

瑞萨 M16C/6C Starter Kit

USB 例程 用户手册

瑞萨单片机

M16C 族/M16C/60 系列

本资料所记载的内容，均为本资料发行时的信息，瑞萨电子对于本资料所记载的产品或者规格可能会作改动，恕不另行通知。
请通过瑞萨电子的主页确认发布的最新信息。

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

产品使用时的注意事项

本文对适用于单片机所有产品的“使用时的注意事项”进行说明。有关个别的使用时的注意事项请参照正文。此外，如果在记载上有与本手册的正文有差异之处，请以正文为准。

1. 未使用的引脚的处理

【注意】将未使用的引脚按照正文的“未使用引脚的处理”进行处理。

CMOS产品的输入引脚的阻抗一般为高阻抗。如果在开路的状态下运行未使用的引脚，由于感应现象，外加LSI周围的噪声，在LSI内部产生穿透电流，有可能被误认为是输入信号而引起误动作。未使用的引脚，请按照正文的“未使用引脚的处理”中的指示进行处理。

2. 通电时的处理

【注意】通电时产品处于不定状态。

通电时，LSI内部电路处于不确定状态，寄存器的设定和各引脚的状态不定。通过外部复位引脚对产品进行复位时，从通电到复位有效之前的期间，不能保证引脚的状态。

同样，使用内部上电复位功能对产品进行复位时，从通电到达到复位产生的一定电压的期间，不能保证引脚的状态。

3. 禁止存取保留地址（保留区）

【注意】禁止存取保留地址（保留区）

在地址区域中，有被分配将来用作功能扩展的保留地址（保留区）。因为无法保证存取这些地址时的运行，所以不能对保留地址（保留区）进行存取。

4. 关于时钟

【注意】复位时，请在时钟稳定后解除复位。

在程序运行中切换时钟时，请在要切换成的时钟稳定之后进行。复位时，在通过使用外部振荡器（或者外部振荡电路）的时钟开始运行的系统中，必须在时钟充分稳定后解除复位。另外，在程序运行中，切换成使用外部振荡器（或者外部振荡电路）的时钟时，在要切换成的时钟充分稳定后再进行切换。

5. 关于产品间的差异

【注意】在变更不同型号的产品时，请对每一个产品型号进行系统评价测试。

即使是同一个群的单片机，如果产品型号不同，由于内部ROM、版本模式等不同，在电特性范围内有时特性值、动作容限、噪声耐量、噪声辐射量等也不同。因此，在变更不认同型号的产品时，请对每一个型号的产品进行系统评价测试。

目 录

1.	前言	5
2.	介绍	6
3.	开发环境	7
3.1	例程配置	7
3.2	目标例程选项	7
3.3	主机应用程序软件	7
4.	USB Stack (Target)	8
4.1	硬件抽象层	8
4.2	USB Core	9
4.3	人机接口设备类	10
4.4	通信设备类	11
4.5	海量存储类	12
5.	应用程序	13
5.1	应用程序介绍	13
5.2	人机接口应用程序	13
5.3	通信设备类应用程序	15
5.4	海量存储类示范	18
5.5	LibUSB	19
6.	附加信息	22

1. 前言

注意

本资料内容可能全部或部分改动，恕不另行通知。

版权所有。未经 Renesas Electronics Europe Limited 书面允许，禁止以任何形式复制本文档的全部或部分内
容。

注册商标

本手册使用的所有商标和产品名称都是公司/组织的商标/注册商标。

版权

© 2010 Renesas Electronics Europe Limited. All rights reserved.

© 2010 Renesas Electronics Corporation. All rights reserved.

© 2010 Renesas Solutions Corporation. All rights reserved.

网址: <http://www.renesas.com/>

术语表

ADC模数转换

CDC 通信设备类

HAL 硬件抽象层

HID 人机接口设备

LCD 液晶屏

LED 发光二极管

MSC海量存储类

RSK Renesas Starter Kit

USB 通用串行总线

2. 介绍

RSK USB 例程为开发者提供向一个系统中增加 USB 设备功能的基本应用。例程包含了用于三种最常用的 USB 设备类的应用例程（注 1）：

- 人接机口设备(HID)
- 通信设备类——抽象控制模型(CDC-ACM)
- 海量存储类(MSC)

除了这三类已定义的USB类，例程还包含一个LibUSB的例子。[LibUSB](#)是一个开源的工程，目的是提供一个可以被用户应用程序用来直接访问USB设备而与操作系统无关的库。这里提供的例子使用的是微软Windows XP主机。

