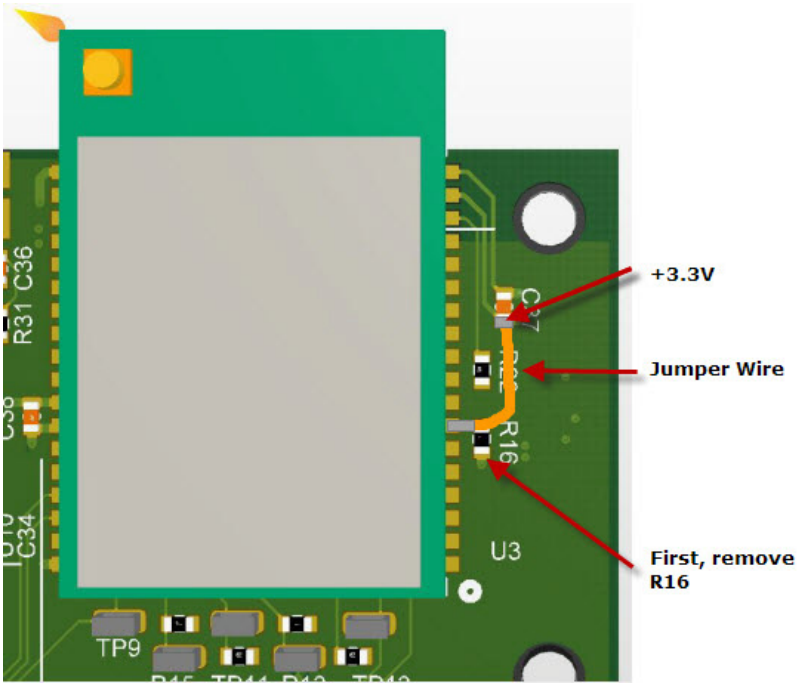
注： PIC32MZ EC 入门工具包 （如下图所示）也可与带 PIC32MZ2048EFM144 的 PIC32MZ EF 入门工具包互换。



PIC32MZ EC 入门工具包或 PIC32MZ EF 入门工具包和 MEB II

由于硬件限制，以太网和 Wi-Fi Web 服务器演示中的 pic32mz_ec_sk+meb2 配置仅通过 Wi-Fi 来运用 Web 服务器。要使能以太网，请使用以下硬件变通方法，并在项目中添加以太网接口，以同时使用以太网和 Wi-Fi 连接运行 Web 服务器。详情请参见下图：

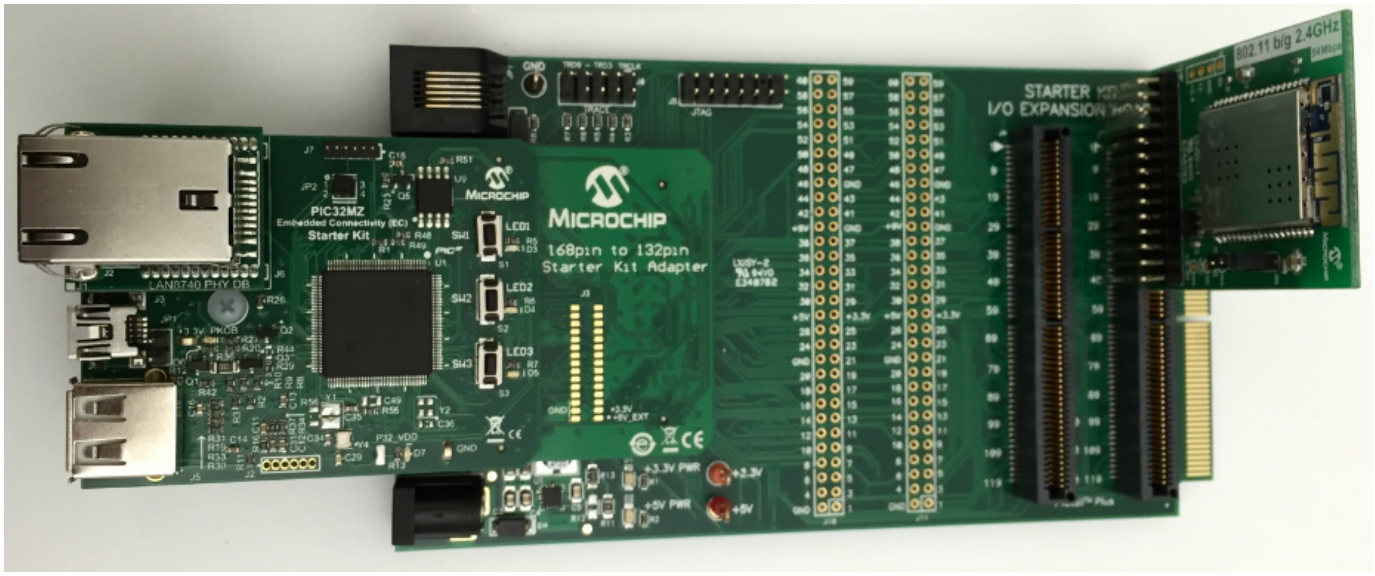
1. 移除 R16，它位于 MEB II 的 Wi-Fi 模块附近。
2. 添加一条从 U3 的引脚 7 连接至 C37 （VCC）下方焊盘的跳线。



带 PIC32MZ2048ECH144 的 PIC32MZ EC 入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

注： PIC32MZ EC 入门工具包 （如下图所示）也可与带 PIC32MZ2048EFM144 的 PIC32MZ EF 入门工具包互换。

配置硬件，如下图所示：

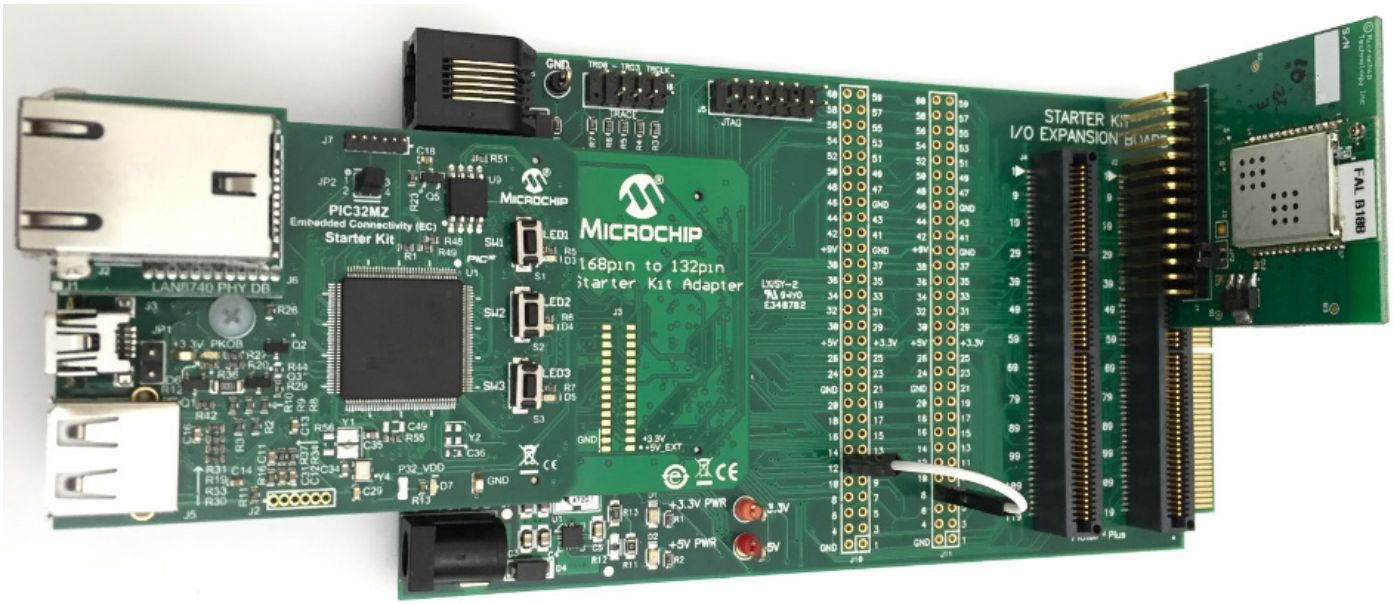


带 PIC32MZ2048ECH144 的 PIC32MZ EC 入门工具包、 MRF24WN0MA Wi-Fi PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

注： PIC32MZ EC 入门工具包 （如下图所示）也可与带 PIC32MZ2048EFM144 的 PIC32MZ EF 入门工具包互换。

配置硬件，如下图所示：

板载跳线： J10/ 引脚 12 连接至 J11/ 引脚 8 （白色跳线电缆）



运行演示

本部分介绍如何编译并使用 MRF24WG Wi-Fi 模块运行 Wi-Fi EasyConfig 演示。

说明

注： 关于支持通过 Wi-Fi 设置进行控制的命令的信息，请参见 Wi-Fi 控制台命令。

该演示会执行以下操作：

- 扫描区域，并在存储器中存储接入点列表
- 切换为 SoftAP 模式，允许其他设备 （智能手机或个人计算机）连接它
- 在某个智能手机或个人计算机以无线方式连接到 MRF24WG 之后，向它供应网页
- 该网页会显示接入点 （Access Point， AP）列表 （来自步骤 1）
- 从智能手机中，您可以选择所需的 AP，并命令 MRF24WG 连接该 AP
- MRF24WG 会连接选定的 AP，并将配置信息存储在非易失性存储器中

使用以下过程来运行演示：

1. 将演示项目装入 MPLAB X IDE。
2. 使用工具包中提供的 USB 电缆将入门工具包板载的 mini-B 调试器端口连接到开发计算机上的 USB 端口。
3. 编译、下载并在电路板上运行演示项目。
4. 在演示运行时，它会扫描本地接入点，并将结果输出到串行控制台。在得到扫描结果后， MRF24WG 会进入 SoftAP 模式 （在该模式下，它的行为类似于一个接入点），并将以下信息输出到串行控制台：

```
==========================
*** Wi-Fi EZConfig Demo ***
==========================
Device:                 MRF24WG （0x3108）
Domain:                 FCC
MAC:                    00 1E C0 10 20 32
SSID:                   MCHPSoftAP
Network Type:           SoftAP
Scan Type:              Passive Scan
Channel List:           6
Beacon Timeout:         40
Retry Count:            3
Security:               Open
Security Key:
Tx Mode:                802.11bg mixed
Power Save:             disabled
```

```
IP Config:            dynamic


Start Wi-Fi Connect  ...
MRF24W IP Address: 0.0.0.0
MRF24W IP Address: 192.168.1.25
MRF24W Event: Connection Successful
  Connected BSSID : 00:00:00:00:00:00
           Channel: 0
```

5. 从智能手机或个人计算机中，连接到 “MCHPSoftAP”网络，它是由该演示启动的 SoftAP 网络。然后，通过在智能手机浏览器中输入 SoftAP 网络的 IP 地址来打开网页。它是在步骤 4 中显示的 IP 地址 （如 192.168.1.25）。当显示网页时：

a. 选择 **Network Configuration**，然后选择 **Scan for Wireless Networks**（扫描无线网络）。MRF24WG 会在网页上显示无线网络的列表。

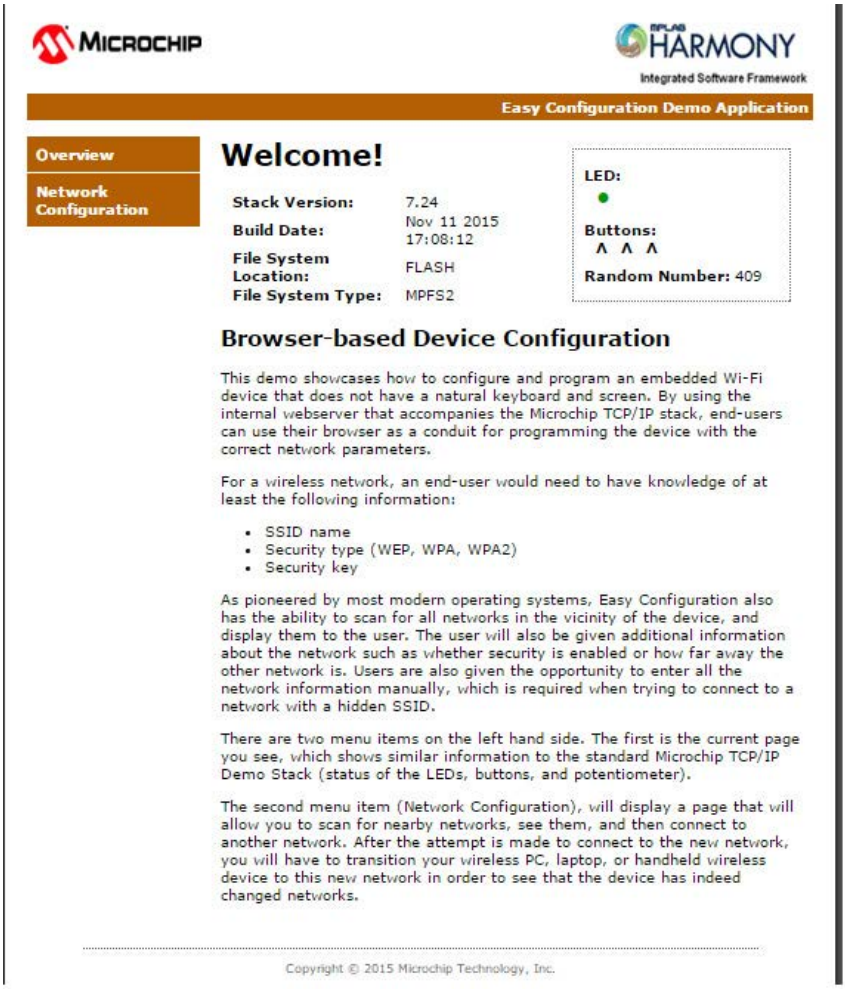
b. 通过单击 AP 的名称，选择 MRF24WG 应连接的所需 AP。

c. 然后， MRF24WG 会连接该接入点，并将配置信息写入非易失性存储器中

d. 控制台输出将显示正在发生的新连接

6. 如果重新运行演示，它会自动连接选定的 AP，因为它会使用存储在非易失性存储器中的配置数据来重新连接所需的 AP。

7. 要重置并从头开始运行演示，可以启动演示，并在命令行输入 **iw_eraseconf** 来擦除所存储的配置。





wifi_g_demo

Wi-Fi G 演示板 TCP/IP 演示。

说明

该演示展示了一个基于浏览器的设备配置应用程序，用于配置无实体键盘和显示屏的嵌入式 Wi-Fi 设备并对其编程。通过使用 Microchip TCP/IP 协议栈随附的内部 Web 服务器，最终用户可以使用他们的浏览器作为一种通道，为设备设定正确的网络参数。

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 PIC32 Wi-Fi G 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wifi_g_demo.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/wifi_g_demo`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wifi_g_demo.X`	`<install-dir>/apps/tcpip/wifi_g_demo/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
wifi_g_demo	wifi_g_db	在 Wi-Fi G 演示板上运行的演示。

配置硬件

介绍如何配置支持的硬件。

说明

Wi-Fi G 演示板与 PICkit™ 3 在线调试器

不需要硬件相关配置或跳线设置更改。



运行演示

本部分介绍如何编译并运行 Wi-Fi G 演示。

说明

该演示展示了一个基于浏览器的设备配置应用程序，用于配置无实体键盘和显示屏的嵌入式 Wi-Fi 设备并对其编程。通过使用 Microchip TCP/IP 协议栈随附的内部 Web 服务器，最终用户可以使用他们的浏览器作为一种通道，为设备设定正确的网络参数。

对于无线网络，最终用户需要至少知道以下信息：

- SSID
- 安全类型 （无、 WEP、 WPA-PSK、 WPA2-PSK 和 WPA-PSK AUTO）
- PSK 安全密钥或密码

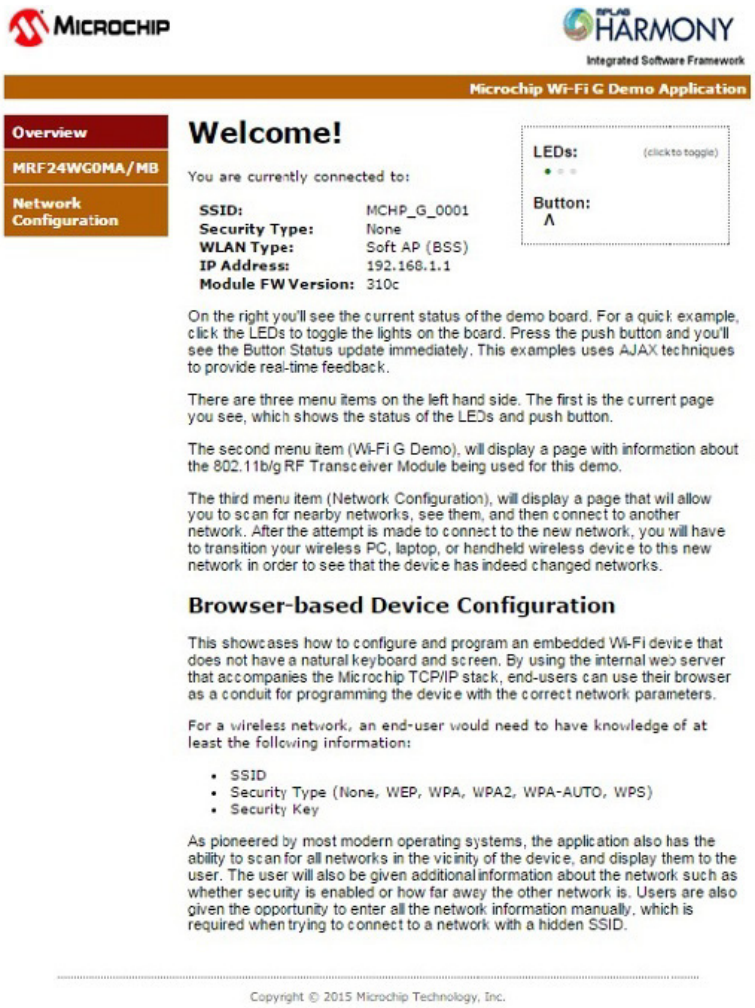
应用程序还能够扫描设备附近的所有网络，并向用户显示它们。此外，还会向用户提供关于网络的其他信息，例如，是否使能了安全性，或其他网络距离多远。此外，用户也可以手动输入所有网络信息，在尝试连接具有隐藏 SSID 的网络时会需要这样。

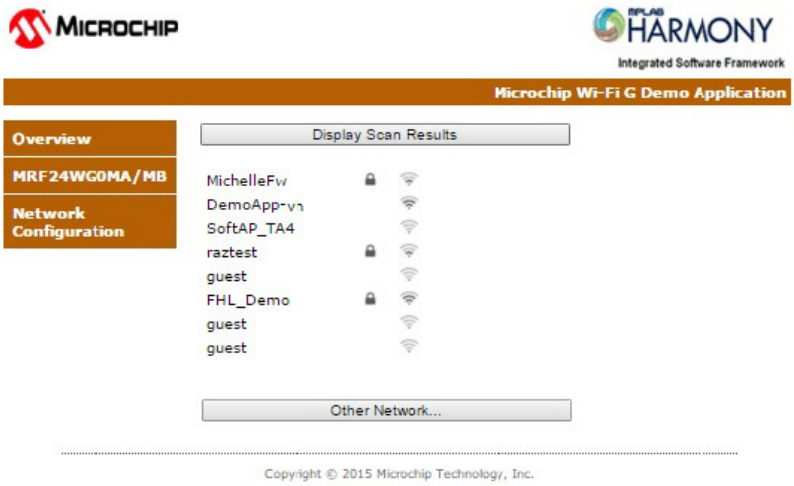
运行演示

该演示可以装入并在 Wi-Fi G 演示板上运行。此外，还需要 PICkit 3 在线调试器，与 MPLAB X IDE 一起用于对 Wi-Fi G 演示板编程。使用以下过程来运行演示：

1. 将 wifi_g_demo 演示项目装入 MPLAB X IDE。
2. 选择 **Run**（运行）进行编译并使用演示十六进制映像对 Wi-Fi G 演示板编程。

注： 关于使用 Wi-Fi 演示板和对其编程的更多信息，请参见 *“Wi-Fi® G Demo Board User's Guide”*（DS50002147）。





wifi_wolfssl_tcp_client

该配置提供 Wi-Fi wolfSSL TCP/IP 客户端演示。

说明

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP 以太网 Wi-Fi wolfSSL TCP 客户端演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wifi_wolfssl_tcp_client.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/wifi_wolfssl_tcp_client`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wifi_wolfssl_tcp_client.X`	`<install-dir>/apps/tcpip/wifi_wolfssl_tcp_client/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk+ioexp	pic32mx_eth_sk	用于连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32 以太网入门工具包的 Wi-Fi wolfSSL TCP 客户端演示。
pic32mz_ec_sk+ioexp	pic32mz_ec_sk	用于连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包的 Wi-Fi wolfSSL TCP 客户端演示。

配置硬件

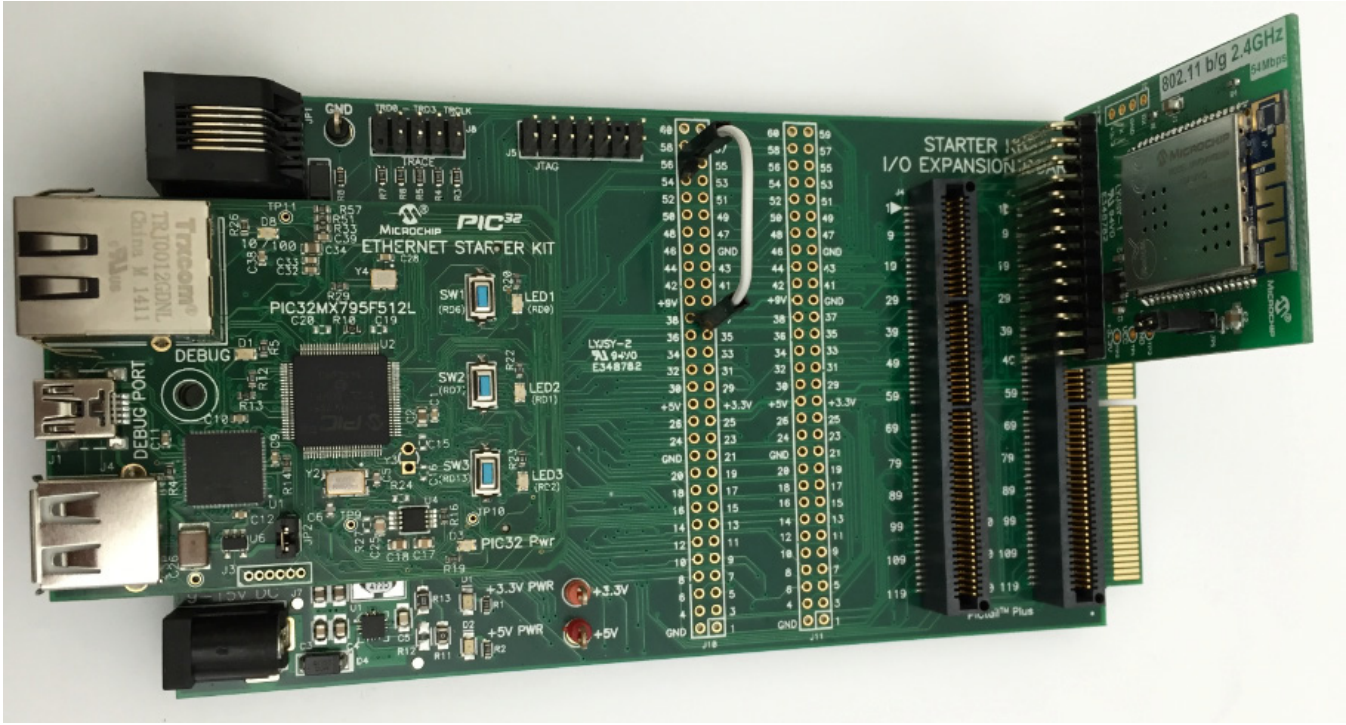
介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

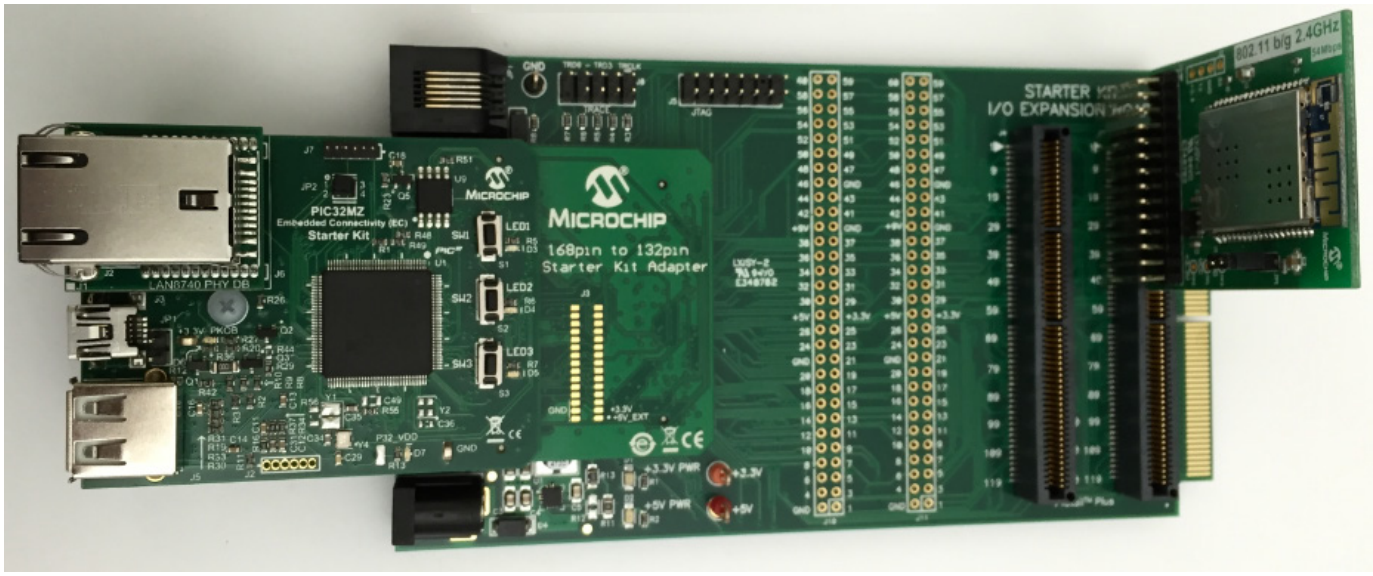
配置入门工具包 I/O 扩展板跳线改装，如下图所示：

板载跳线：J10/ 引脚 56 连接至 J10/ 引脚 37 （白色跳线电缆）



PIC32MZ 嵌入式连接 （EC）入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

对于 PIC32MZ EC 入门工具包，不需要将 J10/ 引脚 56 连接至 J10/ 引脚 37 的板载跳线 （白色跳线电缆）。配置硬件，如下图所示：



运行演示

本部分介绍如何编译并运行演示。

说明

该演示可以通过以太网或 Wi-Fi 执行。

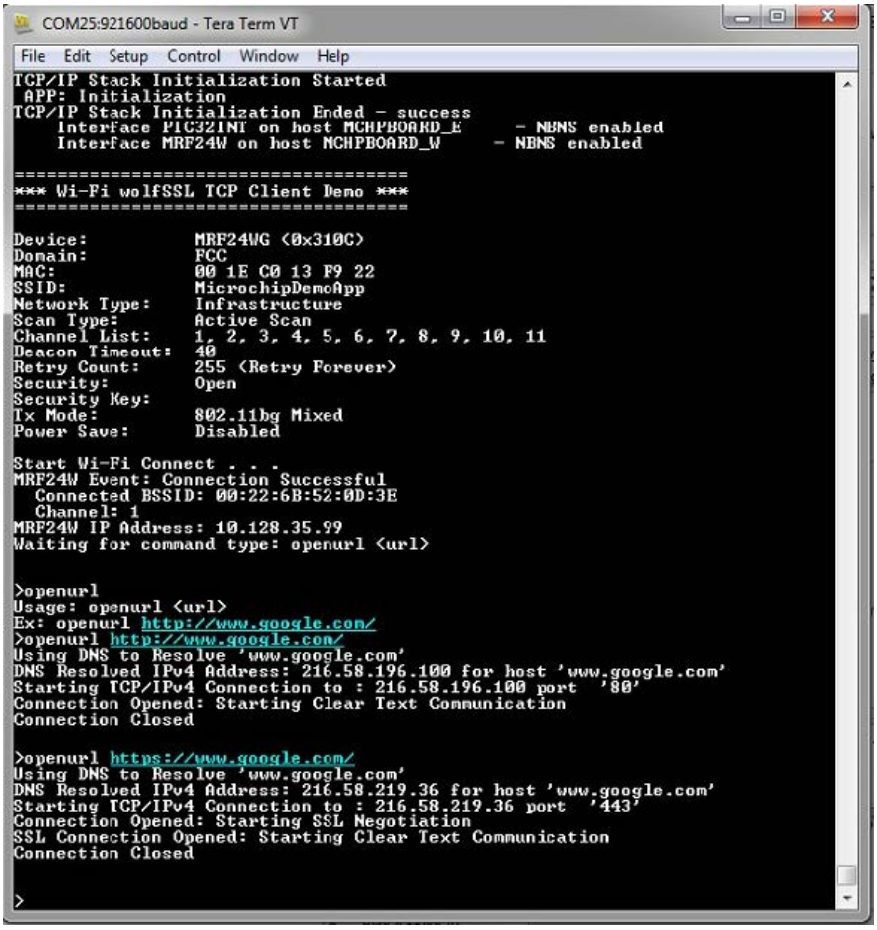
要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

该演示中具有 3 个可通过串行端口使用的命令：

- openurl <url>——<url> 参数必须是完整形式的 URL ；例如， http://www.microchip.com/
- ipmode <mode>——<mode> 参数用于选择 IP 版本。 0—— 任意 IP 版本， 4—— 仅 IPv4， 6—— 仅 IPv6
- stats—— 输出上一次运行 openurl 的统计数据。诸如连接的每个阶段耗用多长时间，以及传输了多少字节之类的统计数据。

在输入 openurl 命令之后，演示会进行 DNS 查询，然后打开到所请求 URL 的连接，并执行简单的 HTTP PUT 命令。结果会被发送到串行端口。如果指定了 https URL，则该连接会先进行 SSL 协商，之后再发送 HTTP PUT 命令。

如果 ipmode 设置为 `0` （任意），则演示会优先使用 IPv6 而不是 IPv4，这意味着它会先查找 IPv6 地址，之后才会查找 IPv4 地址。



wifi_wolfssl_tcp_server

该配置提供 Wi-Fi wolfSSL TCP/IP 服务器演示。

说明

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 TCP/IP 以太网 Wi-Fi wolfSSL TCP 服务器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wifi_wolfssl_tcp_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/wifi_wolfssl_tcp_server`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wifi_wolfssl_tcp_server.X`	`<install-dir>/apps/tcpip/wifi_wolfssl_tcp_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk+ioexp	pic32mx_eth_sk	用于连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32 以太网入门工具包的 Wi-Fi wolfSSL TCP 服务器演示。
pic32mz_ec_sk+ioexp	pic32mz_ec_sk	用于连接到入门工具包 I/O 扩展板与 MRF24WG PICtail 子板的 PIC32MZ EC 入门工具包的 Wi-Fi wolfSSL TCP 服务器演示。

配置硬件

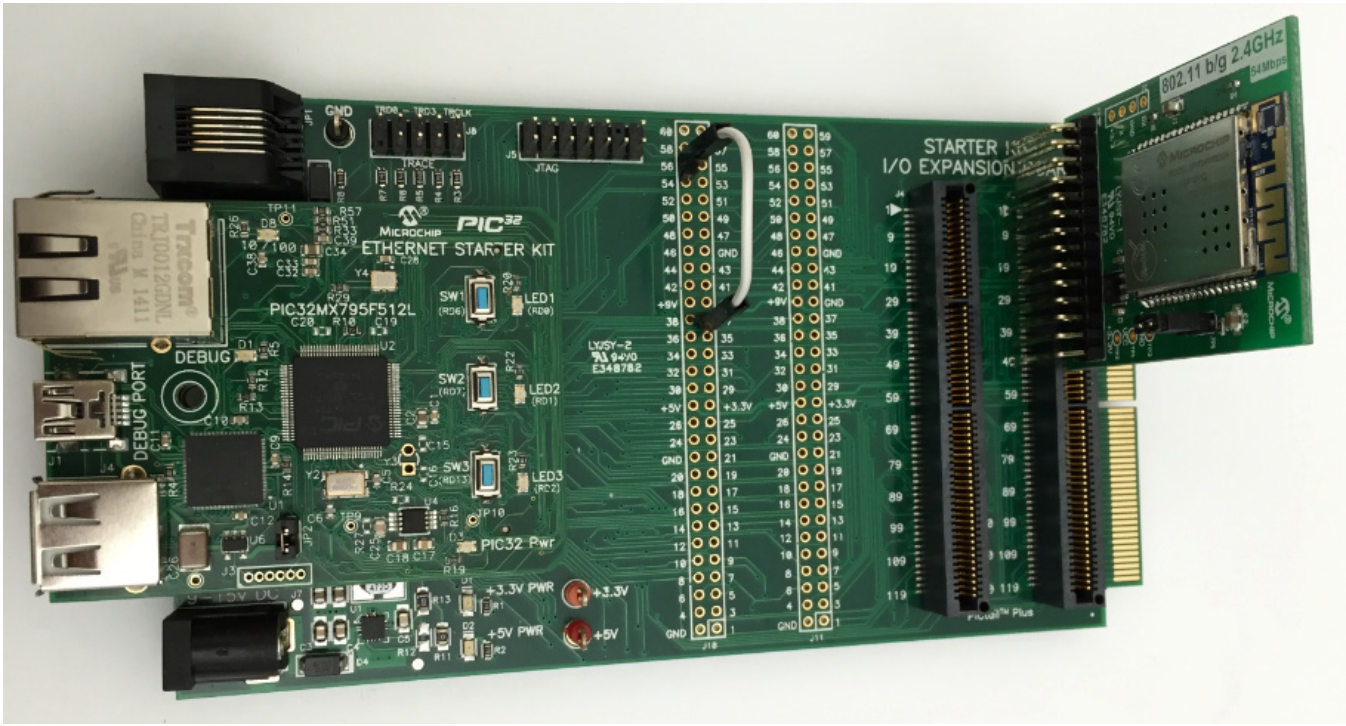
介绍如何配置支持的硬件。

说明

带 PIC32MX795F512L 的 PIC32 以太网入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

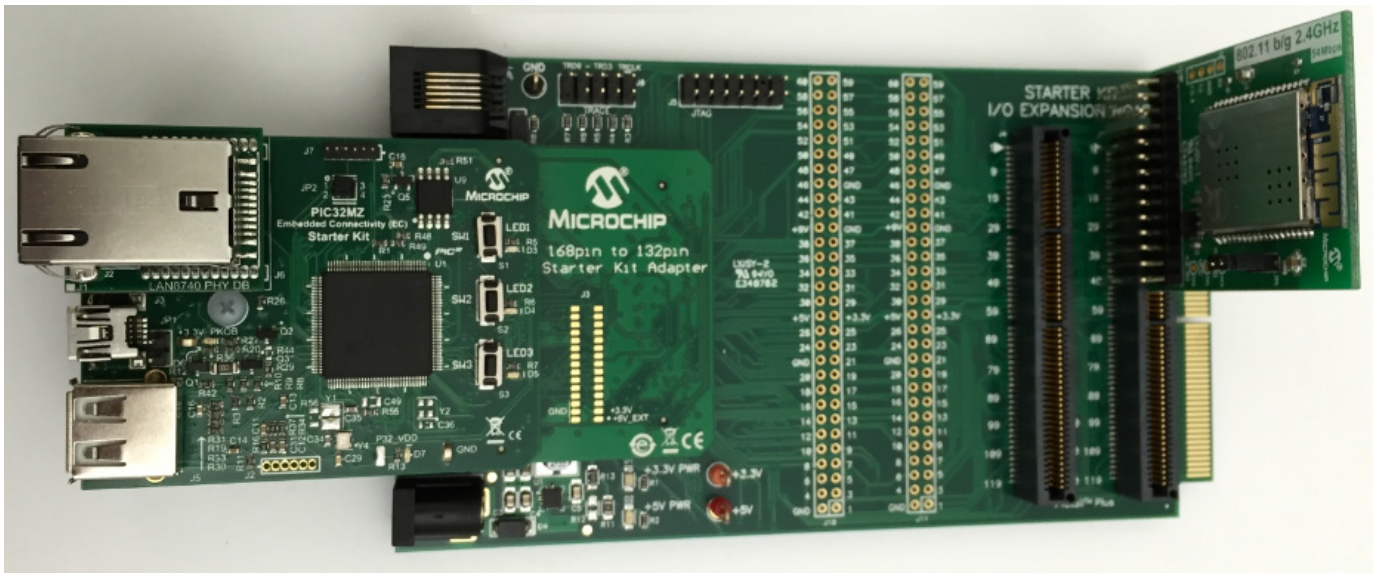
配置入门工具包 I/O 扩展板跳线改装，如下图所示：

板载跳线：J10/ 引脚 56 连接至 J10/ 引脚 37 （白色跳线电缆）



带 PIC32MZ2048ECH144 的 PIC32MZ EC 入门工具包、 MRF24WG0MA Wi-Fi G PICtail/PICtail Plus 子板和入门工具包 I/O 扩展板

对于 PIC32MZ EC 入门工具包，不需要将 J10/ 引脚 56 连接至 J10/ 引脚 37 的板载跳线 （白色跳线电缆）。配置硬件，如下图所示：



运行演示

本部分介绍如何编译并运行演示。

说明

该演示可以通过以太网或 Wi-Fi 执行。

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

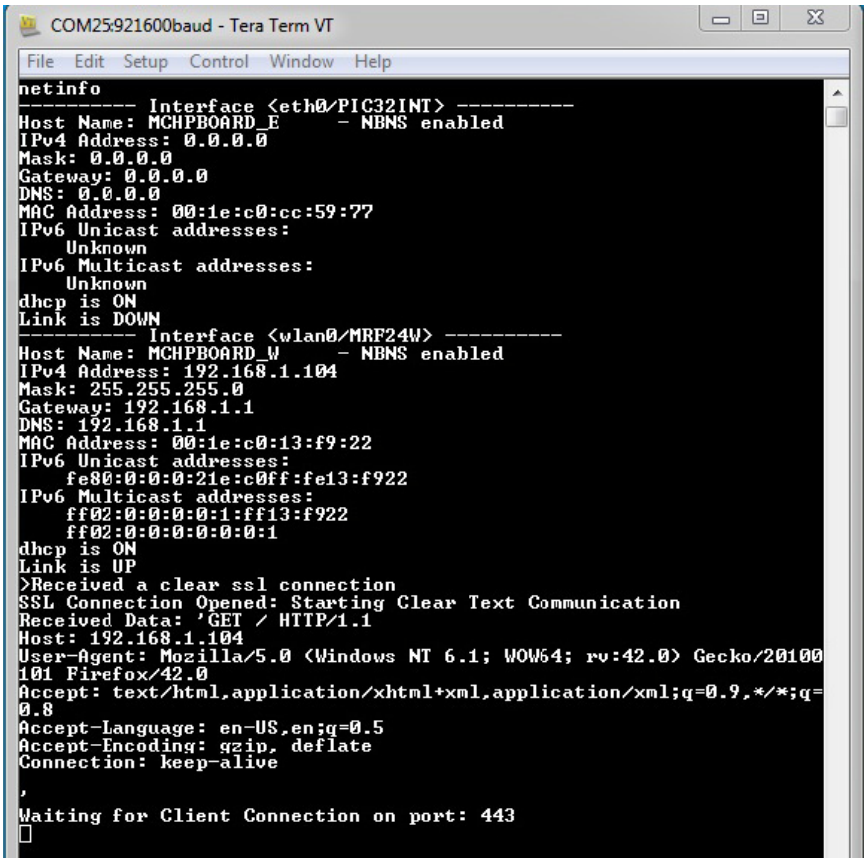
该演示中具有 3 个可通过串行端口使用的命令：

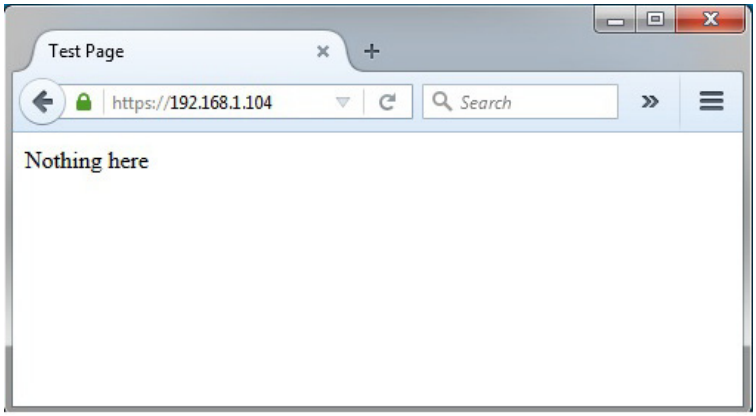
- openurl <url>——<url> 参数必须是完整形式的 URL ； 例如， http://www.microchip.com/

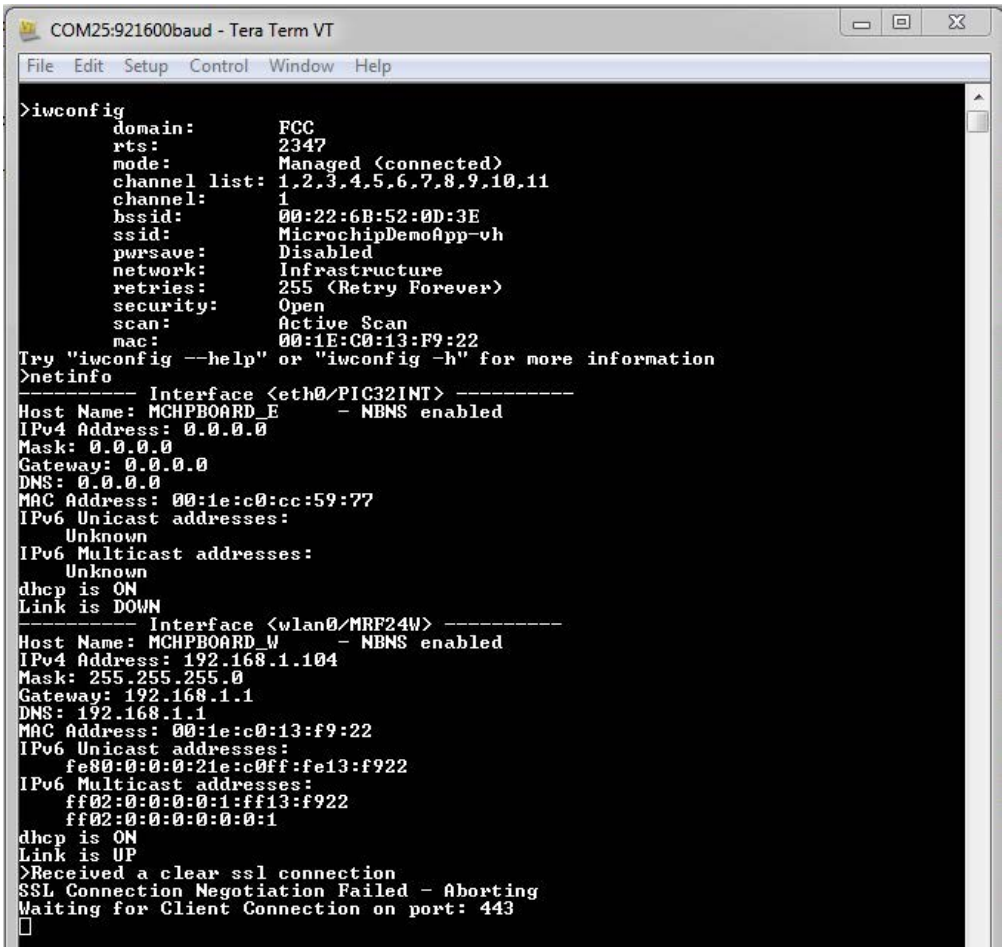
- ipmode <mode>——<mode> 参数用于选择 IP 版本。 0—— 任意 IP 版本， 4—— 仅 IPv4， 6—— 仅 IPv6
- stats—— 输出上一次运行 openurl 的统计数据。诸如连接的每个阶段耗用多长时间，以及传输了多少字节之类的统计数据。

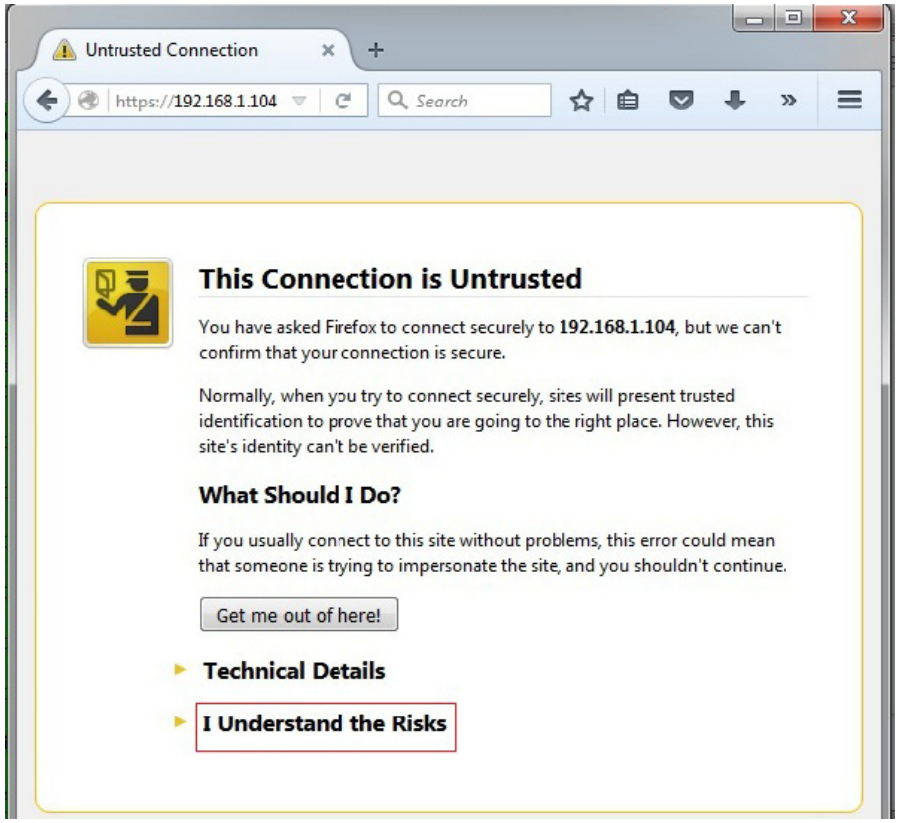
在输入 openurl 命令之后，演示会进行 DNS 查询，然后打开到所请求 URL 的连接，并执行简单的 HTTP PUT 命令。结果会被发送到串行端口。如果指定了 https URL，则该连接会先进行 SSL 协商，之后再发送 HTTP PUT 命令。