注 1：细节参见特定章节。试验时请使用运行 MS Windows XP SP2 的 USB 主机 PC。HID 和 LibUSB 示例都包含 PC 应用程序。

嵌入式软件是用 ANSI C 语言写的源码，无需操作系统的支持。

主机端应用软件是用 MS Windows 下的 Visual C++写的源码。

本手册介绍 USB 示例。快速入门向导和手册指南详细说明了软件的安装和调试环境。

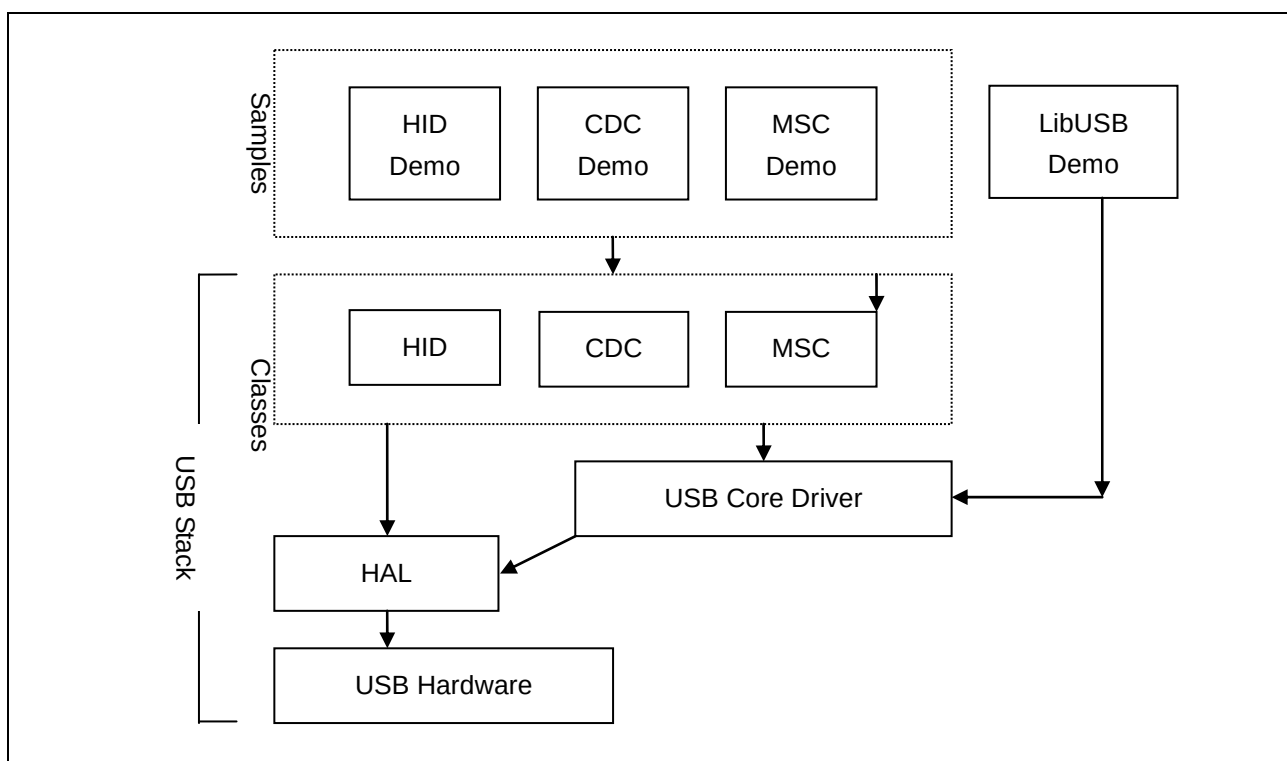


图1 嵌入式软件，包括 USB Stack 层

3. 开发环境

3.1 例程配置

例程以 RSK 的工程向导方式提供。如果要创建例程工程，请参照《RSK Quick Start Guide》。

创建例程的同时会产生目标工程和主机端工程（如果是可应用的）的源码，还有一些配置的驱动文件。

3.2 目标例程选项

开发 USB 软件时，能够在运行时得到调试信息而不需要停止执行代码——就像在调试器中单步运行一样是很有意义的。USB Stack 软件的所有模块都包含调试信息，这种调试信息可以在支持 `printf` 函数的系统中使用。示例应用程序都支持 `printf` 并且输出结果可以通过 RSK 的串口在 PC 端显示。查看串口输出结果需要遵循如下配置：

波特率：57600。数据：8 位。奇偶校验：无。停止位：2 位。数据流控制：无。

用户可以通过“`#define DEBUG_LEVEL`”来设定调试信息的等级。这部分内容将在 USB Stack 路径下的 `usb_common.h` 文件中说明。注意等级较高的调试信息会明显降低系统速度。如果要清除所有调试信息，请使用“`#define RELEASE`”。