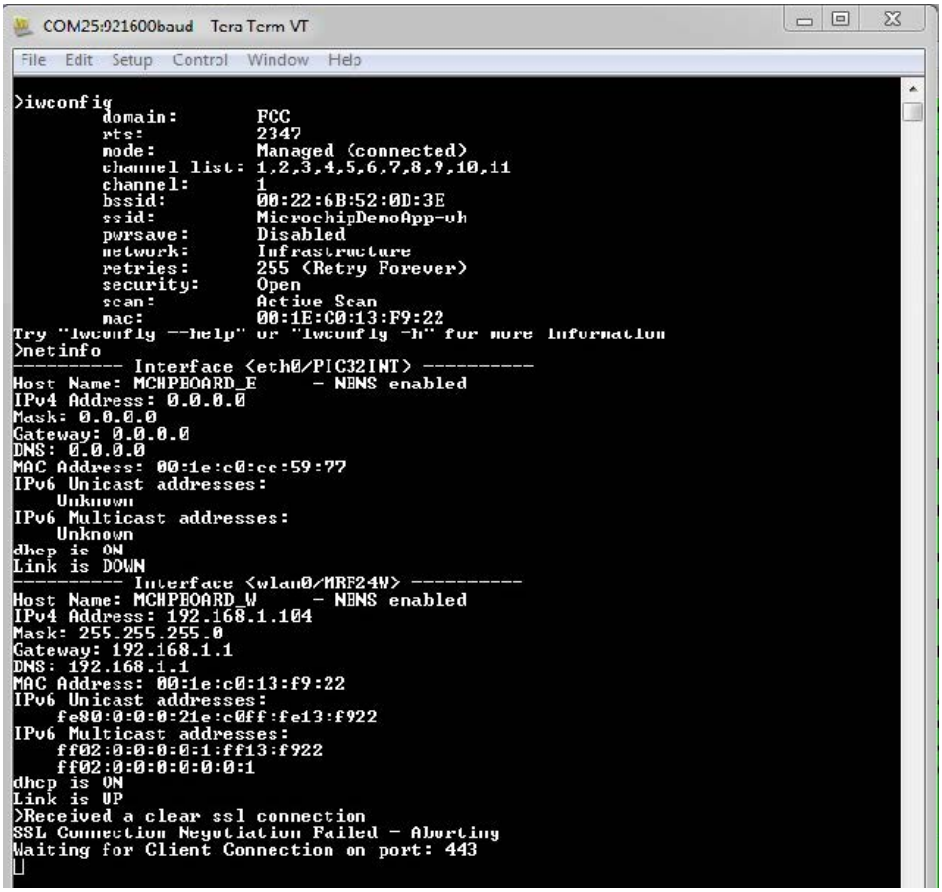
如果 ipmode 设置为 0 （任意），则演示会优先使用 IPv6 而不是 IPv4，这意味着它会先查找 IPv6 地址，之后才会查找 IPv4 地址。

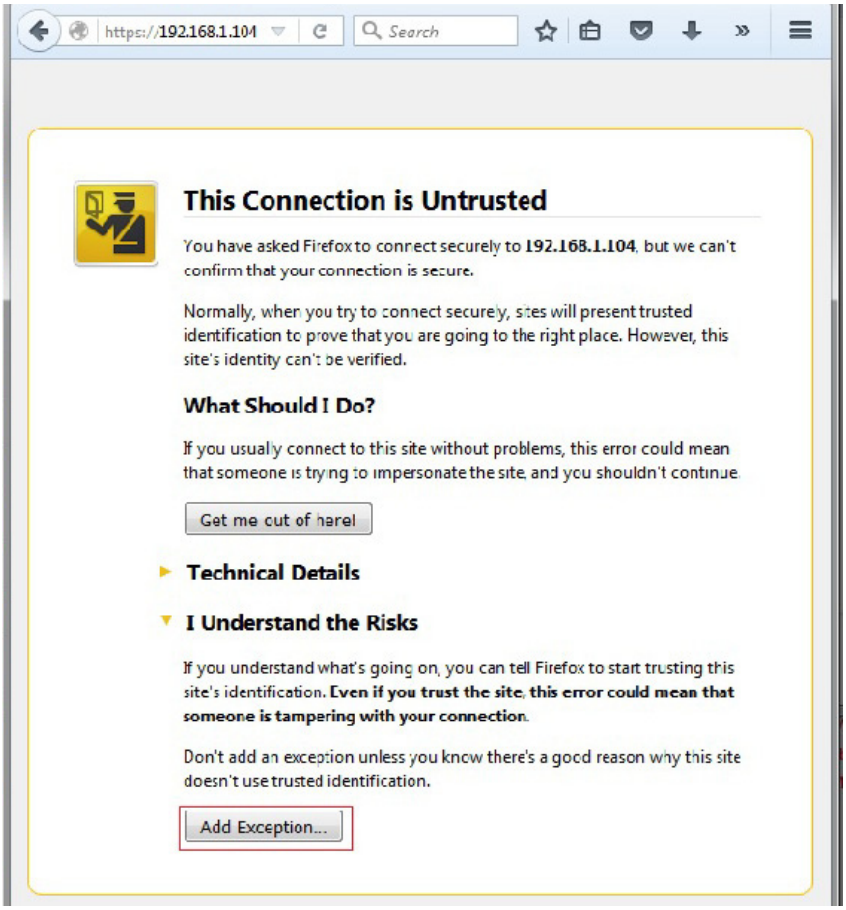


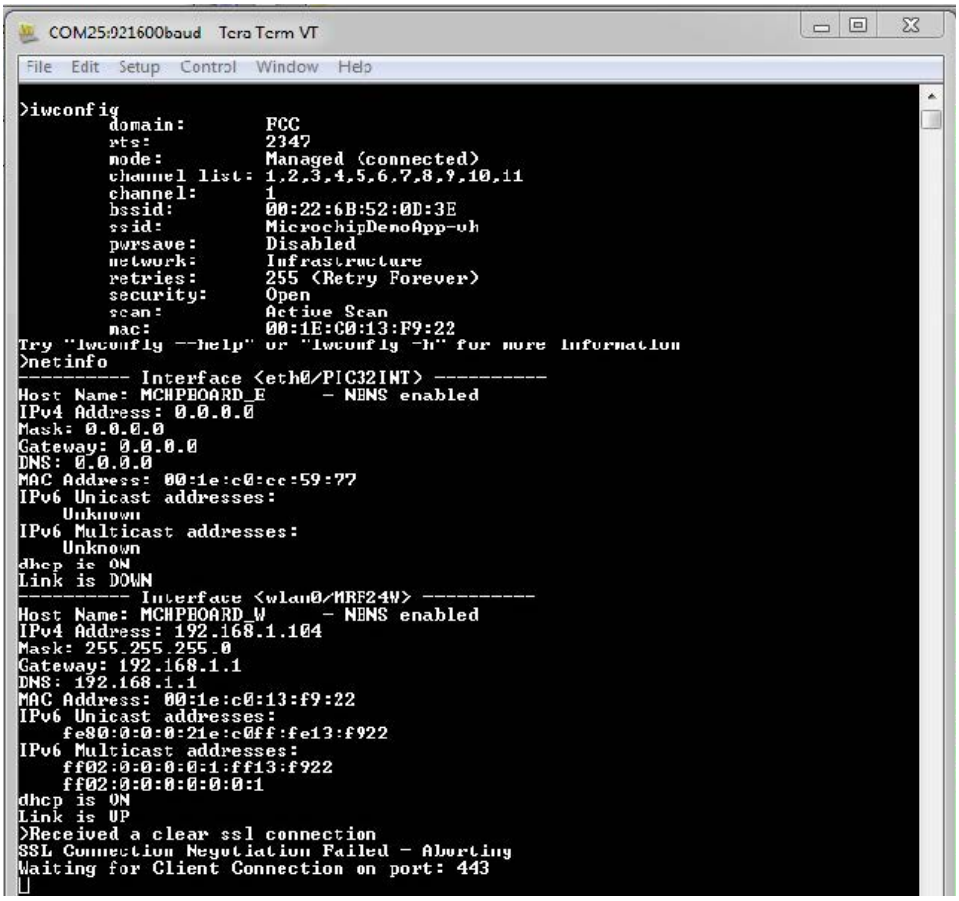


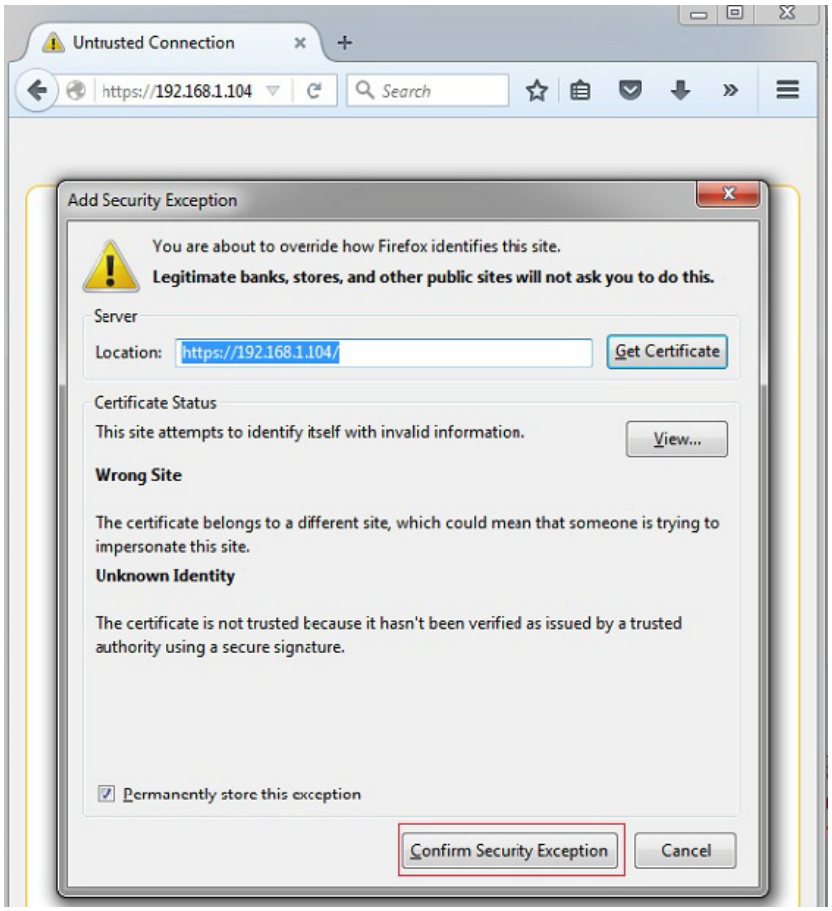

COM25:921600baud - Tera Term VT
File Edit Setup Control Window Help
>iwconfig
domain: FCC
rts: 2347
mode: Managed (connected)
channel list: 1,2,3,4,5,6,7,8,9,10,11
channel: 1
bssid: 00:22:6B:52:0D:3E
ssid: MicrochipDemoApp-vh
pwrsave: Disabled
network: Infrastructure
retries: 255 (Retry Forever)
security: Open
scan: Active Scan
mac: 00:1E:C0:13:F9:22
Try "iwconfig --help" or "iwconfig -h" for more information
>netinfo
---------- Interface <eth0/PIC32INT> ----------
Host Name: MCHPBOARD_E - NBNS enabled
IPv4 Address: 0.0.0.0
Mask: 0.0.0.0
Gateway: 0.0.0.0
DNS: 0.0.0.0
MAC Address: 00:1e:c0:cc:59:77
IPv6 Unicast addresses:
Unknown
IPv6 Multicast addresses:
Unknown
dhcp is ON
Link is DOWN
---------- Interface <wlan0/MRF24W> ----------
Host Name: MCHPBOARD_W - NBNS enabled
IPv4 Address: 192.168.1.104
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS: 192.168.1.1
MAC Address: 00:1e:c0:13:f9:22
IPv6 Unicast addresses:
fe80:0:0:0:21e:c0ff:fe13:f922
IPv6 Multicast addresses:
ff02:0:0:0:0:1:ff13:f922
ff02:0:0:0:0:0:0:1
dhcp is ON
Link is UP
>Received a clear ssl connection
SSL Connection Negotiation Failed - Aborting
Waiting for Client Connection on port: 443

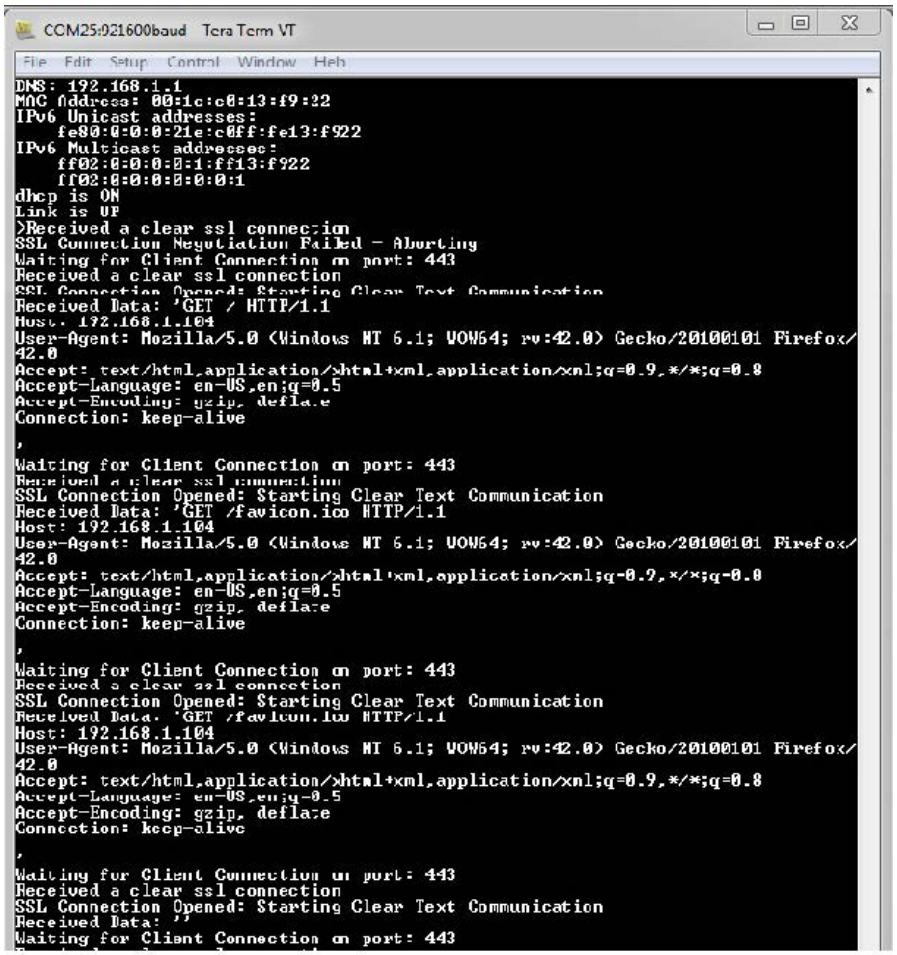

Untrusted Connection
https://192.168.1.104
Search
This Connection is Untrusted
You have asked Firefox to connect securely to 192.168.1.104, but we can't confirm that your connection is secure.
Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.
What Should I Do?
If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.
Get me out of here!
Technical Details
I Understand the Risks

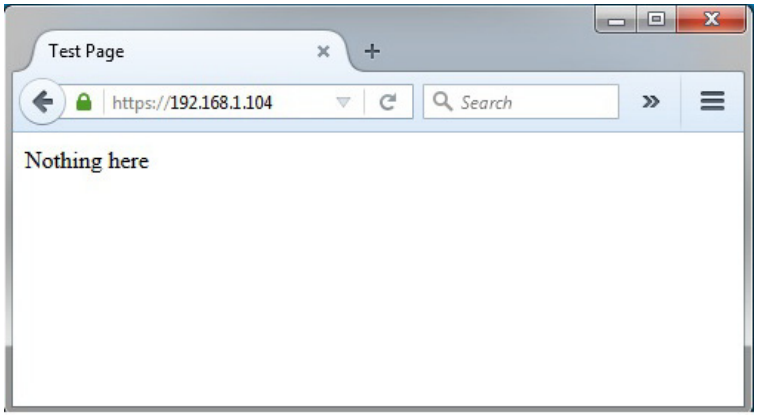

COM25:921600baud Tera Term VT
File Edit Setup Control Window Help
>iwconfig
domain: FCC
rts: 2347
mode: Managed (connected)
channel list: 1,2,3,4,5,6,7,8,9,10,11
channel: 1
bssid: 00:22:6B:52:0D:3E
ssid: MicrochipDemoApp-uh
pwrsave: Disabled
network: Infrastructure
retries: 255 (Retry Forever)
security: Open
scan: Active Scan
mac: 00:1E:C0:13:F9:22
Try "iwconfig --help" or "iwconfig -h" for more information
>netinfo
---------- Interface <eth0/PIC32INT> ----------
Host Name: MCHPBOARD_E - NBNS enabled
IPv4 Address: 0.0.0.0
Mask: 0.0.0.0
Gateway: 0.0.0.0
DNS: 0.0.0.0
MAC Address: 00:1e:c0:cc:59:77
IPv6 Unicast addresses:
Unknown
IPv6 Multicast addresses:
Unknown
dhcp is ON
Link is DOWN
---------- Interface <wlan0/MRF24W> ----------
Host Name: MCHPBOARD_W - NBNS enabled
IPv4 Address: 192.168.1.104
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS: 192.168.1.1
MAC Address: 00:1e:c0:13:f9:22
IPv6 Unicast addresses:
fe80:0:0:0:21e:c0ff:fe13:f922
IPv6 Multicast addresses:
ff02:0:0:0:0:1:ff13:f922
ff02:0:0:0:0:0:0:1
dhcp is ON
Link is UP
>Received a clear ssl connection
SSL Connection Negotiation Failed - Aborting
Waiting for Client Connection on port: 443


https://192.168.1.104
Search
This Connection is Untrusted
You have asked Firefox to connect securely to 192.168.1.104, but we can't confirm that your connection is secure.
Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.
What Should I Do?
If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.
Get me out of here!
Technical Details
I Understand the Risks
If you understand what's going on, you can tell Firefox to start trusting this site's identification. Even if you trust the site, this error could mean that someone is tampering with your connection.
Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.
Add Exception...


COM25:921600baud Tera Term VT
File Edit Setup Control Window Help
>iwconfig
domain: FCC
rts: 2347
mode: Managed (connected)
channel list: 1,2,3,4,5,6,7,8,9,10,11
channel: 1
bssid: 00:22:6B:52:0D:3E
ssid: MicrochipDemoApp-uh
pwrsave: Disabled
network: Infrastructure
retries: 255 (Retry Forever)
security: Open
scan: Active Scan
mac: 00:1E:C0:13:F9:22
Try "iwconfig --help" or "iwconfig -h" for more information
>netinfo
---------- Interface <eth0/PIC32INT> ----------
Host Name: MCHPBOARD_E - NBNS enabled
IPv4 Address: 0.0.0.0
Mask: 0.0.0.0
Gateway: 0.0.0.0
DNS: 0.0.0.0
MAC Address: 00:1e:c0:cc:59:77
IPv6 Unicast addresses:
Unknown
IPv6 Multicast addresses:
Unknown
dhcp is ON
Link is DOWN
---------- Interface <wlan0/MRF24W> ----------
Host Name: MCHPBOARD_W - NBNS enabled
IPv4 Address: 192.168.1.104
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS: 192.168.1.1
MAC Address: 00:1e:c0:13:f9:22
IPv6 Unicast addresses:
fe80:0:0:0:21e:c0ff:fe13:f922
IPv6 Multicast addresses:
ff02:0:0:0:0:1:ff13:f922
ff02:0:0:0:0:0:0:1
dhcp is ON
Link is UP
>Received a clear ssl connection
SSL Connection Negotiation Failed - Aborting
Waiting for Client Connection on port: 443


Untrusted Connection
https://192.168.1.104
Search
Add Security Exception
You are about to override how Firefox identifies this site.
Legitimate banks, stores, and other public sites will not ask you to do this.
Server
Location: https://192.168.1.104/
Get Certificate
Certificate Status
This site attempts to identify itself with invalid information.
View...
Wrong Site
The certificate belongs to a different site, which could mean that someone is trying to impersonate this site.
Unknown Identity
The certificate is not trusted because it hasn't been verified as issued by a trusted authority using a secure signature.
Permanently store this exception
Confirm Security Exception
Cancel





wolfssl_tcp_client

wolfSSL TCP 客户端演示。

说明

该配置演示如何创建互联网客户端，该客户端使用 MPLAB Harmony TCP API 来创建到 Web 服务器的 TCP/IP 连接。该连接可以为明文，也可以使用 SSL 来加密与 wolfSSL 的连接。该演示可以使用 IPv4 或 IPv6。

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 wolfSSL TCP 客户端演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wolfssl_tcp_client.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/wolfssl_tcp_client`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wolfssl_tcp_client.X`	`<install-dir>/apps/tcpip/wolfssl_tcp_client/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 wolfSSL TCP 客户端。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 wolfSSL TCP 客户端。
pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 wolfSSL TCP 客户端。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 wolfSSL TCP 客户端。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

要使用该演示，需要将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。在演示运行时，它会在 USB 总线上创建一个 USB CDC 设备。通过标准终端程序连接该设备，将波特率设置为 921,600 波特，并且设备接收到一个有效 IP 地址之后，就可以执行该演示。

该演示中具有 3 个可通过串行端口使用的命令：

- `openurl <url>`——<url> 参数必须是完整形式的 URL ；例如， http://www.microchip.com/
- `ipmode <mode>`——<mode> 参数用于选择 IP 版本。 0—— 任意 IP 版本， 4—— 仅 IPv4， 6—— 仅 IPv6
- `stats`—— 输出上一次运行 openurl 的统计数据。诸如连接的每个阶段耗用多长时间，以及传输了多少字节之类的统计数据。

在输入 `openurl` 命令之后，演示会进行 DNS 查询，然后打开到所请求 URL 的连接，并执行简单的 HTTP PUT 命令。结果会被发送到串行端口。如果指定了 https URL，则该连接会先进行 SSL 协商，之后再发送 HTTP PUT 命令。

如果 ipmode 设置为 0 （任意），则演示会优先使用 IPv6 而不是 IPv4，这意味着它会先查找 IPv6 地址，之后才会查找 IPv4 地址。

wolfssl_tcp_server

wolfSSL TCP 服务器演示。

说明

该配置演示如何创建使用明文（TCP 端口 80）和密文（TCP 端口 443）工作的简单互联网 Web 服务器。如果使能了 IPv6，则演示也会在 IPv6 上提供这两种类型的连接服务。该 Web 服务器只会向所有 Web 客户端供应一个含有文本 “Nothing Here” 的网页。