3.3 主机应用程序软件

如果要创建微软 Visual C++ 应用程序，用户需要先安装合适的 Windows Software Development Kit (SDK) 和 Windows Driver Development Kit (DDK)。这些开发套件提供了访问 USB 设备时对库函数的调用，可以直接从微软获得。

下面是基于 XP 系统开发包的下载链接，但是不保证这些链接的适用性或者准确性。要创建应用程序必须安装两个文件，建议先安装 SDK。

www.microsoft.com/downloads

[Windows SDK](#)

[Windows DDK](#)

LibUSB 例程需要 LibUSB-Win32 提供的文件。详细情况请参见“5.5 LibUSB”。

4. USB Stack (Target)

USB 软件以三层组成的 USB Stack 的形式实现。

Stack 的顶层是由 HID, CDC 和 MSC 组成的 USB 设备类, 后面会介绍这些类。

中间是核心层 (USB Core), USB Core 处理标准的设备请求。底层是硬件抽象层 (HAL), 硬件抽象层为上层提供了独立于硬件的 API。

这种模块化的设计意味着即使开发一个私有的 USB 接口也可以使用该软件。例如通过调用 USB HAL API 和 USB Core API 开放的函数开发一个私有模块。

4.1 硬件抽象层

HAL 是一个硬件特性的层, 提供了非硬件特性的 API。HAL 支持下列传输模式:

控制 (配置, 数据输入/输出, 状态)

批量 (输入和输出)

中断 (输入)

有些 HAL 并不能支持所有的模式, 但是 M16C/6C 支持。

HAL 处理 USB 挂起和唤醒信号的方式是通过 USB_HAL.h 文件中的 #define 来控制的。如果定义了 “USB_SUSPEND_LOW_POWER”, 当 USB 总线挂起时 HAL 会自动将 MCU 设置为低功耗模式, 而当 USB 总线唤醒时 HAL 自动将 MCU 恢复到正常的功耗模式。当设备是总线供电时需要执行该操作, 因为 USB 协议限制了挂起状态电流最大为 500 μ A。此时选择的低功耗模式是 “等待模式”, 同时 PLL 也被禁止。该软件假定 PLL 作为 CPU 时钟。

下面是组成 USBHAL API 的函数列表。

名称	描述
USBHAL_Init	初始化 HAL。注册回调函数。如果使用 USB 核心层那么该操作自动执行。
USBHAL_Config_Get	得到当前 HAL 配置。
USBHAL_Config_Set	设定当前 HAL 配置。
USBHAL_Control_ACK	通过控制输入管道产生一个 ACK 信号。(在配置包之后发送)
USBHAL_Control_IN	通过控制输入管道发送数据。(在配置包之后发送)
USBHAL_Control_OUT	通过控制输出管道接收数据。(在配置包之后发送)
USBHAL_Bulk_IN	通过批量输入管道发送数据。
USBHAL_Bulk_OUT	通过批量输出管道接收数据。
USBHAL_Interrupt_IN	通过中断输入管道发送数据。
USBHAL_Reset	复位该模块。(在发生错误后发送)。
USBHAL_Control_Stall	停止控制管道。(在配置包之后发送)
USBHAL_Bulk_IN_Stall	停止批量输入管道。
USBHAL_Bulk_OUT_Stall	停止批量输出管道。
USBHAL_Interrupt_IN_Stall	停止中断输入管道。
USBHALInterruptHandler	系统必须已经建立, 因此当任意一个 USB 中断发生时可以调用该函数。

HAL 模块包含下列文件:

usb_hal.c

usb_hal.h

usb_common.h

4.2 USB Core

在枚举阶段 USB Core 层处理对所有 USB 设备来说都是通用的标准 USB 请求。这意味着开发者可以专注于任何一个类或厂商特定的应用。USB Core 需要用当前设备的描述符进行初始化。USB Core 通过 USBHAL 访问特殊的 HW，而 USBHAL 的初始化是由 USB Core 实现的。

Get_Descriptor 请求处理如下：

- 设备
- 配置
- 字符
 - 语言 ID（当前只支持英语）
 - 厂商
 - 产品
 - 序列号

USBCDC API 包含一个叫做“USBCORE_Init”的函数。这个函数初始化 USB Core 和 HAL。除了把设备描述符传递给这个函数以外，在下述情况下还需要回调该函数：