Wi-Fi 演示矩阵

更多信息请参见 Wi-Fi 演示配置矩阵。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 wolfSSL TCP 客户端演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `wolfssl_tcp_server.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tcpip/wolfssl_tcp_server`。

警告

在使用 Microchip Harmony 配置器 （MHC）时，必须小心在生成代码时 *不要* 擦除 `system_init.c` 文件中的 USB 描述符。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`wolfssl_tcp_server.X`	`<install-dir>/apps/tcpip/wolfssl_tcp_server/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_eth_sk	pic32mx_eth_sk	在 PIC32 以太网入门工具包上演示 wolfSSL TCP 服务器。
pic32mx_eth_sk2	pic32mx_eth_sk2	在 PIC32 以太网入门工具包 II 上演示 wolfSSL TCP 服务器。

pic32mz_ec_sk	pic32mz_ec_sk	在 PIC32MZ EC 入门工具包上演示 wolfSSL TCP 服务器。
pic32mz_ef_sk	pic32mz_ef_sk	在 PIC32MZ EF 入门工具包上演示 wolfSSL TCP 服务器。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 以太网入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 以太网入门工具包 II

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

本部分介绍如何编译并运行演示。

说明

该演示不会通过串行端口提供任何附加的功能；但是，可以检查当前的 IP。在通过 DHCP 向演示分配有效 IP 之后，它会立即准备好供应网页。使用任意 Web 浏览器 （如 Chrome、 Internet Explorer 和 Firefox 等），通过 http:// 或 https:// 协议连接 Web 服务器。

可以将 USB 电缆连接到所用入门工具包背面的 micro-B USB 连接器。这会在 USB 总线上创建一个 USB CDC 设备。为了与软件进行通信，通过标准终端程序连接该设备，并将波特率设置为 921,600 波特。

测试应用程序

测试应用程序演示如何使用 MPLAB Harmony 测试套件和预定义的测试库。

简介

测试应用程序帮助

说明

本帮助文件包含关于 MPLAB Harmony 测试应用程序 （包含在 MPLAB Harmony 分发包中）的说明和关联信息。

应用程序

介绍如何运行测试应用程序。

test_sample

MPLAB Harmony 测试示例演示如何使用测试套件来验证示例库模块。

说明

测试套件控制每个测试以及它所测试的库的初始化和执行。测试套件会累计由测试传送给它的结果，并确定整体的通过或失败结果。如果支持并配置了调试系统输出 （该演示中不支持），这些结果会以文本形式在所用终端显示屏上显示。否则，正如该应用程序所演示的，必须使用调试器来确定结果。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了测试示例应用程序的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `test_sample.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/tests/test_sample`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`test_sample.X`	`<install-dir>/apps/tests/test_sample/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
e16mx795f512_freertos_interrupt	无。	该配置在 Explorer 16 开发板与 PIC32MX795F512L PIM 上运行。它会在 FreeRTOS 线程中执行所包含的测试，并在中断驱动模式下运行关联的库。
e16mx795f512_freertos_polled	无。	该配置在 Explorer 16 开发板与 PIC32MX795F512L PIM 上运行。它会在 FreeRTOS 线程中执行所包含的测试，并在另一个线程中在查询模式下运行关联的库。

e16mx795f512_interrupt	无。	该配置在 Explorer 16 开发板与 PIC32MX795F512L PIM 上运行。它会在裸机（无 RTOS）“超级”循环环境中执行所包含的测试，并在中断驱动模式下运行关联的库。
e16mx795f512_polled	无。	该配置在 Explorer 16 开发板与 PIC32MX795F512L PIM 上运行。它会在裸机（无 RTOS）“超级”循环环境中执行所包含的测试和所测试的库。
pic32mz_ec_sk_freertos_polled	无。	该配置在 PIC32MZ EC 入门工具包上运行。它会在 FreeRTOS 线程中执行所包含的测试，并在另一个线程中在查询模式下运行关联的库。
pic32mz_ec_sk_freertos_interrupt	无。	该配置在 PIC32MZ EC 入门工具包上运行。它会在裸机 （无 RTOS）“超级”循环环境中执行所包含的测试，并在中断驱动模式下运行关联的库。
pic32mz_ec_sk_interrupt	无。	该配置在 PIC32MZ EC 入门工具包上运行。它会在裸机 （无 RTOS）“超级”循环环境中执行所包含的测试，并在中断驱动模式下运行关联的库。
pic32mz_ec_sk_polled	无。	该配置在 PIC32MZ EC 入门工具包上运行。它会在裸机 （无 RTOS）“超级”循环环境中执行所包含的测试和所测试的库。
pic32mz_ef_sk_freertos_polled	无。	该配置在 PIC32MZ EF 入门工具包上运行。它会在 FreeRTOS 线程中执行所包含的测试，并在另一个线程中在查询模式下运行关联的库。
pic32mz_ef_sk_freertos_interrupt	无。	该配置在 PIC32MZ EF 入门工具包上运行。它会在裸机 （无 RTOS）“超级”循环环境中执行所包含的测试，并在中断驱动模式下运行关联的库。
pic32mz_ef_sk_interrupt	无。	该配置在 PIC32MZ EF 入门工具包上运行。它会在裸机 （无 RTOS）“超级”循环环境中执行所包含的测试和所测试的库。
pic32mz_ef_sk_polled	无。	该配置在 PIC32MZ EF 入门工具包上运行。它会在裸机 （无 RTOS）“超级”循环环境中执行所包含的测试和所测试的库。

配置硬件

介绍如何配置支持的硬件。

说明

Explorer 16 开发板与 PIC32MX795F512L PIM

在将 PIC32MX795F512L PIM 连接到 Explorer 16 开发板之前，请确保处理器选择开关 （S2）处于 PIM 位置。

对于 PIM 本身，不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行应用程序

介绍如何编译并运行测试示例应用程序。

说明

要运行测试示例应用程序，请选择所需的配置，在调试模式下编译该应用程序，将它烧写到关联硬件中，并在调试器的控制下执行它。

在演示运行时，测试套件库会累计由测试提供给它的结果。如果支持并配置了系统调试输出，则会以文本形式在所用调试终端显示屏上显示结果。但是，根据所需的测试或所用的硬件，可能无法支持调试输出方法，该演示中包含的所有配置都不支持任何调试输出方法。因此，需要利用调试器来分析 TEST_HARNESS_DATA 结构，以确定测试的结果。当所有测试均完成时 （或发生失败时），测试套件会执行硬编码断点并停止。此时，您可以使用调试器来获取结果。

当测试套件进入其空闲状态时，该结构的以下成员就会给出整体结果：

- testsCount—— 这是已执行的测试的总数
- testsPassed—— 这是未报告任何子测试失败，从而产生整体通过结果的测试的数量
- result—— 这是最后的结果，如果所有测试中的所有子测试均通过，则它为 true （1）；否则，它为 false （0）。

USB 演示

本主题介绍 USB 演示。

简介

USB 库演示应用程序帮助

说明

该分发包包含一系列与 USB 相关的固件项目，用于演示 MPLAB Harmony USB 协议栈的功能。本部分介绍硬件要求，以及在 Microchip 演示和开发板上运行这些固件项目的过程。

要了解关于 MPLAB Harmony USB 协议栈、配置 USB 协议栈和 USB 协议栈提供的 API 的更多信息，请参见 USB 库文档。

程序存储器、数据存储器和协议栈组件存储器

关于重要的存储器信息，请参见 USB 设备协议栈演示应用程序的程序和数据存储器要求和 USB 设备协议栈组件的存储器要求。

笔式驱动器测试

关于在 USB 闪存设备上进行的测试的信息，请参见 USB MSD 主机 USB 笔式驱动器测试。

USB 设备协议栈演示应用程序的程序和数据存储器要求

提供关于程序和数据存储器要求以及笔式驱动器测试规范的信息。

说明

使用 -O1 优化时的程序存储器和数据存储器要求

下表列出了 USB 设备协议栈演示应用程序的程序存储器和数据存储器要求。所有大小数字都以字节为单位。这些演示应用程序是使用 MPLAB XC32 C/C++ 编译器 v1.40 以 -O1 优化编译的。

Demonstration Name		Program Memory Components			Data Memory Components	
-O1 Optimization		USB Stack	Other Drivers	System and Application	USB Stack	Others
cdc_com_port_single	PIC32MX	14048	0	5880	262	718
	PIC32MZ	18924	4328	11632	318	1410
cdc_com_port_dual	PIC32MX	13980	0	5776	262	1250
	PIC32MZ	18856	4328	10908	318	2166
cdc_serial_emulator	PIC32MX	13976	9100	5916	262	858
	PIC32MZ	N/A	N/A	N/A	N/A	N/A
cdc_serial_emulator_msd	PIC32MX	20380	12560	39540	438	1950
	PIC32MZ	NA	NA	NA	NA	NA
cdc_msd_basic	PIC32MX	20392	3460	39444	494	1810
	PIC32MZ	25264	7788	45404	494	20438
hid_basic	PIC32MX	13988	0	5440	263	601
	PIC32MZ	18764	4328	10460	319	893
hid_joystick	PIC32MX	13432	0	5332	263	485
	PIC32MZ	18208	4328	10368	319	777
hid_keyboard	PIC32MX	14016	0	6196	263	581
	PIC32MZ	18792	4328	11048	319	873
hid_mouse	PIC32MX	13460	0	5972	263	497
	PIC32MZ	18236	4328	10964	319	793
hid_msd_basic	PIC32MX	20352	3460	39172	495	1861
	PIC32MZ	25232	7788	44216	495	20445
msd_basic	PIC32MX	17848	3460	38108	492	1348
	PIC32MZ	22632	7788	43996	492	20068
vendor	PIC32MX	12560	0	5660	324	560
	PIC32MZ	17356	4328	12080	324	1748

注： msd_basic、 cdc_msd_basic 和 hid_msd_basic 演示使用 PIC32 程序闪存作为 MSD 存储介质。对于这些演示示例，PIC32MX 和 PIC32MZ 单片机的数据存储器要求有些区别，这是因为应用程序演示缓冲区的大小等于 PIC32 单片机的擦除页大小。在 PIC32MX795F512L 上，该大小为 4096 字节。在 PIC32MZ2048ECH144 上，擦除页大小为 16 KB。

使用 -Os 优化时的程序存储器和数据存储器要求

下表列出了 USB 设备协议栈演示应用程序的程序存储器和数据存储器要求。所有大小数字都以字节为单位。这些演示应用程序是使用 MPLAB XC32 C/C++ 编译器 v1.40 以 -Os 优化编译的。

Demonstration Name		Program Memory Components			Data Memory Components	
Optimization -Os		USB Stack	Other Drivers	System and Application	USB Stack	Others
cdc_com_port_single	PIC32MX	12556	0	5656	262	718
	PIC32MZ	16840	4144	11076	318	1410
cdc_com_port_dual	PIC32MX	12548	0	5620	262	1250
	PIC32MZ	16832	4144	10448	318	2166
cdc_serial_emulator	PIC32MX	12504	7744	5880	262	858
	PIC32MZ	N/A	N/A	N/A	N/A	N/A
cdc_serial_emulator_msd	PIC32MX	20380	12560	35080	438	1950
	PIC32MZ	N/A	N/A	N/A	N/A	N/A
cdc_msd_basic	PIC32MX	17896	2884	39232	494	1810
	PIC32MZ	22172	7012	44868	494	20438
hid_basic	PIC32MX	12656	0	5344	263	601
	PIC32MZ	16784	4144	10060	319	893
hid_joystick	PIC32MX	12144	0	5304	263	485
	PIC32MZ	16272	4144	10040	319	777
hid_keyboard	PIC32MX	12664	0	6060	263	581
	PIC32MZ	16792	4144	10604	319	873
hid_mouse	PIC32MX	12152	0	5820	263	497
	PIC32MZ	16280	4144	10512	319	793
hid_msd_basic	PIC32MX	18060	2884	39068	495	1861
	PIC32MZ	20352	7012	45788	495	20445
msd_basic	PIC32MX	15776	2884	38044	492	1348
	PIC32MZ	19090	7012	44426	492	20068
vendor	PIC32MX	11452	0	5540	324	560
	PIC32MZ	15584	4144	10948	324	1748

USB 设备协议栈组件的存储器要求

提供存储器要求。

说明

下表列出了 MPLAB Harmony USB 设备协议栈中的各个组件的程序和数据存储器要求。

设备协议栈组件	程序存储器	数据存储器
设备层	5688	184
CDC 功能驱动程序	2420	64 + (36 * 队列大小)
MSD 功能驱动程序	5352	217
HID 功能驱动程序	2376	40 + (36 * 队列大小)
供应商	912	8 + (36 * 队列大小)
PIC32MX USB 驱动程序	5636	144 + (32 * 端点数量)
PIC32MZ USB 驱动程序	10244	192 + (32 * 端点数量)

注：

1. 单个实例的存储器要求 （以字节为单位）。
2. 测量的是 MPLAB Harmony 中的 USB 设备协议栈组件的大小。
3. 数据存储器不包括函数调用堆栈存储器大小。

USB MSD 主机 USB 笔式驱动器测试

提供笔式驱动器测试规范。

说明

USB MSD 主机 USB 笔式驱动器测试

下表列出了一些市售 USB 笔式驱动器，它们已经过测试，可以使用 MSD 主机驱动程序成功地在 MPLAB Harmony USB 主机中枚举。请注意，如果您使用的 USB 笔式驱动器未包含在该表中，则表示该 USB 笔式驱动器尚未使用 MSD 主机驱动程序进行过测试。但是，该 USB 笔式驱动器仍有可能可使用 MSD 主机驱动程序。该表中的一些 USB 笔式驱动器没有制造商或型号数据。 USB 笔式驱动器是使用 msd_basic USB 主机演示在最新版本的 MPLAB Harmony USB 主机协议栈中测试的。

VID	PID	制造商	型号 / 驱动器容量
0x1B1C	0x1A0F	Corsair Components	Flash Voyager Go 8 GB
0x03F0	0x0AB7	Hewlett-Packard	64 GB
0xABCD	0x1234	Microchip Technology Inc.	4 GB
0x125F	0xCB10	Adata	Dashdrive UV100 8 GB
0x8644	0x8003	Verico	T Series 16 GB
0x8564	0x1000	Transcend	USB 3.0 32 GB
0x0951	0x16A7	Dell	Kingston Technology 16 GB
0x0718	0x0704	Imation	16 GB Pen Drive
0x048D	0x1168	iBall	Jaldi 16 GB Pen Drive
0x058F	0x6366	Alcor	Micro AXL 32 GB
0x154B	0x005B	PNY	Cube 16 GB
0x0930	0x6544	Toshiba	Hatabusa Pen Drive 8 GB
0x058F	0x6387	Alcor	ZipMem 16 GB
0x090C	0x1000	Silicon Motion Inc.	Axl 8GB
0x18A5	0x0245	Verbatim	Store N Go Audio USB 8 GB
0x05DC	0xC75C	Lexar	USB Pen Drive 8 GB
0x1005	0xb113	Apacer	8 GB (AH233)
0x054C	0x06B0	Sony	8 GB
0x054C	0x0862	Sony	Micro Vault USM-V 8 GB
0x0781	0x557c	SanDisk	8 GB
0x1E4E	0x3257	Etron	iBall 16 GB
0x1EC9	0x0101	Moserbaer	Swivel 16 GB Pen Drive
0x0BDA	0x0109	SanDisk	Standard A and Mini-B connector 16 GB
0x1908	0x1320	ZBEL	Wrist Band Flash Drive 4 GB
0x0951	0x1665	Kingston	Data Traveler SE9 16 GB

USB HID 主机键盘和鼠标测试

说明

下表列出了一些市售 USB 键盘和鼠标，它们已经过测试，可以使用 HID 主机驱动程序成功地在 MPLAB Harmony USB 主机中枚举。请注意，如果您使用的USB HID 设备未包含在该表中，则表示该USB HID 设备尚未经过测试，但仍有可能可以使用HID主机驱动程序。

Type	Manufacturer	VID	PID	Device Details
Keyboard	Microsoft	0x045E	0x07B9	Microsoft wired 200.
	ProHT	0x1A2C	0x0C21	110 keys.
	HP	0x04D9	0x1702	K1500 standard keyboard.
	Zebronics	0x1A2C	0x0027	Standard keyboard.
	Logitech	0x046D	0xC31D	112 keys.
	Gear Head	0x04D9	0x2BA0	82 key, mini-USB keyboard plus integrated mouse.
	Intex	0x1C4F	0x0002	122 keys.
Mouse	HP	0x0461	0x4D0F	Three button mouse.
	Intex	0x1BCF	0x0007	Three button mouse.
	Anker	0x1BCF	0x0005	Five button mouse.
	TeraByte	0x10C4	0x8103	Three button mouse.
	Kensington	0x1BCF	0x0002	Five button mouse.
	Logitech	0x046D	0xC069	Five button mouse.
	Microsoft	0x045E	0x0797	Optical mouse 200, three buttons.
	Logitech	0x046D	0xC246	Nine button mouse.
	HP	0x03F0	0x0941	Three button mouse.
	Lenovo	0x1BCF	0x000A	Five button mouse.

演示应用程序配置

本主题介绍可用的 USB 演示项目配置。

说明

可用的 USB 演示应用程序 MPLAB X IDE 项目支持多种配置。选择这些配置可以让演示项目在不同的 PIC32 单片机和开发板上运行。有以下项目配置可供使用：

配置名称	说明
pic32mx_usb_sk2_int_dyn	选择该配置会将演示应用程序设置为在 PIC32 USB 入门工具包 II 开发板与 PIC32MX795F512L 单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mx_usb_sk2_poll_dyn	选择该配置会将演示应用程序设置为在 PIC32 USB 入门工具包 II 开发板与 PIC32MX795F512L 单片机上运行。 USB 协议栈会被配置为查询模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mx_usb_sk3_int_dyn	选择该配置会将演示应用程序设置为在 PIC32 USB 入门工具包 III 开发板与 PIC32MX470F512L 单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mx_usb_sk2_int_sta	选择该配置会将演示应用程序设置为在 PIC32 USB 入门工具包 II 开发板与 PIC32MX795F512L 单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为静态操作模式。
pic32mz_bt_audio_int_dyn	选择该配置会将演示应用程序设置为在 PIC32 蓝牙音频开发工具包与 PIC32MZ2048ECH144 单片机上运行。 USB 设备协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mx_bt_sk_int_dyn	选择该配置会将演示应用程序设置为在 PIC32 蓝牙入门工具包开发板与 PIC32MX270F256D 单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mz_da_sk_int_dyn	选择该配置会将演示应用程序设置为在 PIC32MZ DA 入门工具包开发板与 PIC32MZ2064DAB288 单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。

pic32mz_ec_sk_int_dyn	选择该配置会将演示应用程序设置为在PIC32MZ EC入门工具包开发板与PIC32MZ2048ECH144单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mz_ec_sk_poll_dyn	选择该配置会将演示应用程序设置为在 PIC32MZ EC 入门工具包开发板与 PIC32MC2048ECH144 单片机上运行。 USB 协议栈会被配置为查询模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mz_ec_sk_meb2_int_dyn	选择该配置会将演示应用程序设置为在连接到 MEB II 的 PIC32MZ EC 入门工具包与 PIC32MZ2048ECH144 单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mz_ef_sk_int_dyn	选择该配置会将演示应用程序设置为在 PIC32MZ EF 入门工具包与 PIC32MZ2048EFM144 单片机上运行。 USB 协议栈会被配置为中断模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mz_ef_sk_poll_dyn	选择该配置会将演示应用程序设置为在PIC32MZ EF入门工具包开发板与PIC32MZ2048EFM144单片机上运行。 USB 协议栈会被配置为查询模式操作， USB 驱动程序会被配置为动态操作模式。
pic32mx795_pim_e16_int_dyn	选择该配置会将演示应用程序设置为在Explorer 16开发板以及PIC32MX795F512L单片机接插模块和 USB PICtail Plus 子板上运行。USB 协议栈会被配置为中断模式操作，USB 驱动程序会被配置为动态操作模式。
pic32mx460_pim_e16_int_dyn	选择该配置会将演示应用程序设置为在Explorer 16开发板以及PIC32MX460F512L单片机接插模块和 USB PICtail Plus 子板上运行。USB 协议栈会被配置为中断模式操作，USB 驱动程序会被配置为动态操作模式。

下图说明了如何在 MPLAB X IDE 中选择配置。



或者，也可以在 Project Properties 中选择活动配置。

USB 设备演示矩阵

下表列出了各种配置在可用 USB 设备演示应用程序中的可用性。**绿色**表示支持。红色表示不支持。

Configurations / Demo Apps	pic32mx_usb_sk2_int_dyn	pic32mz_ec_sk_int_dyn	pic32mx795_pim_e16_int_dyn	pic32mx_bt_sk_int_dyn	pic32mz_ec_sk_meb2_int_dyn	pic32mx_usb_sk2_poll_dyn	pic32mz_ec_sk_pol_dyn	pic32mx460_pim_e16_int_dyn	pic32mx_usb_sk2_int_sta	pic32mx_usb_sk3_int_dyn	pic32mz_bt_audio_int_dyn	pic32mz_ef_sk_poll_dyn	pic32mz_ef_sk_int_dyn	pic32mx_125_sk_int_dyn	chipkit_wf32	chipkit_wifire
cdc_com_port_single																
cdc_com_port_dual																
cdc_serial_emulator																
hid_basic																
hid_mouse																
hid_keyboard																
hid_joystick																
msd_basic																
vendor																
hid_msd_basic																
cdc_serial_emulator_msd																
cdc_msd_basic																
msd_sdcard																

USB 主机演示矩阵

下表列出了各种配置在可用 USB 主机演示应用程序中的可用性。绿色表示支持。红色表示不支持。

Configurations / Demo Apps	pic32mx_usb_sk2_int_dyn	pic32mz_ec_sk_int_dyn	pic32mz_ec_sk_meb2_int_dyn	pic32mz_ef_sk_int_dyn	pic32mz_ef_sk_meb2_int_dyn	pic32mx795_pim_e16_int_dyn	chipkit_wf32	chipkit_wifire
cdc_basic								
msd_basic								
cdc_msd								
hid_basic_mouse								
audio_speaker								
hub_msd								
hub_cdc_hid								
hid_basic_keyboard								

演示

USB 演示分为 USB 设备协议栈和 USB 主机协议栈演示。

设备

本部分介绍 USB 设备演示。

说明

MPLAB Harmony USB 设备协议栈演示应用程序使用开发板上的 LED 来指示设备的 USB 状态。下表详细说明了开发板特定 LED 以及它们在有效时指示的 USB 设备状态。所有 USB 设备协议栈演示应用程序都实现了这种指示方案。

USB 设备状态和 LED 指示

演示板	复位状态	配置状态	暂停状态
Explorer 16 开发板和 PIM	D3 和 D4	D5	D4 和 D5
PIC32 USB 入门工具包 II	LED1 和 LED2	LED3	LED2 和 LED3
PIC32MZ 嵌入式连接 （EC）入门工具包	LED1 和 LED2	LED3	LED2 和 LED3
PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包	LED1 和 LED2	LED3	LED2 和 LED3
PIC32 USB 入门工具包 III	LED1 和 LED2	LED3	LED2 和 LED3
PIC32 蓝牙入门工具包	红色 LED 和绿色 LED	蓝色 LED	绿色 LED 和蓝色 LED

cdc_com_port_dual

演示仿真两个串行 COM 端口的 USB CDC 设备 —— 一个环回到另一个。

说明

该演示应用程序会创建在 USB 主机个人计算机上枚举为两个串行端口的 USB CDC 设备。该应用程序演示 MPLAB Harmony USB 设备协议栈支持同一设备类的多个实例的能力。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB CDC 设备双 COM 端口演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_com_port_dual.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/cdc_com_port_dual`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_com_port_dual.X`	`<install-dir>/apps/usb/device/cdc_com_port_dual/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx460_pim_e16_int_dyn	pic32mx460_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX460F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mx_bt_sk_int_dyn	pic32mx_bt_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙入门工具包上运行配置为中断模式和动态操作的演示。

pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_int_sta	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以使用配置为静态编译的 USB 驱动程序在 PIC32 USB 入门工具包 II 上运行配置为中断模式和静态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32 蓝牙入门工具包

跳线 J8 应在引脚 2 和 3 之间短接或应完全开路。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX460F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路

运行演示

介绍如何编译并运行 CDC 双 COM 端口演示。

说明

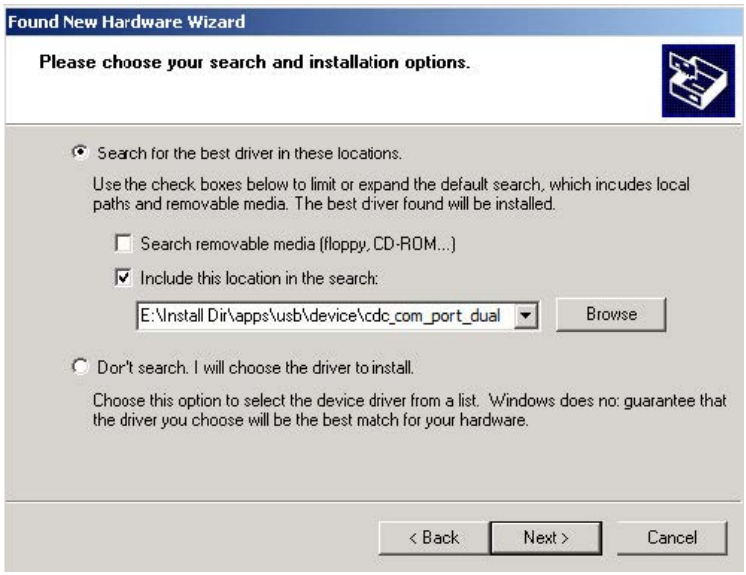
该演示让设备对于主机来说就像是两个串行 （COM）端口。执行以下操作来运行该演示：

1. 首先编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。

2. 将设备连接到主机。如果主机是一台个人计算机，并且这是第一次将该设备插入计算机中，则可能会提示您安装 `.inf` 文件。



3. 选择 “从列表或指定位置安装 （高级）” 选项。指定 `<install-dir>/apps/usb/device/cdc_com_port_dual/inf` 目录。



4. 成功安装设备之后，打开终端程序 （如 HyperTerminal）的两个实例。为其中每个终端实例选择适当的 COM 端口。
5. 要运行演示，请在一个终端窗口中输入字符或字符串。相同的字符或字符串会显示在第二个终端窗口中。同样，在第二个窗口中输入的任何字符都会显示在第一个窗口中。

注： 一些终端程序（如 HyperTerminal）要求用户先单击断开连接按钮，之后才能从计算机中移除设备。如果不这样做，则为了重新连接设备，可能必须重新关闭并打开程序。

cdc_com_port_single

演示仿真一个串行 COM 端口的 USB CDC 设备。

说明

该演示应用程序会创建在主机个人计算机上枚举为单个 COM 端口的 USB CDC 设备。该应用程序演示了 USB 设备和个人计算机主机之间的双向通信。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB CDC 设备单 COM 端口演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_com_port_single.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/cdc_com_port_single`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_com_port_single.X`	`<install-dir>/apps/usb/device/cdc_com_port_single/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx460_pim_e16_int_dyn	pic32mx460_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX460F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mx_usb_sk2_poll_dyn	pic32mx_bt_sk	选择该MPLAB X IDE项目配置可以在PIC32 USB入门工具包II上使用USB设备协议栈运行配置为查询模式和动态操作的演示。
pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_int_sta	pic32mx_usb_sk2	选择该MPLAB X IDE项目配置可以在PIC32 USB入门工具包II上使用USB驱动程序运行配置为中断模式和静态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_da_sk_int_dyn	pic32mz_da_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 图形（DA）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_poll_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上使用 USB 设备协议栈运行配置为查询模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

pic32mz_ef_sk_poll_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上使用 USB 设备协议栈运行配置为查询模式和动态操作的演示。
pic32mx_125_sk_int_dyn	pic32mx_125_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MX1/2/5 入门工具包上使用 USB 设备协议栈运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX460F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路

运行演示

介绍如何编译并运行 CDC 单 COM 端口演示。

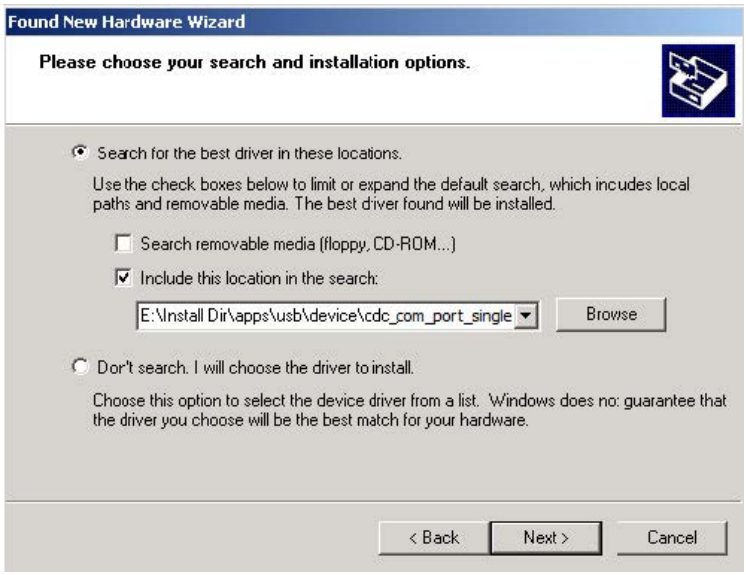
说明

该演示让设备对于主机来说就像是一个串行 （COM）端口。执行以下操作来运行该演示：

1. 首先编译并对目标器件编程。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。
2. 将设备连接到主机。如果主机是一台个人计算机，并且这是第一次将该设备插入计算机中，则可能会提示您安装 `.inf` 文件。



3. 选择 “从列表或指定位置安装 （高级）” 选项。指定 `<install-dir>/apps/usb/device/cdc_com_port_single/inf` 目录。



4. 成功安装设备之后，打开一个终端程序 （如 HyperTerminal）并选择适当的 COM 端口。在大多数计算机上，它将为 COM5 或更高编号的端口。将通信属性设置为 9600 波特率， 1 个停止位，无奇偶校验，流量控制设置为无。
5. 连接设备之后，有两种方式可以运行该示例项目：

a) 在终端窗口中输入某个键会导致所连接的设备回显下一个字母。因此，如果按下字母 b，设备将回显 c。

b) 如果按下按钮，设备会在终端窗口中回显 “PUSH BUTTON PRESSED”（按钮被按下）。

下表列出了对于不同演示板需要按下的开关按钮。

演示板	按钮
PIC32 USB 入门工具包 II PIC32 USB 入门工具包 III PIC32MZ 图形 （DA）入门工具包 PIC32MZ 嵌入式连接 （EC）入门工具包 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包	SW1
Explorer 16 开发板	S3

注： 一些终端程序（如 HyperTerminal）要求用户先单击断开连接按钮，之后才能从计算机中移除设备。如果不这样做，则为了重新连接设备，可能必须重新关闭并打开程序。

cdc_msd_basic

演示仿真 COM 端口和闪存驱动器的复合 USB 设备。

说明

该演示应用程序会创建同时枚举为 COM 端口和闪存驱动器的复合 USB 设备。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB CDC MSD 复合设备演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_msd_basic.X` 项目，然后选择所需的配置。以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/cdc_msd_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_msd_basic.X`	`<install-dir>/apps/usb/device/cdc_msd_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 USB CDC MSD 复合设备演示。

说明

该演示应用程序会创建同时作为 CDC 和 MSD 设备工作的复合 USB 设备。该应用程序将 cdc_com_port_single 和 msd_basic 演示应用程序的功能组合到一个设备中。

关于运用CDC和MSD设备功能的详细信息，请分别参见cdc_com_port_single演示的运行演示部分和msd_basic演示的运行演示部分。

cdc_serial_emulator

该应用程序演示如何在实现 USB 转串行转换器中使用 CDC 设备类。

说明

该应用程序演示如何在实现 USB 转串行转换器中使用 CDC 设备类。该应用程序会在个人计算机上枚举一个 COM 端口。通过 CDC USB 接口接收到的数据会被转发到 UART。在 UART 上接收到的数据会被转发到 CDC USB 接口。它仿真一个 USB 转串行转换器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB CDC 设备 USB 转串行演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_serial_emulator.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/cdc_serial_emulator`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_serial_emulator.X`	`<install-dir>/apps/usb/device/cdc_serial_emulator/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16_int_dyn	pic32mx795_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX795F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ EF 入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MX795F512L PIM

需要移除跳线 J10。跳线 J1 和 J2 应连接到位置 1 和 2。该 PIM 应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX795F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路
- 跳线 J1 应在位置 1 和 2 之间短接。该配置仅适用于 PIC32MX795F512L USB CAN PIM （MA320003），不适用于 PIC32MX795F512L USB PIM （MA320002）。
- 跳线 J2 应在位置 1 和 2 之间短接。该配置仅适用于 PIC32MX795F512L USB CAN PIM （MA320003），不适用于 PIC32MX795F512L USB PIM （MA320002）。

运行演示

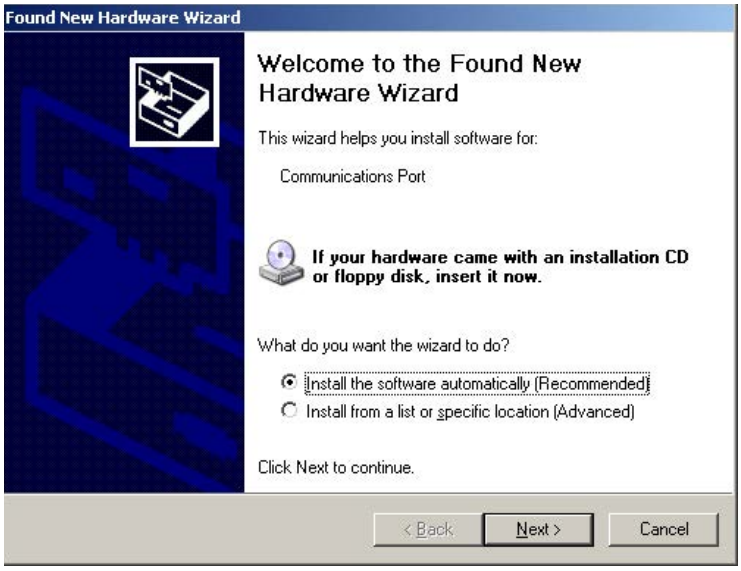
介绍如何编译并运行 CDC 串行仿真器演示。关于演示限制，请参见发行说明 （如适用）。

说明

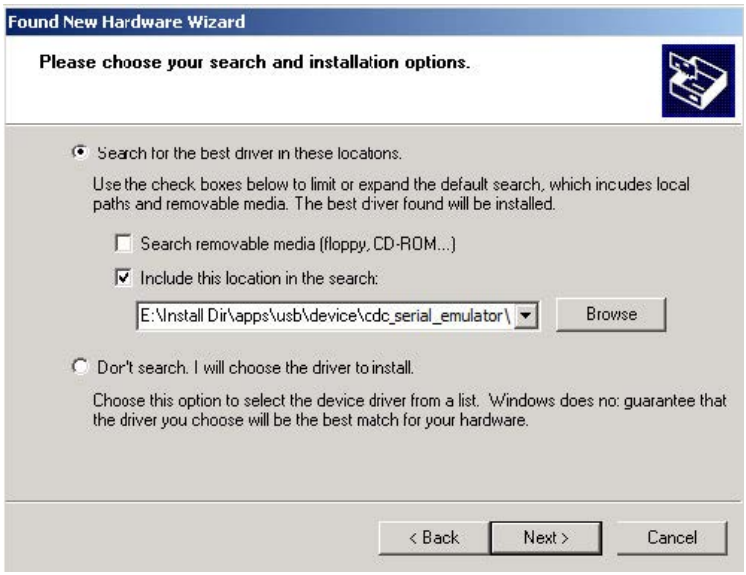
该应用程序演示如何在实现 USB 转串行转换器中使用 CDC 设备类。该应用程序会在个人计算机上枚举一个 COM 端口。通过 CDC USB 接口接收到的数据会被转发到 UART。在 UART 上接收到的数据会被转发到 CDC USB 接口。它仿真一个 USB 转串行转换器。

1. 在 MPLAB X IDE 中打开项目，并选择所需的配置。
2. 编译代码并对器件编程。
3. 根据所用硬件，执行以下操作之一：

- 如果使用的是 Explorer 16 开发板，则将 USB PICtail Plus 子板上的 mini-B 设备连接器连接到个人计算机
- 如果使用的是 PIC32MZ EF 入门工具包，则将 micro-USB 设备连接器连接到个人计算机



4. 选择 “从列表或指定位置安装 （高级）” 选项。指定 `<install-dir>/apps/usb/device/cdc_serial_emulator/inf` 目录。

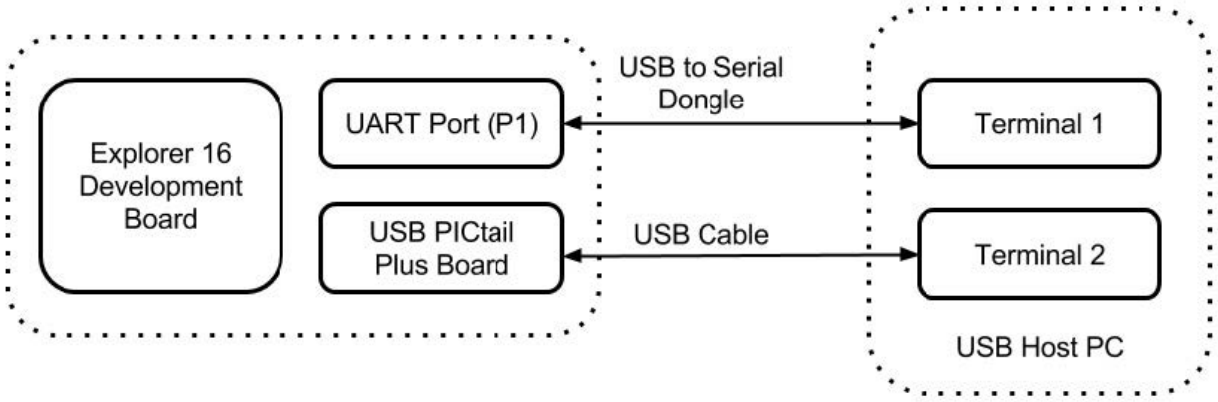


当枚举成功时，以下 LED 会点亮：

- Explorer 16 开发板：LED D5
- PIC32MZ EF 入门工具包：LED3

5. 打开所选的终端仿真程序，并选择枚举的 USB COM 端口。
6. 将 USB 转串行转换器连接到同一台个人计算机。
7. 打开终端仿真程序的另一个实例，并选择 USB 转串行转换器。
8. 将 USB 转串行转换器的串行连接器连接到 Explorer 16 开发板上的 UART 连接器 （P1）。
9. 选择 9600 波特率， 1 个停止位，无奇偶校验，并同时打开两个终端仿真程序。

其设置应类似于下图：



在终端 1 程序中输入的任何文本都会在终端 2 上回显，反之亦然。

cdc_serial_emulator_msd

演示与 MSD 类组合的 USB 转串行转换器。

说明

该演示应用程序会创建组合了 cdc_serial_emulator 和 msd_basic 演示应用程序功能的 USB 设备。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了该演示应用程序的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 cdc_serial_emulator_msd.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/usb/cdc_serial_emulator_msd。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
cdc_serial_emulator_msd.X	<install-dir>/apps/usb/device/cdc_serial_emulator_msd/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16_int_dyn	pic32mx795_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX795F512L 接插模块（PIM）和 USB PICtail Plus 子板。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MX795F512L PIM

需要移除跳线 J10。跳线 J1 和 J2 应连接到位置 1 和 2。该 PIM 应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX795F512L PIM 上：

- 跳线 J10 保持开路。
- J9 中的所有跳线保持开路
- 跳线 J1 应在位置 1 和 2 之间短接。
- 跳线 J2 应在位置 1 和 2 之间短接。

运行演示

介绍如何编译并运行演示。

说明

该演示作为一个复合 USB 设备工作，它组合了 cdc_serial_emulator 和 msd_basic 演示应用程序创建的设备的功能。关于运用 CDC 和 MSD 功能的详细信息，请分别参见 cdc_serial_emulator 演示的运行演示部分和 msd_basic 演示的运行演示部分。

hid_basic

该演示应用程序会创建可通过基于个人计算机的实用程序控制的定制 HID 设备。

说明

该应用程序会创建可通过基于个人计算机的实用程序控制的定制 HID 设备。该设备允许 USB 主机实用程序控制电路板上的 LED 和查询开关的状态。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB HID 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hid_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/hid_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hid_basic.X`	`<install-dir>/apps/usb/device/hid_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx460_pim_e16_int_dyn	pic32mx460_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX460F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX460F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路

运行演示

介绍如何编译并运行 HID 基本演示。

说明

该演示使用选定的硬件平台作为 HID 类 USB 设备，但使用 HID 类来进行通用 I/O 操作。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。

通常情况下， HID 类用于实现人机接口产品，如鼠标和键盘。但是， HID 协议是非常灵活的，可以改造并用于向 / 从 USB 设备发送和接收通用数据。使用 HID 类进行通用 I/O 操作是很有利的，因为它不需要任何类型的定制驱动程序安装过程。HID 类驱动程序已经随常用操作系统提供和分发。因此，在将 HID 类设备插入典型计算机系统时，不需要用户安装驱动程序，安装是完全自动的。

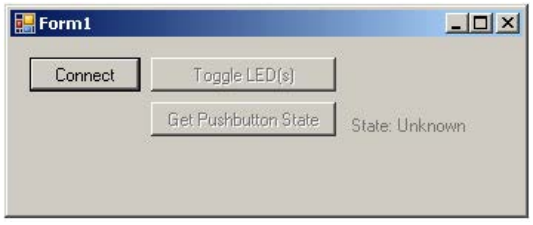
HID 设备主要通过一个中断 IN 端点和一个中断 OUT 端点进行通信。在大多数应用中，这实际上会将全速 HID 设备的最大可实现带宽限制为 64 KB/s 的 IN 通信和 64 KB/s 的 OUT 通信 （64 KB/s，但实际上是 “全双工”）。

`GenericHIDSimpleDemo.exe` 程序和关联的固件演示如何使用 HID 协议来进行基本的通用 USB 数据传输。

您需要先在计算机上安装 Microsoft® .NET Framework 版本 2.0 可再发行组件包（更高的版本可能可接受，但未经过测试），之后才能运行 `GenericHIDSimpleDemo.exe` 可执行文件。以 Visual Studio® .NET 语言编译的程序需要 .NET 可再发行组件包。可再发行组件包可以从 Microsoft 网站免费下载。 Windows Vista® 操作系统的用户不需要安装 .NET Framework，因为它会作为操作系统的一部分预装。

启动应用程序

要启动应用程序，只需双击 `<install-dir>\apps\usb\device\hid_basic\bin` 目录中的可执行文件 `GenericHIDSimpleDemo.exe`。此时会显示一个类似于下图的属性表：



注： 如果尝试启动应用程序时不是显示该窗口，而是显示错误消息，则很可能是因为 Microsoft .NET Framework 版本 2.0 可再发行组件包尚未安装。请安装它，然后重试。

发送 / 接收数据包

要开始向设备发送 / 接收数据包，必须先找到并连接设备。根据默认情况下的配置，应用程序会查找 VID = 0x04D8 和 PID = 0x003F 的 HID 类 USB 设备。用于该演示的固件项目中的设备描述符使用相同的 VID/PID。如果插入已烧写了正确的预编译 `.hex` 文件的 USB 设备，并单击 **Connect**，其他按钮会变为使能状态。如果单击 **Connect** 没有任何作用，则很可能是因为 USB 设备未连接或未烧写正确的固件。

单击 **Toggle LED(s)**（切换 LED）会向 HID 类 USB 外设设备发送单个通用数据包。数据会到达中断 OUT 端点。固件已被配置为接收该通用数据包，通过解析该数据包来查找 Toggle LED(s) 命令，并通过控制演示板上的 LED 作出适当的响应。

Get Pushbutton State （获取按钮状态）选项会通过 USB 向外设设备发送一个数据包 （发送到中断 OUT 端点），以请求当前按钮状态。固件会处理接收到的 Get Pushbutton State 命令，并根据按钮状态准备相应的响应数据包。

下表列出了为看到按钮状态变化需要在演示板上按下的按钮。

演示板	按钮
PIC32 USB 入门工具包 II PIC32 USB 入门工具包 III PIC32MZ 嵌入式连接 （EC）入门工具包 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包	SW1
Explorer 16 开发板	S3

hid_joystick

演示仿真游戏杆的 USB HID 设备。

说明

该演示应用程序会创建一个定制 HID 游戏杆。该应用程序仅用于演示如何创建游戏杆 HID 报告描述符，不能作为一个明确的最终解决方案。最终应用程序的要求可能需要修改报告描述符。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB HID 游戏杆演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hid_joystick.X` 项目，然后选择所需的配置。以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/hid_joystick`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hid_joystick.X`	`<install-dir>/apps/usb/device/hid_joystick/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx460_pim_e16_int_dyn	pic32mx460_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX460F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX460F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路