- 收到了该层不能处理的配置包。这使得更高的一层来处理类或者厂商特定的请求。
- 更高层处理完配置包之后，控制数据输出也已完成。
- USB 线连接或断开。
- 发生了未处理错误。

HAL 模块包含下列文件：

usb_core.c
usb_core.h
usb_common.h

4.3 人机接口设备类

顾名思义 HID 设备类指通常用于人的操作发起通信请求的键盘、鼠标和游戏杆等，然而并非仅限如此。HID 类适用于任何设备，只要该设备实现通信的方式是在大小预先确定的“报告”中发送数据，并且对数据的传输速率不做严格要求。

HID 类已经获得微软 Windows98 的支持。这种支持包括内核层面和用户层面。当一个 HID 类型的设备插入到 Window PC 时，它将自动被识别。Windows 会给该设备安装它自己的驱动，因此无需开发客户 Windows 驱动甚至 Windows“inf”文件。

这种 HID 类的使用支持单个 IN 报告和单个 OUT 报告。同时也支持向主机端发送报告的中断 IN 和控制 IN（通过 Get_Report）传输。主机端发出的报告必须使用控制 OUT（通过 Set_Report）。

下列是组成 USBHID API 的函数。

名称	描述
USBHID_Init	初始化 HID 模块。当从主机端收到一个报告时，注册一个回调函数。提供发送到主机端报告的最初始内容。初始化 Core 和 HAL 层。
USBHID_ReportIN	使用中断 IN 传输向主机端发送报告。

HID 模块包含下列文件：

- usb_hid.c
- usb_hid.h
- usb_descriptors.c
- usb_descriptors.h
- usb_common.h

4.4 通信设备类

CDC ACM 允许主机把一个设备当做标准串行口（COM）。在留用使用串行通讯的应用程序时，该特征非常有用。可以采用批量 IN 和批量 OUT 传输来传输所有的 non-setup 数据。

CDC 模块使用 USB Core 层处理所有的标准请求。除此之外，CDC 模块还处理下列类请求：

GET_LINE_CODING

SET_LINE_CODING(MS 超级终端所要求的)

SET_CONTROL_LINE_STATE

MS Windows 也支持 CDC 类，因此无需开发客户的 Windows 内核驱动。尽管如此用户“inf”文件是需要的，示例的 CDC 应用程序就包含这样一个文件。当一个 CDC ACM 设备插入到 Windows PC 时，额外的（虚拟的）一个 COM 口变得可用，因此应用程序可以把它当做一个标准 COM 端口使用。

下列是组成 USBCDC API 的函数。

名称	说明
USBCDC_Init	初始化 CDC 模块。也初始化 Core 和 HAL 层。
USBCDC_IsConnected	返回设备的连接状态。
USBCDC_WriteString	向主机端发送字符串的模块功能。
USBCDC_PutChar	向主机端发送字符的模块功能。
USBCDC_GetChar	从主机端获取字符的模块功能。
USBCDC_Write	向主机端发送已有数据缓冲的模块功能。
USBCDC_Write_Async	启动异步的向主机端写数据缓冲。回调该函数用以表明操作已完成。
USBCDC_Read	从主机端读数到已有数据缓冲中的模块功能。
USBCDC_Read_Async	启动异步的从主机端读数到数据缓冲。回调该函数用以表明操作已完成。
USBCDC_Cancel	取消待定的异步操作。

CDC 模块包含下列文件：

usb_cdc.c

usb_cdc.h.

usb_descriptors.c

usb_descriptors.h

usb_common.h

4.5 海量存储类

MSC 类广受欢迎，例如用来跟 PC 共享数据的相机和 U 盘。MSC 类成功的原因是，当设备插入到主机 PC 时，MSC 类对主机来说相当于另一个设备，因此用户可以使用像 Windows Explorer 之类的应用程序来存取数据。从 Windows 2000 开始，不需要用户的“inf”驱动文件就可以支持 MSC 类。

批量 IN 和批量 OUT 传输用来传输 non-setup 数据。

MSC 模块使用 USB Core 层处理所有的标准请求。除此之外，它还处理下列类请求：

BULK_ONLY_MASS_STORAGE_RESET

GET_MAX_LUN

除了支持标准的 USB 协议外，MSC 设备必须支持一套 SCSI 命令。所有 SCSI 命令都是批量 OUT 传输时在 Command Block Wrapper (CBW) 中打包发送的。接下来的数据阶段可遵循任一方向，为完成 SCSI 命令，设备以 Command Status Wrapper (CSW) 的形式发送状态响应。支持的 SCSI 命令如下所示：