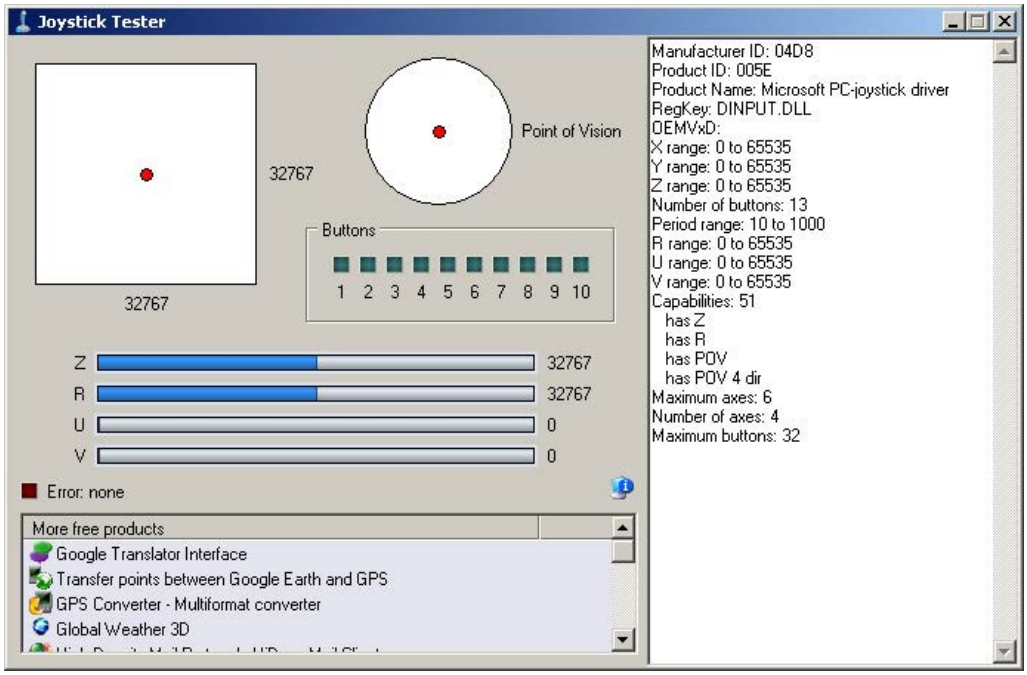
运行演示

介绍如何编译并运行 USB HID 游戏杆演示。

说明

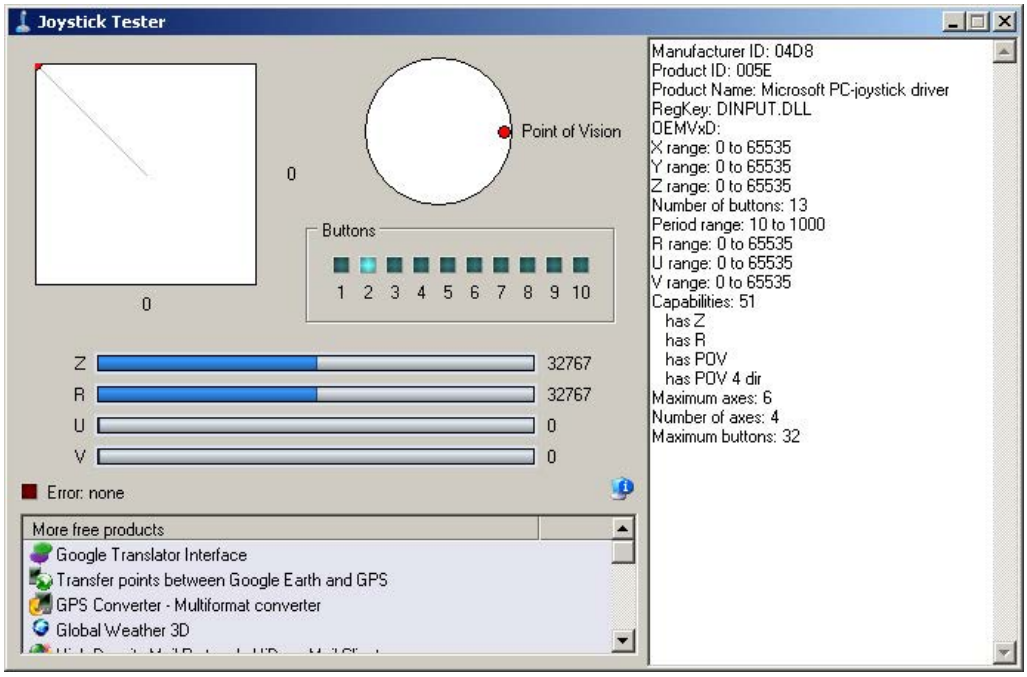
该演示使用选定的硬件平台作为 USB 游戏杆。基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。

要测试游戏杆功能，请导航到`<install-dir>/apps/usb/device/hid_joystick/bin`目录，然后打开`JoystickTester.exe`



按下按钮会导致设备：

- 指示 “x” 按钮已被按下，但不会指示其他信息
- 将顶帽开关移到 “东方” 位置
- 将 X 和 Y 坐标移动到其极值



下表列出了为仿真游戏杆需要在演示板上按下的按钮。

演示板	按钮
PIC32 USB 入门工具包 II PIC32 USB 入门工具包 III PIC32MZ 嵌入式连接 （EC）入门工具包 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包	SW1
Explorer 16 开发板	S3

hid_keyboard

演示仿真键盘的 USB HID 设备。

说明

该演示应用程序会创建一个通用 HID 键盘。按下电路板上的按键时，会仿真键盘按键按下。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB HID 键盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hid_keyboard.X` 项目，然后选择所需的配置。以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/hid_keyboard`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hid_keyboard.X`	`<install-dir>/apps/usb/device/hid_keyboard/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx460_pim_e16_int_dyn	pic32mx460_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX460F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX460F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路

运行演示

介绍如何编译并运行 USB HID 键盘演示。

说明

该演示使用选定的硬件平台作为 USB 键盘。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。

在按下按钮之前，请选择一个可安全地自由输入文本的窗口。按下演示板上的按钮会导致设备在屏幕上打印一个字符。

下表列出了为打印字符需要在演示板上按下的按钮。

演示板	按钮
PIC32 USB 入门工具包 II PIC32 USB 入门工具包 III PIC32MZ 嵌入式连接 （EC）入门工具包 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包	SW1
Explorer 16 开发板	S3

hid_mouse

演示仿真鼠标设备的 USB HID 设备。

说明

该演示应用程序会创建一个基于 USB HID 的双按钮鼠标设备。当连接时，该设备会通过以环形方式移动光标来仿真鼠标操作。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB HID 鼠标演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hid_mouse.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/hid_mouse`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hid_mouse.X`	`<install-dir>/apps/usb/device/hid_mouse/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx460_pim_e16_int_dyn	pic32mx460_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX460F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_da_sk_int_dyn	pic32mz_da_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 图形（DA）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX460F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路

运行演示

介绍如何编译并运行 HID 鼠标演示。

说明

该演示使用选定的硬件平台作为 USB 鼠标。在编译时，请基于演示板选择适当的 MPLAB X IDE 项目配置。详情请参见编译应用程序。

通过 USB 电缆将电路板连接到计算机之前，请务必注意，该设备会开始移动计算机上的鼠标光标。有两种方法可以让设备禁止光标继续移动。第一种方法是将该设备与计算机断开。第二种是在硬件平台上按下正确的按钮。再次按下按钮会导致鼠标光标再次开始以环形移动。

下表列出了为停止环形移动需要在演示板上按下的按钮。

演示板	按钮
PIC32 USB 入门工具包 II PIC32 USB 入门工具包 III PIC32MZ 图形 （DA）入门工具包 PIC32MZ 嵌入式连接 （EC）入门工具包 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包	SW1
Explorer 16 开发板	S3

hid_msd_basic

演示 HID 设备类和 MSD 类复合 USB 设备。

说明

该演示应用程序会创建组合了 hid_basic 和 msd_basic 演示应用程序功能的 USB 设备。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了该演示应用程序的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hid_msd_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/hid_msd_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hid_msd_basic.X`	`<install-dir>/apps/usb/device/hid_msd_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示应用程序。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的应用程序演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该配置可以在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上运行配置为中断模式和动态操作的应用程序演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行演示。

说明

该演示作为一个复合 USB 设备工作，它组合了 hid_basic 和 msd_basic 演示应用程序创建的设备的功能。关于运用 HID 和 MSD 功能的详细信息，请分别参见 hid_basic 演示的运行演示部分和 msd_basic 演示的运行演示部分。

msd_basic

演示仿真闪存驱动器的 USB MSD 设备。

说明

该演示应用程序会使用海量存储设备类创建一个闪存驱动器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB MSD 基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `msd_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/msd_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`msd_basic.X`	`<install-dir>/apps/usb/device/msd_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_bt_sk_int_dyn	pic32mx_bt_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32 蓝牙入门工具包上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_poll_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上使用 USB 设备协议栈运行配置为查询模式和动态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_poll_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上使用 USB 设备协议栈运行配置为查询模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_poll_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上运行配置为查询模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32 蓝牙入门工具包

不需要任何硬件相关的配置或跳线设置。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

运行演示

介绍如何编译并运行 USB MSD 基本演示。

说明

该演示使用选定的硬件平台作为计算机上的逻辑驱动器，使用设备的内部闪存作为驱动器存储介质。通过 USB 电缆将硬件平台连接到计算机。设备将在计算机上显示为名为 “Drive Name”的新驱动器。该驱动器可以用于存储文件。

注： 重新对开发板编程会使存储的所有文件都被擦除。

msd_sdcard

演示通过 USB MSD 从 SD 卡到计算机的数据传输。

说明

该应用程序演示如何使用 SD 卡读卡器通过 USB 海量存储设备 （Mass Storage Device， MSD）类在计算机和 SD 卡之间传输数据。高速 USB 用于主机计算机与 PIC32 设备之间的通信，而 SD 卡用作存储介质。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB MSD SD 卡演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `msd_sdcard.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/msd_sdcard`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`msd_sdcard.X`	`<install-dir>/apps/usb/device/msd_sdcard/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk+meb2	选择该 MPLAB X IDE 项目配置可以在连接到 MEB II 的 PIC32MZ EC 入门工具包上运行演示。介质驱动程序配置为中断模式和动态操作。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ 嵌入式连接 （EC）入门工具包和多媒体扩展板 II （MEB II）

不需要任何硬件相关的配置或跳线设置。

运行演示

介绍如何编译并运行 USB MSD SD 卡演示。

说明

该演示使用选定的硬件平台作为计算机上的逻辑驱动器，使用 SD 卡作为驱动器存储介质。通过 USB 电缆将硬件平台连接到计算机。设备将在计算机上显示为名为 “Drive Name”的新驱动器。然后，可以使用该驱动器来存储文件。

vendor

演示通过使用 USB 设备层端点函数创建的定制 USB 设备。

说明

该演示应用程序会使用 USB 设备层端点函数创建定制 USB 设备。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了供应商 USB 设备演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `vendor.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/vendor`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`vendor.X`	`<install-dir>/apps/usb/device/vendor/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx460_pim_e16_int_dyn	pic32mx460_pim+e16	选择该 MPLAB X IDE 项目配置可以在 Explorer 16 开发板上运行配置为中断模式和动态操作的演示。该配置还需要 PIC32MX460F512L 接插模块（PIM）和 USB PICtail Plus 子板。
pic32mx_usb_sk3_int_dyn	pic32mx_usb_sk3	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 III 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

移除跳线 JP2。

PIC32MZ EC 入门工具包

移除跳线 JP1。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32 USB 入门工具包 III

移除跳线 JP1。

PIC32MX460F512L PIM

需要移除跳线 J10。该接插模块应与 Explorer 16 开发板和 USB PICtail Plus 子板一起使用。单片机 PIM 应插入开发板上的 PIM 插座。应将 USB PICtail Plus 子板连接到边缘连接器 J9。

在 Explorer 16 开发板上：

- 开关 S2 应设置到 PIM
- 跳线 JP2 应就位

在 USB PICtail Plus 子板上：

- 跳线 JP1 应就位
- 需要移除跳线 JP2 和 JP4

在 PIC32MX460F512L PIM 上：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路

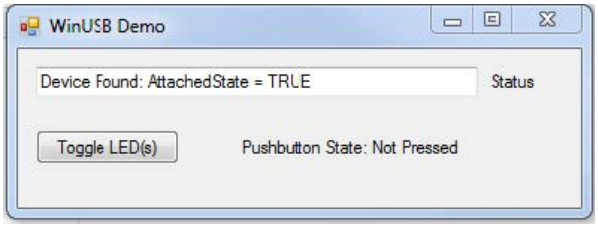
运行演示

介绍如何编译并运行供应商 USB 设备演示。

说明

供应商设备可以通过使用 WinUSB PnP 演示应用程序来运用，该应用程序在 MPLAB Harmony 安装中提供。该应用程序可以使电路板上的 LED 的状态发生翻转，并指示电路板上的开关的状态 （按下 / 释放）。

要启动该应用程序，请双击位于 `<install dir>/apps/usb/device/vendor/bin` 的 `WinUSB PnP Demo.exe`。此时会显示一个类似于下图的对话框：



按下 Toggle LED 按钮会导致电路板上的 LED 发生翻转。应用程序中的 Pushbutton State （按钮状态）字段会指示所连接 USB 设备上的某个按钮的状态。按下开发板上的该开关时，将会更新 Pushbutton State 字段的 Pressed/Not Pressed （按下 / 未按下）状态。

演示板	按钮
PIC32 USB 入门工具包 II PIC32 USB 入门工具包 III PIC32MZ 嵌入式连接 （EC）入门工具包 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包	SW1
Explorer 16 开发板	S3

主机

本部分介绍 USB 主机演示。

audio_speaker

该应用程序演示如何使用音频 v1.0 主机类驱动程序来枚举并操作音频扬声器设备。

说明

该应用程序演示如何使用音频 v1.0 主机类驱动程序来枚举音频扬声器设备。该应用程序使用 USB 主机层和音频 1.0 类驱动程序来枚举音频 v1.0 USB 设备。然后，演示主机应用程序会操作并使用所连接的音频扬声器设备的功能。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB 设备音频扬声器演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `audio_speaker.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/host/audio_speaker`.。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`audio_speaker.X`	`<install-dir>/apps/usb/host/audio_speaker/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

JP2 应就位。

PIC32MZ EC 入门工具包

JP1 应就位，并且应移除以太网接插板。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 USB 主机音频 v1.0 基本演示。

说明

该应用程序演示如何使用音频 v1.0 主机类驱动程序来枚举并操作音频 v1.0 设备。该应用程序使用 USB 主机层和音频 v1.0 类驱动程序来枚举音频 v1.0 USB 设备。然后，演示主机应用程序会操作并使用所连接的音频 v1.0 设备的功能。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对器件编程。
3. 将市售 USB 扬声器连接到电路板。
4. 如果音频 1.0 类驱动程序接受所连接的设备，则 LED1 会点亮。
5. 扬声器会产生 1 kHz 正弦波。
6. 如果演示应用程序无法接受设备， LED2 将持续闪烁。
7. 按下开关 SW1 可使音频变为静音。
8. 按下开关 SW2 可取消音频静音。

关于演示限制，请参见发行说明 （如适用）。

cdc_basic

该应用程序演示如何使用 CDC 主机类驱动程序来枚举并操作 CDC 设备。

说明

该应用程序演示如何使用 CDC 主机类驱动程序来枚举并操作 CDC 设备。该应用程序会使用 USB 主机层和 CDC 类驱动程序来枚举 CDC USB 设备。然后，演示主机应用程序会操作并使用所连接 CDC 设备的功能。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB CDC 主机基本演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `cdc_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/host/cdc_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`cdc_basic.X`	`<install-dir>/apps/usb/host/cdc_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mx_usb_sk2_poll_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为查询模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_poll_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为查询模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元 （EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

JP2 应就位。

PIC32MZ EC 入门工具包

JP1 应就位，并且应移除以太网接插板。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

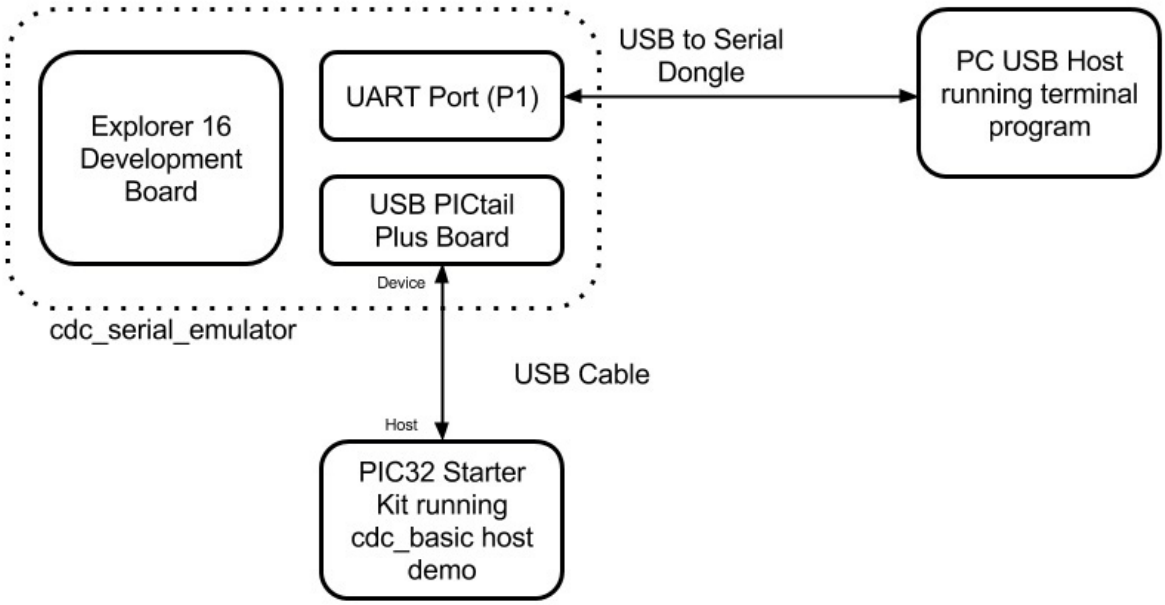
介绍如何编译并运行 USB 主机 CDC 基本演示。关于演示限制，请参见发行说明 （如适用）。

说明

该应用程序演示如何使用 CDC 主机类驱动程序来枚举并操作 CDC 设备。该应用程序会使用 USB 主机层和 CDC 类驱动程序来枚举 CDC USB 设备。然后，演示主机应用程序会操作并使用所连接 CDC 设备的功能。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对器件编程。
3. 按照指示来设置并运行 cdc_serial_emulator USB 设备演示。
4. 将 Explorer 16 开发板 （运行 cdc_serial_emulator 演示）上的 UART （P1）端口通过市售串行转 USB 转换器连接到 USB 主机个人计算机。
5. 在 USB 主机个人计算机上启动终端程序，并选择串行转 USB 转换器作为通信端口。选择波特率 9600、无奇偶校验、1 个停止位和无流量控制。
6. 将 cdc_serial_emulator 演示设置的 USB PICtail Plus 子板上的 mini-B 连接器连接到入门工具包上的 A 型 USB 主机连接器。
7. 终端仿真程序上会立即显示一个提示符 （LED :）。
8. 按下 USB 主机键盘上的 1、 2 或 3 键会分别导致 PIC32 入门工具包 （运行 USB CDC 主机应用程序）上的 LED 1、 2 或 3 点亮。
9. 终端仿真程序上将再次显示该提示符，然后可以重复步骤 8。

其设置应类似于下图：



cdc_serial_emulator 演示会仿真 USB 转串行转换器。CDC 主机（运行 cdc_basic 演示应用程序）会将提示符处的消息发送到 CDC 设备。CDC 设备会将提示符消息转发到 UART 端口，在这里它通过 USB 转串行转换器发送到个人计算机 USB 主机。在个人计算机 USB 主机上按下某个键时，按键值会被发送到 CDC 设备， CDC 设备接着会将按键数据发送到 CDC 主机。然后， cdc_basic 演示会分析按键数据，并点亮相应的 LED。

cdc_msd

演示主机对于多个设备类的支持。

说明

该演示应用程序会创建可以在一个应用程序中支持不同设备类的 USB 主机。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了该 USB CDC MSD 主机演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 cdc_msd.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/usb/host/cdc_msd。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
cdc_msd.X	<install-dir>/apps/usb/host/cdc_msd/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

JP2 应就位。

PIC32MZ EC 入门工具包

JP1 应就位，并且应移除以太网接插板。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 USB CDC MSD 演示。

说明

该演示应用程序会创建一个枚举 CDC 和 MSD 设备的 USB 主机应用程序。该应用程序将主机 cdc_basic 和 msd_basic 演示应用程序的功能组合到一个应用程序中。如果连接了 CDC 设备，该演示应用程序的行为会类似于 cdc_basic 主机应用程序。如果连接了 MSD 设备，该演示应用程序的行为会类似于 msd_basic 主机应用程序。

关于运用演示的 CDC 和 MSD 主机方面的详细信息，请分别参见主机 cdc_basic 演示的运行演示部分和主机 msd_basic 演示的运行演示部分。

hid_basic_keyboard

演示如何使用 USB HID 主机客户端驱动程序与键盘使用驱动程序来方便 USB HID 键盘与 PIC32 USB 主机的使用。

说明

该应用程序演示如何使用 USB HID 主机客户端驱动程序来枚举并操作 HID 键盘设备。USB 主机层会枚举一个 USB 键盘。USB HID 主机客户端驱动程序可以方便 USB 键盘和 PIC32 USB 主机之间的通信。键盘事件使用个人计算机上的终端模拟器显示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB HID 基本键盘演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hid_basic_keyboard.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/device/hid_keyboard`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hid_basic_keyboard.X`	`<install-dir>/apps/usb/host/hid_basic_keyboard/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx795_pim_e16_int_dyn	pic32mx795_pim+e16	选择该 MPLAB X IDE 项目配置可以在连接到 Explorer 16 开发板与 USB PICtail Plus 子板的 PIC32MX795F512L PIM 上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

Explorer 16 开发板：

- 开关 S2 应设置到 PIM

USB PICtail Plus 子板：

- 使用跳线连接主机使能引脚
- 设备使能引脚和 OTG 使能引脚应开路

PIC32MX795F512L 接插模块 （PIM）：

- 跳线 J10 保持开路
- J9 中的所有跳线保持开路
- 跳线 J1 应在位置 1 和 2 之间短接。该配置仅适用于 PIC32MX795F512L USB CAN PIM （MA320003），不适用于 PIC32MX795F512L USB PIM （MA320002）。
- 跳线 J2 应在位置 1 和 2 之间短接。该配置仅适用于 PIC32MX795F512L USB CAN PIM （MA320003），不适用于 PIC32MX795F512L USB PIM （MA320002）。

在配置硬件之后，执行以下步骤：

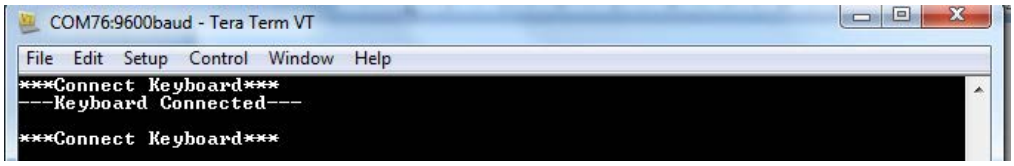
1. 确保 PIC32MX795F512L PIM 正确连接到 Explorer 16 开发板上的 PIM 插座。
2. 使用串行转 USB 转换器电缆将 Explorer 16 开发板上的串行端口连接器连接到 PC。
3. 将 USB PICtail Plus 子板连接到 Explorer 16 开发板的水平边缘连接器 （J9）。

运行演示

介绍如何编译并运行 USB HID 基本键盘演示。

说明

1. 在 MPLAB X IDE 中打开项目，并选择项目配置。
2. 编译代码并对器件编程。
3. 启动终端仿真器 （如 Tera Term）并选择适当的 COM 端口，并将串行端口设置为 9600-N-1。
4. 如果未使用 USB PICtail Plus 子板上的 J4 将 USB 键盘连接到主机连接器，终端仿真器窗口会显示 *Connect Keyboard*（连接键盘）提示。
5. 使用 USB PICtail Plus 子板上的 J4 将 USB 键盘连接到主机连接器。终端仿真器窗口中会显示消息 *Keyboard Connected* （键盘已连接）。
6. 开始在键盘上输入，将会显示相应的键符。
7. 断开键盘会导致显示消息 *Connect Keyboard*。



hid_basic_mouse

演示 USB 主机应用程序对于 USB HID 鼠标的支持。

说明

该应用程序演示如何使用 USB HID 主机客户端驱动程序来枚举并操作 HID 鼠标设备。USB 主机层会枚举一个 USB 鼠标。USB HID 主机客户端驱动程序可以方便 USB 鼠标和应用程序之间的通信。特定于鼠标的事件通过 MEB II 显示屏上的光标移动来演示。鼠标按钮单击通过 LED 来指示。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB 主机 HID 基本鼠标演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hid_basic_mouse.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/host/hid_basic_mouse`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hid_basic_mouse.X`	`<install-dir>/apps/usb/host/hid_basic_mouse/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mz_ec_sk_meb2	pic32mz_ec_sk+meb2	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ EC 入门工具和 MEB II 上运行演示。

pic32mz_ef_sk_meb2	pic32mz_ef_sk+meb2	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ EF 入门工具和 MEB II 上运行演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32MZ EC 入门工具包和 MEB II