INQUIRY

READ_CAPACITY10

READ10

REQUEST_SENSE

TEST_UNIT_READY

WRITE10

VERIFY10

PREVENT_ALLOW_MEDIUM_REMOVAL(Optional)

MODE_SENSE6(可选，支持有限)

USBMSC API 包含一个叫做“USBMSC_Init”的单个函数。该函数初始化 MSC 模块并且也初始化 USB Core 和 HAL 层。

MSC 类的执行直接访问简单的 RAM 磁盘块设备，该设备使用 RSK 的 24KB RAM。换言之，MSC 类跟 MSC 应用程序之间没有分隔。因此当使用不同的存储器时，需要更换应用程序而不是更换示例 RAM 磁盘。

MSC 模块包含下列文件：

usb_msc.c

usb_msc.h

usb_msc_scsi.c

usb_msc_scsi.h

ram_disk.c

ram_disk.h

usb_descriptors.c

usb_descriptors.h

usb_common.h

5. 应用程序

5.1 应用程序介绍

接下来的章节介绍一些示例应用程序，这些应用程序可以用来示范各个 USB 解决方案。HID 和 LibUSB 工程需要专用的主机端应用程序，同时提供可执行程序 and 源码。CDC 和 MSC 工程使用标准的 Windows 应用程序。

所有的应用软件都要求已经使用适合于该应用程序的例子代码对 RSK 进行了编程。对 RSK 进行编程的细节已经在产品手册中作了介绍。要获得手册的电子版，请访问该网址www.renesas.com/renesas_starter_kits 并从列表中选择所用的 RSK。