1. 确保 PIC32MZ EC 入门工具包紧固在 MEB II 扩展板上。
2. PIC32MZ EC 入门工具包上的 JP1 应就位。
3. PIC32MZ EC 以太网入门工具包上的接插模块应移除。

PIC32MZ EF 入门工具包和 MEB II

确保 PIC32MZ EC 入门工具包紧固在 MEB II 扩展板上。

运行演示

介绍如何编译并运行 USB 主机 HID 基本鼠标演示。关于演示限制，请参见发行说明 （如适用）。

说明

该应用程序演示如何使用 HID 主机客户端驱动程序来枚举 USB HID 鼠标并与鼠标建立通信。该应用程序使用 USB 主机层、 HID 客户端驱动程序和 MPLAB Harmony 图形库来枚举 USB HID 鼠标并在 USB 主机端捕捉特定于鼠标的事件。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对器件编程。
3. 如果 USB 鼠标未连接到 PIC32MZ EC/EF 入门工具包上的 USB 主机连接器， MEB II 上的显示屏会在红色背景上显示 “Connect Mouse”（连接鼠标）提示。
4. 将 USB 鼠标连接到 PIC32MZ EC/EF 入门工具包上的主机连接器。“Connect Mouse”提示会消失，屏幕上会出现光标 （一个小白点）。
5. 移动鼠标可以使光标移动。鼠标按钮单击会使 MEB II 上的 LED 发生翻转。

LED 指示详细信息请参见下表。

鼠标单击	MEB II
左	D3
右	D4
中	D5
左下	D6
右下	D7

6. 断开鼠标会导致重新出现 “Connect Mouse”提示。

hub_cdc_hid

演示如何通过外部集线器来枚举 HID 鼠标和 CDC 仿真器设备。

说明

该应用程序演示 USB 主机协议栈通过集线器访问和管理多个 USB 设备的功能。

该演示应用程序会通过外部集线器枚举 HID 鼠标和 CDC 仿真器设备。主机将演示来自 CDC 仿真器设备和 HID 鼠标的通信。

编译应用程序

本主题指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB 主机集线器 CDC HID 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 hub_cdc_hid.X 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 <install-dir>/apps/usb/host/hub_cdc_hid。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
hub_cdc_hid.X	<install-dir>/apps/usb/host/hub_cdc_hid/firmware

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 ./firmware/src/system_config 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ EC 入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ EF 入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

JP2 应就位。

PIC32MZ EC 入门工具包

JP1 应就位，并且应移除以太网接插板。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 USB 主机集线器 CDC HID 演示。

说明

该应用程序演示USB主机协议栈通过集线器访问和管理多个USB设备的功能。该演示应用程序会通过外部集线器枚举HID鼠标和CDC仿真器设备。主机将演示来自 CDC 仿真器设备和 HID 鼠标的通信。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对器件编程。
3. 将集线器连接到所需电路板上的 A 型主机连接器。
4. 将鼠标连接到集线器上的备用端口。
5. 将 CDC 仿真器设备连接到集线器上的另一个备用端口。
6. 单击鼠标使入门工具包上的 LED 发生翻转。
7. 在个人计算机上，打开终端仿真器。在提示符 （LED:）处，输入 1、 2 或 3 来使入门工具包上的 LED 发生翻转。

hub_msd

该应用程序演示 USB 主机协议栈通过集线器支持多个 MSD 设备的功能。

说明

该应用程序演示如何使用集线器驱动程序和 MSD 主机客户端驱动程序以及文件系统来支持多个 MSD 设备和集线器。该演示应用程序会将一个文件从一个笔式驱动器复制到另一个笔式驱动器。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB 主机集线器 MSD 演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `hub_msd.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/host/hub_msd`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`hub_msd.X`	`<install-dir>/apps/usb/host/hub_msd/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接（EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

JP2 应就位。

PIC32MZ EC 入门工具包

JP1 应就位，并且应移除以太网接插板。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 USB 主机集线器 MSD 演示。

说明

该应用程序演示 USB 主机协议栈通过集线器访问和管理多个 USB 设备的功能。该演示应用程序会将一个文件从一个 USB 笔式驱动器（即 USB 闪存存储设备）复制到另一个 USB 笔式驱动器，而这两个笔式驱动器连接到一个集线器。

注： 该演示会在已连接的任意笔式驱动器上搜索名为file.txt的文件。应通过任意适合方法在笔式驱动器之一上创建该文件。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对器件编程。
3. 将集线器连接到所需电路板上的 A 型主机连接器。
4. 将包含名为 `file.txt` 的任意文件的 USB 笔式驱动器连接到集线器上的一个备用端口。
5. 将另一个 USB 笔式驱动器连接到集线器上的另一个备用端口。
6. 应用程序会将文件 `file.txt` 从包含该文件的驱动器复制到另一个驱动器上。复制的文件会被重命名为 `newfile.txt`。演示板上的 LED 2 会点亮，指示文件传输完成。
7. 断开驱动器并将它们插入个人计算机，并验证文件传输已按预期完成，以此确认演示成功。

该演示应用程序会总是处于等待两个笔式驱动器连接到集线器，并且至少其中一个笔式驱动器包含名为 `file.txt` 的文件的状态。

msd_basic

该应用程序演示如何使用 MSD 主机类驱动程序将文件写入 USB 闪存驱动器。

说明

该应用程序演示如何使用 MSD 主机类驱动程序将文件写入 USB 闪存驱动器。该应用程序会使用 USB 主机层、 MSD 类驱动程序和 MPLAB Harmony 文件系统框架来枚举 USB 闪存驱动器并向其中写入一个文件。

编译应用程序

本部分指明了 MPLAB X IDE 项目名称和位置，并且列出和说明了 USB MSD 主机类驱动程序演示的可用配置。

说明

要编译该项目，必须在 MPLAB X IDE 中打开 `msd_basic.X` 项目，然后选择所需的配置。

以下几个表列出并说明了项目和支持的配置。这些文件的父文件夹为 `<install-dir>/apps/usb/host/msd_basic`。

MPLAB X IDE 项目

下表列出了演示的 MPLAB X IDE 项目文件夹的名称和位置。

项目名称	位置
`msd_basic.X`	`<install-dir>/apps/usb/host/msd_basic/firmware`

MPLAB X IDE 项目配置

下表列出并说明了演示支持的配置，它们位于 `./firmware/src/system_config` 中。

项目配置名称	使用的 BSP	说明
chipkit_wf32	chipkit_wf32	在 chipKIT WF32 开发板上运行的演示。
chipkit_wifire	chipkit_wifire	在 chipKIT Wi-FIRE 开发板上运行的演示。
pic32mx_usb_sk2_int_dyn	pic32mx_usb_sk2	选择该 MPLAB X IDE 项目配置可以在 PIC32 USB 入门工具包 II 上运行配置为中断模式和动态操作的演示。

pic32mz_da_sk_int_dyn	pic32mz_da_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 图形 （DA）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ec_sk_int_dyn	pic32mz_ec_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接 （EC）入门工具包上运行配置为中断模式和动态操作的演示。
pic32mz_ef_sk_int_dyn	pic32mz_ef_sk	选择该 MPLAB X IDE 项目配置可以在 PIC32MZ 嵌入式连接与浮点单元（EF）入门工具包上运行配置为中断模式和动态操作的演示。

配置硬件

介绍如何配置支持的硬件。

说明

PIC32 USB 入门工具包 II

JP2 应就位。

PIC32MZ DA 入门工具包

不需要硬件相关配置或跳线设置更改。

PIC32MZ EC 入门工具包

JP1 应就位，并且应移除以太网接插板。

PIC32MZ EF 入门工具包

不需要硬件相关配置或跳线设置更改。

chipKIT WF32 Wi-Fi 开发板

不需要硬件相关配置或跳线设置更改。

chipKIT Wi-FIRE 开发板

不需要硬件相关配置或跳线设置更改。

运行演示

介绍如何编译并运行 USB 主机 MSD 基本演示。关于演示限制，请参见发行说明 （如适用）。

说明

该应用程序演示如何使用 MSD 主机类驱动程序将文件写入 USB 闪存驱动器。该应用程序会使用 USB 主机层、 MSD 类驱动程序和 MPLAB Harmony 文件系统框架来枚举 USB 闪存驱动器并向其中写入一个文件。

1. 在 MPLAB X IDE 中打开项目，并选择所需的项目配置。
2. 编译代码并对器件编程。
3. 在代码运行时，将 USB 闪存驱动器连接到所需入门工具包上的主机连接器。
4. 然后，该演示应用程序会创建一个名为 `file.txt` 的文件。之后，它会在该文件中写入 “Hello World”，然后关闭该文件。
5. 接着演示会切换为空闲模式，这通过点亮入门工具包上的 LED2 来指示。
6. 然后，可以将 USB 闪存驱动器连接到 USB 主机个人计算机，以验证演示应用程序的操作。
7. 可以重复步骤 3 至 6。
8. 如果 USB 闪存驱动器已包含名为 `file.txt` 的文件，该演示应用程序会将文本 “Hello World”追加到文件内容的末尾。
9. 如果文件创建或写操作失败，入门工具包上的 LED1 会点亮。

索引

数字

12 位高速 SAR ADC （ADCHS）外设库示例 74

A

adchs_3ch_dma 75
adchs_oversample 76
adchs_pot 78
adchs_sensor 80
adchs_touchsense 82
adcp_cal 84
adc_pot 73
adc_pot_dma 73
ADC 外设库示例 72
a2dp_avrcp 31
APP_ChangeMode 函数 202
APP_DATA 结构 194
APP_DrawBubble 函数 189
APP_DrawMenu 函数 189
APP_FillMenuBackBuffer 函数 190
APP_GenerateSysMsgGet 函数 202
APP_GLCD_LAYER0_BUFFER_ADDR 宏 197
APP_GLCD_LAYER1_ALPHA_RESOLUTION 宏 197
APP_GLCD_LAYER1_BUFFER_ADDR 宏 197
APP_GLCD_LAYER2_BUFFER_ADDR 宏 198
APP_GLCD_LAYER2_HOR_RES 宏 198
APP_GoToNextSlide 函数 203
APP_HandleHomeSlide 函数 190
APP_HandleLanguageSetting 函数 203
APP_HandleTouchTest 函数 203
APP_HOME_MODE_SEMI_TRANSPARENT_ALPHA_VALUE 宏 198
APP_Initialize 函数 190
APP_IsCircle 函数 191
APP_LANGUAGES 枚举 206
APP_MENU_STATES 枚举 195
APP_MODES 枚举 196
APP_MoveBubble 函数 191
APP_PrepareHomeMode 函数 192
APP_ProcessMenu 函数 192
APP_ProcessModeState 函数 204
APP_ReceiveMenuTouch 函数 192
APP_RedrawRectangle 函数 204
APP_SelectNewSlide 函数 193
APP_SetSlidePauseTime 函数 204
APP_STATES 枚举 196
APP_Tasks 函数 193
APP_TouchEventHandler 函数 194
APP_TouchMessageCallback 函数 204
APP_TRANSPARENT_ALPHA_VALUE 宏 199
APP_UpdateFeatureList 函数 205
APP_UpdateLanguageTexts 函数 205
APP_UpdateMainMenu 函数 205
APP_UpdateSlideShowTips 函数 206
audio_microphone_loopback 3
audio_speaker 354
audio_tone 5

B

basic 34, 217, 223, 232, 233, 238, 243
berkeley_tcp_client 253
berkeley_tcp_server 254
berkeley_udp_client 255
berkeley_udp_relay 257
berkeley_udp_server 258
blinky_leds 107
BMX 外设库示例 86
比较器外设库示例 90
编译应用程序 3, 5, 8, 12, 16, 20, 24, 32, 34, 39, 42, 44, 46, 50, 52, 55, 56, 61, 64, 65, 67, 73, 74, 75, 77, 78, 80, 82, 84, 86, 87, 91, 92, 94, 95, 97, 99, 101, 102, 104, 106, 107, 108, 110, 111, 112, 114, 116, 118, 119, 121, 122, 124, 125, 128, 129, 132, 134, 136, 137, 139, 141, 143, 148, 150, 153, 155, 157, 160, 164, 168, 171, 175, 180, 182, 184, 199, 207, 208, 210, 213, 214, 218, 219, 220, 222, 223, 224, 225, 227, 228, 232, 233, 234, 236, 237, 238, 239, 241, 242, 243, 245, 246, 247, 253, 254, 255, 257, 258, 260, 262, 263, 266, 267, 268, 269, 271, 272, 276, 280, 288, 290, 292, 302, 305, 308, 316, 317, 319, 329, 332, 335, 336, 338, 340, 342, 345, 347, 348, 350, 351, 352, 354, 356, 357, 359, 360, 361, 363, 364
ADCHS 外设库演示 （adchs_3ch_dma）75
ADCHS 外设库演示 （adchs_oversample）77
ADCHS 外设库演示 （adchs_pot）78
ADCHS 外设库演示 （adchs_sensor）80
ADCHS 外设库演示 （adchs_touchsense）82
ADC 外设库演示 （adc_pot）73
ADC 外设库演示 （adc_pot_dma）74
BMX 外设库演示 （mem_partition）86
CAN 外设库演示 （echo_send）87
CVREF 外设库演示 （triangle_wave）92
DMA 外设库演示 （dma_led_pattern）94
DMA 系统服务库演示 （dma_crc）139
EBI 外设库演示 （sram_read_write）95
I2C 驱动程序演示 （i2c_rtcc）50
I2C 外设库演示 （i2c_interrupt）97
MEB II 演示 （gfx_camera）207
MEB II 演示 （gfx_cdc_com_port_single）208
MEB II 演示 （gfx_photo_frame）210
MEB II 演示 （gfx_web_server_nvm_mpfs）213
NVM 驱动程序演示 （nvm_read_write）52
NVM 外设库演示 （flash_modify）101
OSC 外设库演示 （osc_config）104
PIC32 蓝牙协议栈库演示 （a2dp_avrcp）32
PIC32 蓝牙协议栈库演示 （data_temp_sens_rgb）24
PIC32 蓝牙协议栈演示 （data_basic）20
PMP 外设库演示 （pmp_lcd）106
RTCC 外设库演示 （rtcc_alarm）114
RTOS 演示 /FreeRTOS （basic）223
RTOS 演示 /FreeRTOS （cdc_com_port_dual）224
RTOS 演示 /FreeRTOS （cdc_msd_basic）225
RTOS 演示 /FreeRTOS （gfx）227
RTOS 演示 /FreeRTOS （tcpip_client_server）228
RTOS 演示 /Micriμm （basic）232, 233
RTOS 演示 /Micriμm （gfx）234
RTOS 演示 /Micriμm （gfx_usb）236
RTOS 演示 /Micriμm （usb）237, 238
RTOS 演示 /OPENRTOS （basic）238
RTOS 演示 /OPENRTOS （cdc_com_port_dual）239
RTOS 演示 /OPENRTOS （cdc_msd_basic）241

RTOS 演示 /OPENRTOS （gfx）242
RTOS 演示 /SEGGER embOS （basic）243
RTOS 演示 /SEGGER embOS （gfx）245
RTOS 演示 /SEGGER embOS （gfx_usb）246
RTOS 演示 /SEGGER embOS （usb）247
RTOS 演示 /ThreadX （basic）218
RTOS 演示 /ThreadX （gfx）219
RTOS 演示 /ThreadX （gfx_usb）220
RTOS 演示 /ThreadX （usb）222
SPI 驱动程序演示 （serial_eeprom）55
SPI 驱动程序演示 （spi_loopback）56
SPI 驱动程序演示 （spi_multislave）61
SPI 闪存驱动程序演示 （sst25vf020b）64
SPI 演示 （spi_loopback）116
SQI 演示 （flash_read_dma_mode）118
SQI 演示 （flash_read_pio_mode）119
SQI 演示 （flash_read_xip_mode）121
TCPIP 演示 （berkeley_tcp_client）253
TCPIP 演示 （berkeley_tcp_server）254
TCPIP 演示 （berkeley_udp_client）255
TCPIP 演示 （berkeley_udp_relay）257
TCPIP 演示 （berkeley_udp_server）258
TCPIP 演示 （pic32_eth_wifi_web_server）280
TCPIP 演示 （pic32_wifi_web_server）288
TCPIP 演示 （snmpv3_nvm_mpfs）260
TCPIP 演示 （snmpv3_sdcard_fatfs）262
TCPIP 演示 （tcpip_tcp_client）263
TCPIP 演示 （tcpip_tcp_client_server）266
TCPIP 演示 （tcpip_tcp_server）267
TCPIP 演示 （tcpip_udp_client）268
TCPIP 演示 （tcpip_udp_client_server）269
TCPIP 演示 （tcpip_udp_server）271
TCPIP 演示 （web_net_server_nvm_mpfs）272
TCPIP 演示 （web_server_nvm_mpfs）276
TCPIP 演示 （web_server_sdcard_fatfs）290
TCPIP 演示 （wifi_easyconf）292
TCPIP 演示 （wifi_wolfssl_tcp_client）305
TCPIP 演示 （wifi_wolfssl_tcp_server）308
TCPIP 演示 （wolfssl_tcp_client）316
TCPIP 演示 （wolfssl_tcp_server）317
TMR 外设库演示 （timer3_interrupt）122
USART 驱动程序演示 （usart_echo）65
USART 驱动程序演示 （usart_loopback）67
USART 外设库演示 （uart_basic）124
USB 设备演示 （audio_speaker）354
USB 设备演示 （cdc_com_port_dual_build）329
USB 设备演示 （cdc_com_port_single）332
USB 设备演示 （cdc_serial_emulator）336
USB 设备演示 （cdc_serial_emulator_msd）338
USB 设备演示 （hid_basic）340
USB 设备演示 （hid_joystick）342
USB 设备演示 （hid_keyboard）345
USB 设备演示 （hid_mouse）347
USB 设备演示 （hid_msd_basic）348
USB 设备演示 （msd_basic）350
USB 设备演示 （msd_sdcard）351
USB 设备演示 （vendor_device）352
USB 演示 （cdc_msd_basic）335
USB 主机演示 （cdc_basic）356
USB 主机演示 （cdc_msd）357
USB 主机演示 （hid_basic_mouse）360
USB 主机演示 （hub_cdc_hid）361
USB 主机演示 （hub_msd）363
USB 主机演示 （msd_basic）364
WDT 外设库演示 （wdt_timeout）125
比较器外设库演示 （simple_comparator）91
测试应用程序 （test_sample）319
电源外设库演示 （deep_sleep_mode）111
电源外设库演示 （sleep_mode）110
调试系统服务库演示 （debug_uart）129
调试系统服务库演示 （debug_usb_cdc_2）132
端口外设库演示 （blinky_leds）107
端口外设库演示 （cn_interrupt）108
复位外设库演示 （reset_handler）112
加密库演示 （encrypt_decrypt）44
加密库演示 （large_hash）46
流水线 ADC 外设库演示 （adcp_cal）84
命令处理器系统服务库演示 （command_appio）128
器件控制系统服务库演示 （devcon_cache_clean）134
器件控制系统服务库演示 （devcon_cache_invalidate）136
器件控制系统服务库演示 （devcon_sys_config_perf）137
输出比较外设库演示 （oc_pwm）102
输入捕捉外设库演示 （ic_basic）99
图形库演示 （object）171
图形库演示 （primitive）175
图形演示 （lcc）168
图形演示 （s1d13517）180
图形演示 （ssd1926）182
图形演示 （wvga_glcd）184
图形演示 （wvga_lcc）199
外部存储器编程器演示 （external_flash）141
外部存储器编程器演示 （sqi_flash）143
文件系统演示 （nvm_fat_single_disk）148
文件系统演示 （nvm_mpfs_single_disk）150
文件系统演示 （nvm_sdcard_fat_mpfs_multi_disk）153
文件系统演示 （nvm_sdcard_fat_multi_disk）155
文件系统演示 （sdcard_fat_single_disk）157
文件系统演示 （sdcard_msd_fat_multi_disk）160
音频演示 （audio_microphone_loopback）3
音频演示 （audio_tone）5
音频演示 （mac_audio_hi_res）8
音频演示 （universal_audio_decoders）12
音频演示 （usb_headset）16
自举程序演示 （basic）34
自举程序演示 （LiveUpdate_uart_bootloader）39

C

CAN 外设库示例 87
cdc_basic 355
cdc_com_port_dual 224, 239, 329
cdc_com_port_single 331
cdc_msd 357
cdc_msd_basic 225, 241, 335
cdc_serial_emulator 336
cdc_serial_emulator_msd 338
cn_interrupt 108

command_appio 127
CVREF 外设库示例 92
测试应用程序 319

D

data_basic 20
data_temp_sens_rgb 24
debug_uart 129
debug_usb_cdc_2 131
deep_sleep_mode 111
devcon_cache_clean 134
devcon_cache_invalidate 135
devcon_sys_config_perf 137
dma_crc 139
dma_led_pattern 94
DMA 外设库示例 93
DMA 系统服务示例 138
电源外设库示例 109
调试系统服务示例 129
定时器外设库示例 121
端口外设库示例 106

E

EBI 外设库示例 95
echo_send 87
encrypt_decrypt 44
Express Logic ThreadX 演示 217
external_flash 141
external_resources 162

F

flash_modify 100
flash_read_dma_mode 117
flash_read_pio_mode 119
flash_read_xip_mode 120
FreeRTOS 演示 223
复位外设库示例 112

G

gfx 219, 227, 234, 242, 244
gfx_camera 207
gfx_cdc_com_port_single 208
gfx_photo_frame 209
gfx_usb 220, 235, 246
gfx_web_server_nvm_mpfs 212
高级演示 31

H

hid_basic 340
hid_basic_keyboard 358
hid_basic_mouse 360
hid_joystick 342
hid_keyboard 345
hid_mouse 346
hid_msd_basic 348
hub_cdc_hid 361
hub_msd 363

I

i2c_interrupt 97
I2C 驱动程序演示 50
i2c_rtcc 50
I2C 外设库示例 97
ic_basic 99

J

加密演示 44
简介 3, 18, 34, 44, 50, 52, 54, 64, 65, 72, 74, 84, 86, 87, 90, 92, 93, 95, 97, 99, 100, 102, 104, 105, 106, 109, 112, 114, 115, 117, 122, 123, 125, 127, 129, 134, 138, 141, 148, 162, 207, 217, 249, 319, 322
ADCHS 演示 74
ADC 演示 72
BMX 演示 86
CAN 库演示 87
CMP 演示 90
CVREF 演示 92
DMA 系统服务库演示 138
DMA 演示 93
EBI 演示 95
I2C 驱动程序演示 50
I2C 演示 97
IC 演示 99
NVM 驱动程序演示 52
NVM 演示 100
OC 演示 102
OSC 演示 104
PIC32 蓝牙协议栈库演示 18
PMP 演示 105
RTCC 演示 114
RTOS 演示 217
SEGGER emWin MEB II 演示 207
SPI 驱动程序演示 54, 64
SPI 演示 115
SQI 演示 117
TCPIP 演示 249
TMR 演示 122
USART 驱动程序演示 65
USART 演示 123
USB 演示 322
WDT 演示 125
测试帮助 319
电源演示 109
调试系统服务库演示 129
端口演示 106
复位演示 112
加密库演示 44
流水线 ADC 演示 84
命令处理器系统服务库演示 127
器件控制系统服务库演示 134
外部存储器编程器演示 141
外设库示例应用程序 72
文件系统演示 148
自举程序演示 34