5.2 人机接口应用程序

HID 主机示例应用程序 RSK_HID 是为 Windows 主机 PC 编写的。

预编译连接的可执行文件也由工程产生器提供。在工程中到 release 目录下运行 RSK_HID.exe，会显示如下窗口：

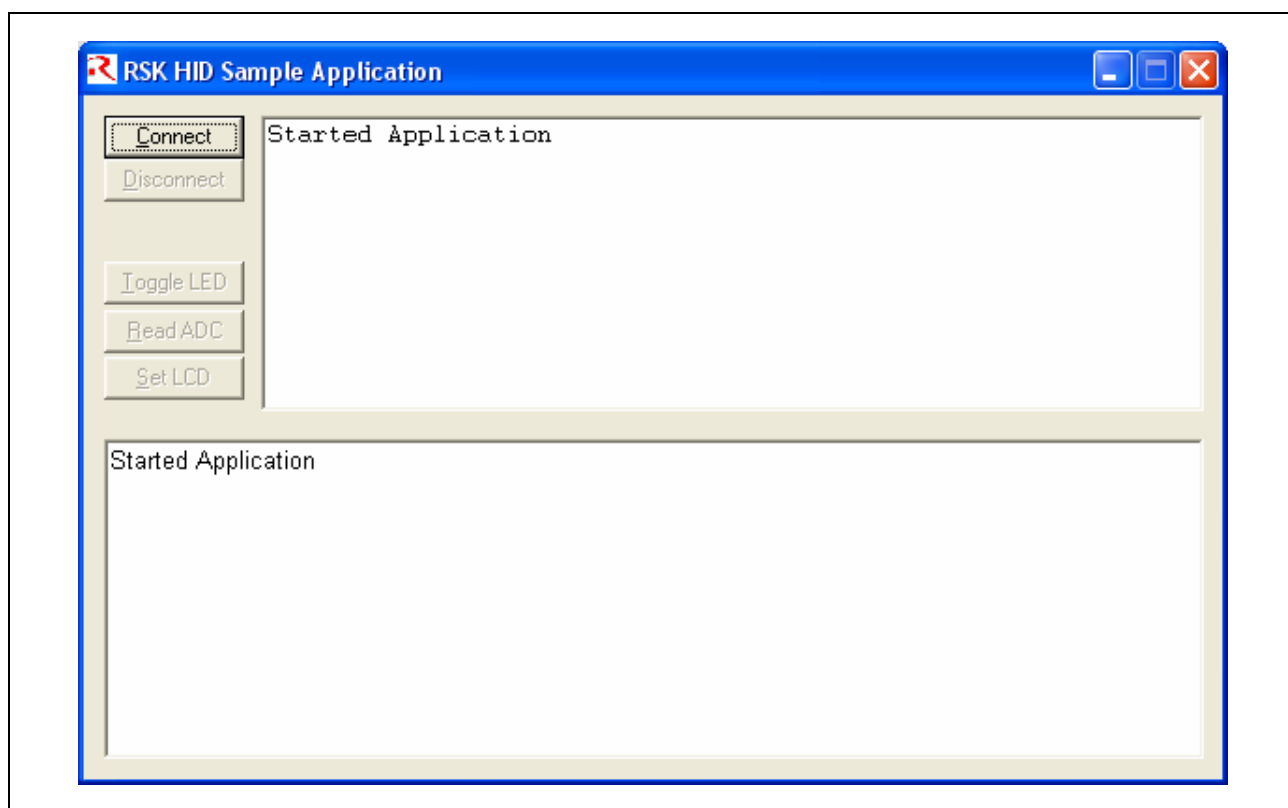


图2 HID 主机 PC 应用程序

对 RSK 进行编程并运行代码。用 USB 线连接 PC 和 RSK。设备第一次连接到某一 USB 端口时，Windows 会检测到新设备并自动加载固有的 HID 类驱动。

当 Windows 已经完成了枚举过程，用户需要把应用程序跟目标连接起来。点击“连接”按钮，电脑会提示确认要连接设备的 VID 和 PID。如果没有在 RSK 的固件里使用你自己的 VID 和 PID，那么显示缺省值。当成功连接后，设备的信息会显示出来，而且剩下的按钮被使能。

“Toggle LED”按钮使得 RSK 上的 LED 灯闪烁。

“Read ADC”按钮会向 RSK 发送读 ADC 的命令，并把读到的值返回到 PC 端然后显示出来。

“Set LCD”按钮可以变换 RSK 的 LCD 上的文字。

为了演示 RSK 产生通信请求，可以按 RSK 上的一个按键，这个信息将会返回到主机端并显示在对话框里。这表明 PC 和 RSK 之间可以成功的发送输入和输出 HID 报告。报告的格式如下：

Input Report:**Byte 1**

Bit 0 = LED status.

Bit 1 = ADC value valid indicator.

Bit 2 = Switch pressed indicator.

Byte 2-5 = 32 bit, little endian ADC Value.

Output Report:**Byte 1**

Bit 0 = LED toggle request.

Bit 1 = ADC read request.

Bit 2 = LCD set request.

Byte 2-17 = 16 ASCII Characters for LCD.

当 RSK 上有一个按键按下时或者主机请求一个报告时，应用程序会发送一个输入报告。

当用户点击对话框上的一个按钮时，应用程序会发送一个输出报告。

跟 USB 相关的 HID 应用程序功能包含下列文件：

Target:

usb_hid_app.c

usb_hid_app.h

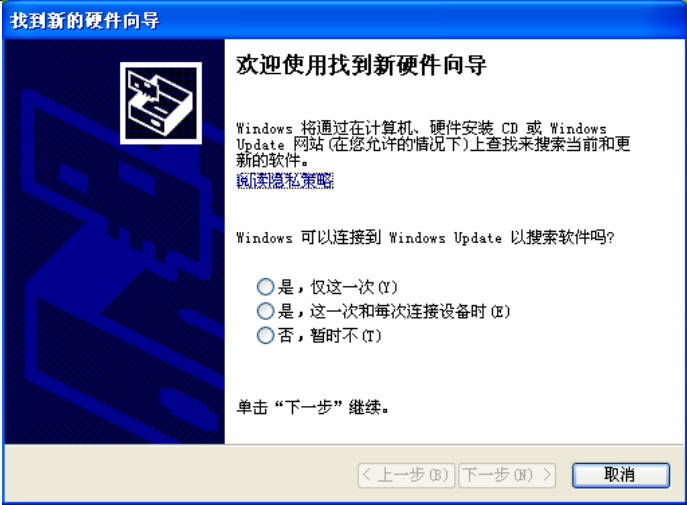
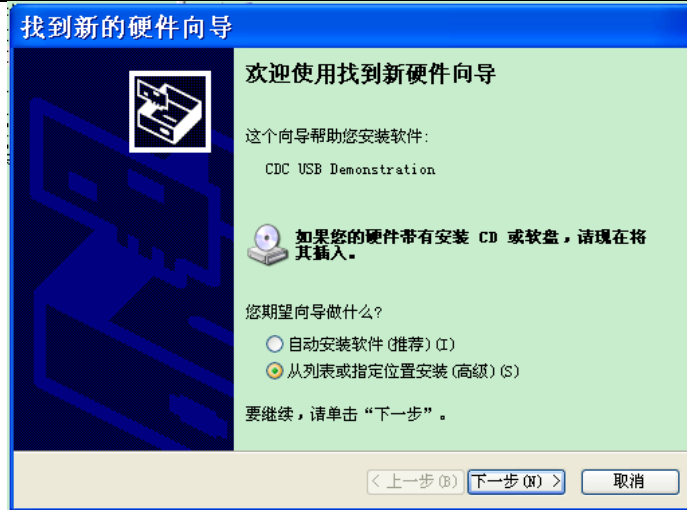
Host Application:

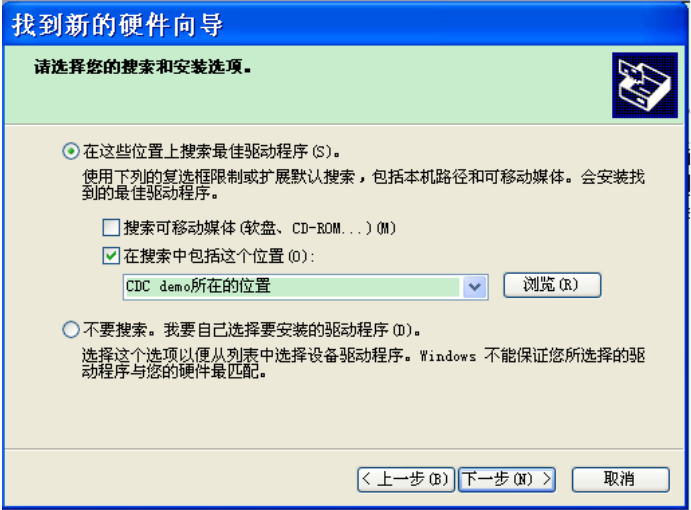
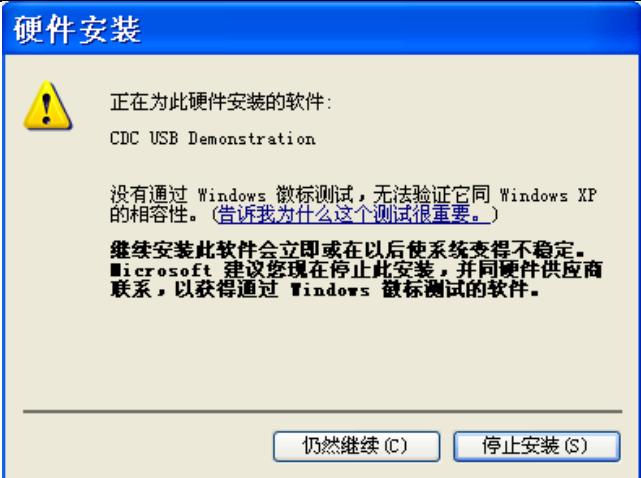
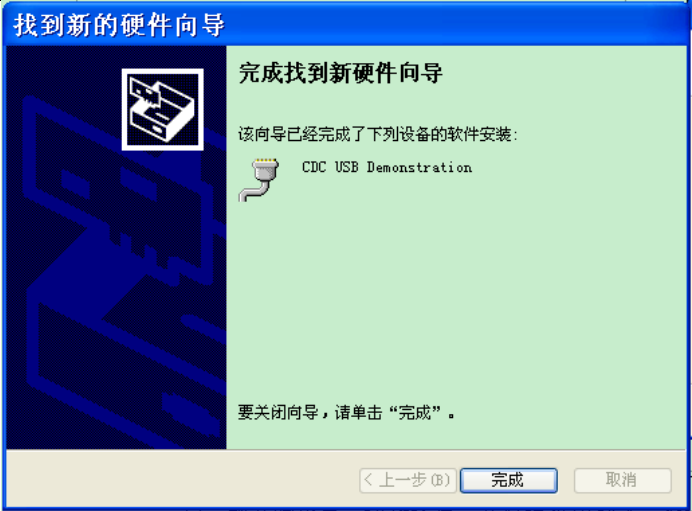
\\PC\\RSK_HID\\...

5.3 通信设备类应用程序

CDC 示例应用程序示范了使用一个标准的终端程序实现设备跟 Windows PC 之间的通信。Windows 提供了一个叫做超级终端的应用程序。如果有需要也可以使用其他任何一个串行终端程序。

用 CDC 应用程序对 RSK 进行编程并按照 RSK 指南手册上描述的来运行程序。用一根 USB 线把 PC 跟 RSK 连接起来。对于一个特定的 USB 端口，设备第一次连接的时候，Windows 会检测到新设备并且执行“找到新硬件向导”：

	
Windows 会出现如下对话框，你可以选择“否，暂时不”。	
在如下对话框中，点击“从列表或指定位置安装（高级）”来选择正确的“inf”文件。	

输入或者浏览到你所创建和编译的 CDC 工程位置和 host/driver 文件夹。	
点击“下一步”安装 CDC 支持。 安装过程中可能会提示如下警告。 请选择“仍然继续”来安装驱动。 请访问微软网站获取 Windows Logo Testing 程序的更多内容。	
然后 Windows 完成了 CDC USB 驱动的安装。	

对 Windows 应用程序来说多了一个 COM 口。为了看到已分配的端口，可以打开 Windows 设备管理器窗口。到开始菜单，选择运行。在对话框中敲入“devmgmt.msc”。这样设备管理器就打开了。展开串口组，则已安装的端口都列出来了。当串口终端程序连接到该 COM 端口（注 2）时，将会从 RSK 重复收到一个信息写着：

“Renesas USB CDC Sample, Press Switch SW1.”