L

large_hash 45
lcc 168
LiveUpdate 39
LiveUpdate_StateRetention 42
蓝牙演示 18
流水线 ADC （ADCP）外设库示例 84

M

mac_audio_hi_res 8
MEB II 演示 207
mem_partition 86
Micrium uC/OS-III 演示 233
Micrium uC_OS_II 演示 232
msd_basic 349, 364
msd_sdcard 351
my_first_app 72
命令处理器系统服务示例 127

N

nvm_fat_single_disk 148
nvm_mpfs_single_disk 150
NVM 驱动程序演示 52
nvm_read_write 52
nvm_sdcard_fat_mpfs_multi_disk 152
nvm_sdcard_fat_multi_disk 155
NVM 外设库示例 100

O

object 170
oc_pwm 102
OPENRTOS 演示 238
osc_config 104

P

pic32_eth_web_server 276
pic32_eth_wifi_web_server 279
pic32_wifi_web_server 288
pmp_lcd 105
PMP 外设库示例 105
primitive 174
配置 MHC 144, 165, 273
配置硬件 4, 6, 8, 14, 17, 20, 24, 32, 35, 39, 42, 45, 46, 51, 53, 55, 57, 62, 64, 66, 67, 73, 74, 75, 77, 79, 81, 83, 85, 87, 88, 91, 93, 95, 96, 98, 100, 101, 103, 105, 106, 107, 109, 110, 111, 113, 115, 116, 118, 120, 121, 123, 124, 126, 128, 130, 132, 135, 136, 138, 139, 142, 144, 149, 151, 153, 155, 158, 160, 165, 169, 172, 176, 181, 183, 185, 200, 208, 209, 210, 213, 215, 218, 219, 221, 222, 223, 225, 226, 228, 229, 232, 234, 235, 236, 238, 239, 240, 241, 243, 244, 245, 246, 248, 253, 255, 256, 257, 259, 260, 263, 264, 266, 268, 269, 270, 271, 273, 277, 281, 289, 291, 293, 302, 306, 308, 316, 318, 320, 330, 333, 335, 336, 339, 340, 343, 346, 347, 349, 350, 352, 353, 355, 356, 358, 359, 361, 362, 363, 365
ADCHS 演示 （adchs_3ch_dma）75
ADCHS 演示 （adchs_oversample）77
ADCHS 演示 （adchs_pot）79
ADCHS 演示 （adchs_sensor）81
ADCHS 演示 （adchs_touchsense）83
ADC 演示 （adc_pot）73
ADC 演示 （adc_pot_dma）74
BMX 演示 （mem_partition）87
CAN 库演示 （echo_send）88
DMA 系统服务库演示 （dma_crc）139
DMA 演示 （dma_led_pattern）95
EBI 演示 （sram_read_write）96
I2C 驱动程序演示 （i2c_rtcc）51
I2C 演示 （i2c_interrupt）98
MEB II 演示 （gfx_camera）208
MEB II 演示 （gfx_cdc_com_port_single）209
MEB II 演示 （gfx_photo_frame）210
MEB II 演示 （gfx_web_server_nvm_mpfs）213
NVM 驱动程序演示 （nvm_read_write）53
NVM 演示 （flash_modify）101
PIC32 蓝牙协议栈库演示 （a2dp_avrcp）32
PIC32 蓝牙协议栈库演示 （data_basic）20
PIC32 蓝牙协议栈库演示 （data_temp_sens_rgb）24
PMP 演示 （pmp_lcd）106
RTCC 演示 （rtcc_alarm）115
RTOS 演示 /FreeRTOS （basic）223
RTOS 演示 /FreeRTOS （cdc_com_port_dual）225
RTOS 演示 /FreeRTOS （cdc_msd_basic）226
RTOS 演示 /FreeRTOS （gfx）228
RTOS 演示 /FreeRTOS （tcpip_client_server）229
RTOS 演示 /Micriμm （basic）232, 234
RTOS 演示 /Micriμm （gfx）235
RTOS 演示 /Micriμm （gfx_usb）236
RTOS 演示 /Micriμm （usb）238
RTOS 演示 /OPENRTOS （basic）239
RTOS 演示 /OPENRTOS （cdc_com_port_dual）240
RTOS 演示 /OPENRTOS （cdc_msd_basic）241
RTOS 演示 /OPENRTOS （gfx）243
RTOS 演示 /SEGGER embOS （basic）244
RTOS 演示 /SEGGER embOS （gfx）245
RTOS 演示 /SEGGER embOS （gfx_usb）246
RTOS 演示 /SEGGER embOS （usb）248
RTOS 演示 /ThreadX （basic）218
RTOS 演示 /ThreadX （gfx）219
RTOS 演示 /ThreadX （gfx_usb）221
RTOS 演示 /ThreadX （usb）222
SPI 驱动程序演示 （serial_eeprom）55
SPI 驱动程序演示 （spi_loopback）57
SPI 驱动程序演示 （spi_multislave）62
SPI 闪存驱动程序演示 （sst25vf020b）64
SPI 演示 （spi_loopback）116
SQI 演示 （flash_read_pio_mode）120
SQI 演示 （flash_read_xip_mode）121
TCPIP 演示 （berkeley_tcp_client）253
TCPIP 演示 （berkeley_tcp_server）255
TCPIP 演示 （berkeley_udp_client）256
TCPIP 演示 （berkeley_udp_relay）257
TCPIP 演示 （berkeley_udp_server）259
TCPIP 演示 （pic32_eth_wifi_web_server）281
TCPIP 演示 （pic32_wifi_web_server）289
TCPIP 演示 （snmpv3_nvm_mpfs）260
TCPIP 演示 （snmpv3_sdcard_fatfs）263
TCPIP 演示 （tcpip_tcp_client）264
TCPIP 演示 （tcpip_tcp_client_server）266
TCPIP 演示 （tcpip_tcp_server）268

TCPIP 演示 （tcpip_udp_client）269
TCPIP 演示 （tcpip_udp_client_server）270
TCPIP 演示 （tcpip_udp_server）271
TCPIP 演示 （web_net_server_nvm_mpfs）273
TCPIP 演示 （web_server_nvm_mpfs）277
TCPIP 演示 （web_server_sdcard_fatfs）291
TCPIP 演示 （wifi_easyconf）293
TCPIP 演示 （wifi_wolfssl_tcp_client）306
TCPIP 演示 （wifi_wolfssl_tcp_server）308
TCPIP 演示 （wolfssl_tcp_client）316
TCPIP 演示 （wolfssl_tcp_server）318
TMR 演示 （timer3_interrupt）123
USART 驱动程序演示 （usart_loopback）67
USART 演示 （uart_basic）124
USART 演示 （usart_echo）66
USB 设备演示 （audio_speaker）355
USB 设备演示 （cdc_com_port_dual）330
USB 设备演示 （cdc_com_port_single）333
USB 设备演示 （cdc_msd_basic）335
USB 设备演示 （cdc_serial_emulator）336
USB 设备演示 （cdc_serial_emulator_msd）339
USB 设备演示 （hid_basic）340
USB 设备演示 （hid_joystick）343
USB 设备演示 （hid_keyboard）346
USB 设备演示 （hid_mouse）347
USB 设备演示 （hid_msd_basic）349
USB 设备演示 （msd_basic）350
USB 设备演示 （msd_sdcard）352
USB 设备演示 （vendor_device）353
USB 主机演示 （cdc_basic）356
USB 主机演示 （cdc_msd）358
USB 主机演示 （hid_basic_mouse）361
USB 主机演示 （hub_cdc_hid）362
USB 主机演示 （hub_msd）363
USB 主机演示 （msd_basic）365
WDT 演示 （wdt_timeout）126
测试应用程序 （test_sample）320
电源演示 （deep_sleep_mode）111
电源演示 （sleep_mode）110
调试系统服务库演示 （debug_uart）130
调试系统服务库演示 （debug_usb_cdc_2）132
端口外设库演示 （cn_interrupt）109
端口演示 （blinky_leds）107
复位演示 （reset_handler）113
复位演示 （simple_comparator）91
复位演示 （triangle_wave）93
加密库演示 （encrypt_decrypt）45
加密库演示 （large_hash）46
流水线 ADC 演示 （adcp_cal）85
命令处理器系统服务库演示 （command_appio）128
器件控制系统服务库演示 （devcon_cache_clean）135
器件控制系统服务库演示 （devcon_cache_invalidate）136
器件控制系统服务库演示 （devcon_sys_config_perf）138
图形库演示 （object）172
图形库演示 （primitive）176
图形演示 （lcc）169
图形演示 （s1d13517）181
图形演示 （ssd1926）183
图形演示 （wvga_glcd）185
图形演示 （wvga_lcc）200
外部存储器编程器演示 （external_flash）142
外部存储器编程器演示 （sqi_flash）144
文件系统演示 （nvm_fat_single_disk）149
文件系统演示 （nvm_mpfs_single_disk）151
文件系统演示 （nvm_sdcard_fat_mpfs_multi_disk）153
文件系统演示 （nvm_sdcard_fat_multi_disk）155
文件系统演示 （sdcard_fat_single_disk）158
文件系统演示 （sdcard_msd_fat_multi_disk）160
音频演示 （audio_microphone_loopback）4
音频演示 （audio_tone）6
音频演示 （mac_audio_hi_res）8
音频演示 （universal_audio_decoders）14
音频演示 （usb_headset）17
振荡器演示 （osc_config）105
自举程序演示 （basic）35
自举程序演示 （LiveUpdate_uart_bootloader）39
配置硬件 \SQI 演示 （flash_read_dma_mode）118

Q

器件控制系统服务示例 134
其他蓝牙资源 19
驱动程序演示 50

R

reset_handler 112
rtcc_alarm 114
RTCC 外设库示例 114
RTOS 演示 217

S

s1d13517 180
sdcard_fat_single_disk 157
sdcard_msd_fat_multi_disk 159
SEGGER embOS 演示 243
segger_emwin 214
serial_eeprom 55
simple_comparator 91
sleep_mode 109
snmpv3_nvm_mpfs 259
snmpv3_sdcard_fatfs 262
spi_loopback 56, 116
spi_multislave 61
SPI 驱动程序演示 54
SPI 闪存驱动程序演示 63
SPI 外设库示例 115
sqi_flash 142
SQI 外设库示例 117
sram_read_write 95
ssd1926 182
sst25vf020b 64
设备 328
示例 72
使用 MHC 选择解码器 13
输出比较外设库示例 102
数据演示 19
输入捕捉外设库示例 99

T

TCP/IP 演示 249
tcpip_client_server 228
tcpip_tcp_client 263
tcpip_tcp_client_server 265
tcpip_tcp_server 267
tcpip_udp_client 268
tcpip_udp_client_server 269
tcpip_udp_server 271
test_sample 319
timer3_interrupt 122
triangle_wave 92
图形演示 162

U

uart_basic 124
universal_audio_decoders 11
usart_echo 65
usart_loopback 66
USART 驱动程序演示 65
USART 外设库示例 123
usb 221, 237, 247
USB HID 主机键盘和鼠标测试 325
USB MSD 主机 USB 笔式驱动器测试 324
usb_headset 16
USB 设备协议栈演示应用程序的程序和数据存储器要求 322
USB 设备协议栈组件的存储器要求 324
USB 演示 322

V

vendor 352

W

wdt_timeout 125
WDT 外设库示例 125
web_net_server_nvm_mpfs 272
web_server_nvm_mpfs 276
web_server_sdcard_fatfs 290
wifi_easy_configuration 292
wifi_g_demo 302
Wi-Fi 控制台命令 250
wifi_wolfssl_tcp_client 305
wifi_wolfssl_tcp_server 307
Wi-Fi 演示配置矩阵 249
wolfssl_tcp_client 315
wolfssl_tcp_server 317
wvga_glcd 184
wvga_lcc 199
外部存储器编程器演示 141
外设库示例 72
文件系统演示 148

X

系统服务库示例 127

Y

演示 3, 18, 34, 44, 50, 52, 54, 64, 65, 72, 75, 84, 86, 87, 91, 92, 94, 95, 97, 99, 100, 102, 104, 105, 107, 109, 112, 114, 115, 117, 122, 123, 125, 127, 129, 134, 139, 141, 148, 162, 207, 252, 328
- ADCHS 外设库 75
- ADC 外设库 72
- BMX 外设库 86
- CAN 库 87
- CVREF 外设库 92
- DMA 外设库 94
- DMA 系统服务库 139
- EBI 外设库 95
- I2C 驱动程序 50
- I2C 外设库 97
- MEB II 演示 207
- NVM 驱动程序 52
- NVM 外设库 100
- OSC 外设库 104
- PIC32 蓝牙协议栈库 19
- PMP 外设库 105
- RTCC 外设库 114
- SPI 驱动程序 54
- SPI 闪存驱动程序 64
- SPI 外设库 115
- SQI 外设库 117
- TCPIP 252
- TMR 外设库 122
- USART 驱动程序 65
- USART 外设库 123
- USB 328
- WDT 外设库 125
- 比较器外设库 91
- 编程器 141
- 电源外设库 109
- 调试系统服务库 129
- 端口外设库 107
- 复位外设库 112
- 加密库演示 44
- 流水线 ADC 外设库 84
- 命令处理器系统服务库 127
- 器件控制系统服务库 134
- 输出比较外设库 102
- 输入捕捉外设库 99
- 图形库 162
- 文件系统 148
- 音频演示 （audio_microphone_loopback）3
- 自举程序 34

演示功能 18
演示应用程序配置 326
音频演示 3
应用程序 319
应用程序帮助 2
应用程序函数和原型 188, 202

运行演示 4, 6, 9, 14, 17, 21, 25, 32, 37, 40, 43, 45, 48, 52, 53, 56, 60, 63, 65, 66, 70, 73, 74, 76, 77, 79, 81, 83, 85, 87, 89, 92, 93, 95, 96, 98, 100, 101, 103, 105, 106, 108, 109, 110, 111, 113, 115, 117, 119, 120, 121, 123, 125, 126, 128, 130, 133, 135, 137, 138, 140, 142, 147, 149, 151, 154, 156, 158, 160, 167, 169, 173, 179, 181, 183, 185, 201, 208, 209, 210, 213, 215, 219, 220, 221, 222, 224, 225, 226, 228, 229, 233, 234, 235, 236, 238, 239, 240, 241, 243, 244, 245, 247, 248, 254, 255, 256, 258, 259, 260, 263, 265, 267, 268, 269, 270, 272, 274, 277, 287, 289, 292, 300, 303, 306, 309, 316, 318, 330, 333, 336, 337, 339, 341, 343, 346, 348, 349, 351, 352, 353, 355, 356, 358, 360, 361, 362, 363, 365

- CAN 库演示 （echo_send）89
- DMA 系统服务库演示 （dma_crc）140
- I2C 驱动程序演示 （i2c_rtcc）52
- MEB II 演示 （gfx_camera）208
- NVM 驱动程序演示 （nvm_read_write）53
- PIC32 蓝牙协议栈库演示 （a2dp_avrcp）32
- PIC32 蓝牙协议栈库演示 （data_basic）21
- PIC32 蓝牙协议栈库演示 （data_temp_sens_rgb）25
- RTOS 演示 /FreeRTOS （basic）224
- RTOS 演示 /FreeRTOS （cdc_com_port_dual）225
- RTOS 演示 /FreeRTOS （cdc_msd_basic）226
- RTOS 演示 /FreeRTOS （gfx）228
- RTOS 演示 /FreeRTOS （tcpip_client_server）229
- RTOS 演示 /Micriμm （basic）233, 234
- RTOS 演示 /Micriμm （gfx）235
- RTOS 演示 /Micriμm （gfx_usb）236
- RTOS 演示 /OPENRTOS （basic）239
- RTOS 演示 /OPENRTOS （cdc_com_port_dual）240
- RTOS 演示 /OPENRTOS （cdc_msd_basic）241
- RTOS 演示 /OPENRTOS （gfx）243
- RTOS 演示 /SEGGER embOS （basic）244
- RTOS 演示 /SEGGER embOS （gfx）245
- RTOS 演示 /SEGGER embOS （gfx_usb）247
- RTOS 演示 /SEGGER embOS （usb）248
- RTOS 演示 /ThreadX （basic）219
- RTOS 演示 /ThreadX （gfx）220
- RTOS 演示 /ThreadX （gfx_usb）221
- RTOS 演示 /ThreadX （usb）222
- SPI 驱动程序演示 （serial_eeprom）56
- SPI 闪存驱动程序演示 （sst25vf020b）65
- TCPIP 演示 （berkeley_tcp_client）254
- TCPIP 演示 （berkeley_tcp_server）255
- TCPIP 演示 （berkeley_udp_client）256
- TCPIP 演示 （berkeley_udp_relay）258
- TCPIP 演示 （berkeley_udp_server）259
- TCPIP 演示 （pic32_eth_wifi_web_server）287
- TCPIP 演示 （pic32_wifi_web_server）289
- TCPIP 演示 （snmpv3_nvm_mpfs）260
- TCPIP 演示 （snmpv3_sdcard_fatfs）263
- TCPIP 演示 （tcpip_tcp_client）265
- TCPIP 演示 （tcpip_tcp_client_server）267
- TCPIP 演示 （tcpip_tcp_server）268
- TCPIP 演示 （tcpip_udp_client）269
- TCPIP 演示 （tcpip_udp_client_server）270
- TCPIP 演示 （tcpip_udp_server）272
- TCPIP 演示 （web_net_server_nvm_mpfs）274
- TCPIP 演示 （web_server_nvm_mpfs）277
- TCPIP 演示 （web_server_sdcard_fatfs）292
- TCPIP 演示 （wifi_easyconf）300
- TCPIP 演示 （wifi_wolfssl_tcp_client）306
- TCPIP 演示 （wifi_wolfssl_tcp_server）309
- TCPIP 演示 （wolfssl_tcp_client）316
- TCPIP 演示 （wolfssl_tcp_server）318
- USART 演示 （usart_echo）66
- USB 设备演示 （audio_speaker）355
- USB 设备演示 （cdc_com_port_dual）330
- USB 设备演示 （cdc_com_port_single）333
- USB 设备演示 （cdc_msd_basic）336
- USB 设备演示 （hid_basic）341
- USB 设备演示 （hid_joystick）343
- USB 设备演示 （hid_keyboard）345, 346
- USB 设备演示 （hid_mouse）348
- USB 设备演示 （hid_msd_basic）349
- USB 设备演示 （msd_basic）351
- USB 设备演示 （msd_sdcard）352
- USB 设备演示 （vendor_device）353
- USB 演示 （cdc_serial_emulator）337
- USB 演示 （cdc_serial_emulator_msd）339
- USB 主机演示 （cdc_basic）356
- USB 主机演示 （cdc_msd）358
- USB 主机演示 （hid_basic_mouse）361
- USB 主机演示 （hub_cdc_hid）362
- USB 主机演示 （hub_msd）363
- USB 主机演示 （msd_basic）365
- 调试系统服务库演示 （debug_uart）130
- 调试系统服务库演示 （debug_usb_cdc_2）133
- 端口外设库演示 （cn_interrupt）109
- 加密库演示 （encrypt_decrypt）45
- 加密库演示 （large_hash）48
- 命令处理器系统服务库演示 （command_appio）128
- 器件控制系统服务库演示 （devcon_cache_clean）135
- 器件控制系统服务库演示 （devcon_cache_invalidate）137
- 器件控制系统服务库演示 （devcon_sys_config_perf）138
- 输出比较演示 （oc_pwm）103
- 输入捕捉演示 （ic_basic）100
- 图形库演示 （object）173
- 图形库演示 （primitive）179
- 图形演示 （s1d13517）181
- 图形演示 （ssd1926）183
- 图形演示 （wvga_glcd）185
- 图形演示 （wvga_lcc）201
- 外部存储器编程器演示 （external_flash）142
- 外部存储器编程器演示 （sqi_flash）147
- 文件系统演示 （nvm_fat_single_disk）149
- 文件系统演示 （nvm_mpfs_single_disk）151
- 文件系统演示 （nvm_sdcard_fat_mpfs_multi_disk）154
- 文件系统演示 （nvm_sdcard_fat_multi_disk）156
- 文件系统演示 （sdcard_fat_single_disk）158
- 文件系统演示 （sdcard_msd_fat_multi_disk）160
- 音频演示 （audio_microphone_loopback）4
- 音频演示 （audio_tone）6
- 音频演示 （mac_audio_hi_res）9
- 音频演示 （universal_audio_decoders）14
- 音频演示 （usb_headset）17
- 自举程序演示 （basic）37
- 自举程序演示 （LiveUpdate_uart_bootloader）40

运行演示 \ADCHS 演示 （adchs_3ch_dma）76

运行演示 \ADCHS 演示 （adchs_oversample）77
运行演示 \ADCHS 演示 （adchs_pot）79
运行演示 \ADCHS 演示 （adchs_sensor）81
运行演示 \ADCHS 演示 （adchs_touchsense）83
运行演示 \ADC 演示 （adc_pot）73
运行演示 \ADC 演示 （adc_pot_dma）74
运行演示 \BMX 演示 （mem_partition）87
运行演示 \DMA 演示 （dma_led_pattern）95
运行演示 \ 电源演示 （deep_sleep_mode）111
运行演示 \ 电源演示 （sleep_mode）110
运行演示 \ 端口演示 （blinky_leds）108
运行演示 \EBI 演示 （sram_read_write）96
运行演示 \ 复位演示 （reset_handler）113
运行演示 \ 复位演示 （simple_comparator）92
运行演示 \ 复位演示 （triangle_wave）93
运行演示 \I2C 演示 （i2c_interrupt）98
运行演示 \ 流水线 ADC 演示 （adcp_cal）85
运行演示 \MEB II 演示 （gfx_cdc_com_port_single）209
运行演示 \MEB II 演示 （gfx_photo_frame）210
运行演示 \MEB II 演示 （gfx_web_server_nvm_mpfs）213
运行演示 \NVM 演示 （flash_modify）101
运行演示 \PMP 演示 （pmp_lcd）106
运行演示 \RTCC 演示 （rtcc_alarm）115
运行演示 \SPI 演示 （spi_loopback）117
运行演示 \SQI 演示 （flash_read_dma_mode）119
运行演示 \SQI 演示 （flash_read_pio_mode）120
运行演示 \SQI 演示 （flash_read_xip_mode）121
运行演示 \ 输出比较演示 （oc_pwm）103
运行演示 \ 输入捕捉演示 （ic_basic）100
运行演示 \TMR 演示 （timer3_interrupt）123
运行演示 \ 图形演示 （lcc）169
运行演示 \USART 演示 （uart_basic）125
运行演示 \WDT 演示 （wdt_timeout）126
运行演示 \ 振荡器演示 （osc_config）105
运行应用程序 320
    SPI 驱动程序演示 （spi_loopback）60
    SPI 驱动程序演示 （spi_multislave）63
    USART 驱动程序演示 （usart_loopback）70
    测试应用程序 （test_sample）320

Z

振荡器外设库示例 103
主机 354
自举程序演示 34