注 2：诸如波特率和奇偶校验等配置条件对该虚拟 COM 口来说无关紧要。

按下 RSK 上的 SW1 将会停止该重复信息并且显示如下主菜单。

为了演示双向通信，按下 SW2 将 RSK 设定到 echo 模式。在这种模式下，敲到超级终端上的任何内容都会被 RSK 读取并且返回至超级终端。按下 SW3 退出该 echo 模式。

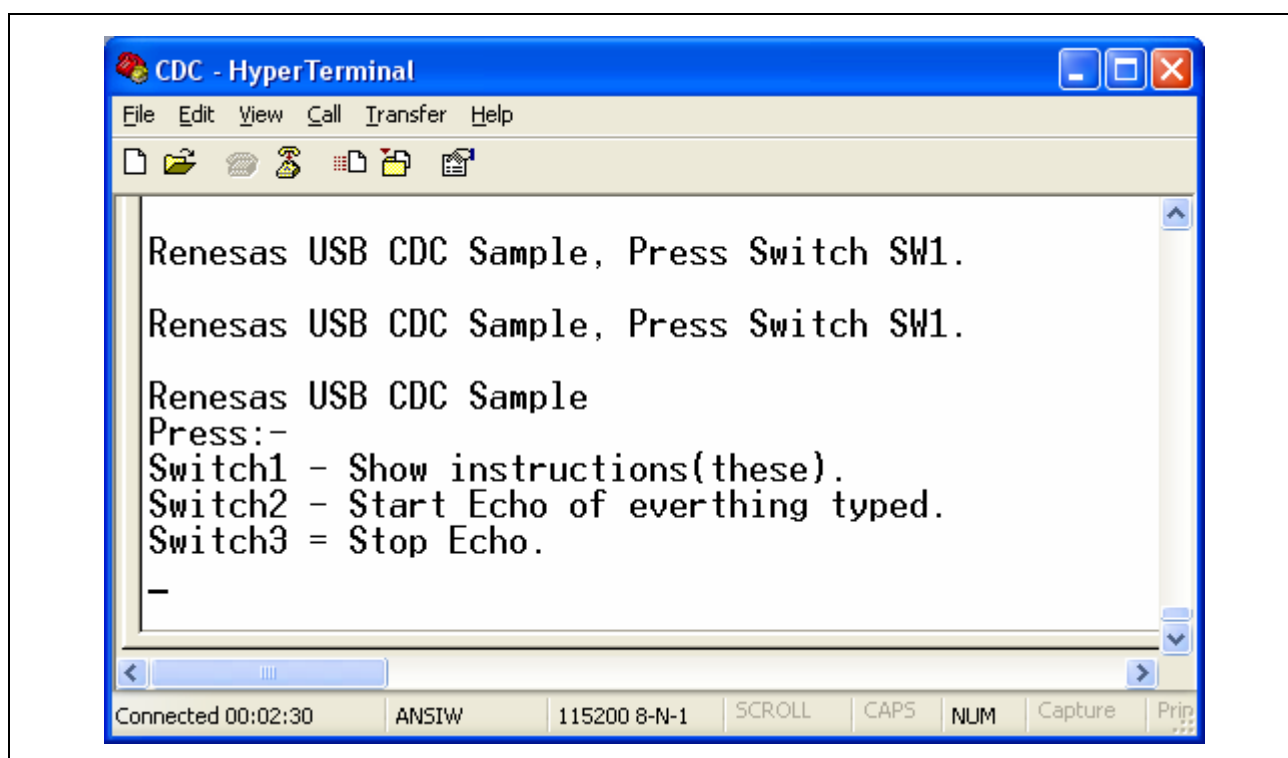


图3 串行通信对话框

跟 USB 相关的 CDC 应用程序功能包括下列文件：

Target:

usb_cdc_app.c

usb_cdc_app.h

Host:

\\PC\\CDC_Demo.inf

5.4 海量存储类示范

MSC 示例演示了主机如何把一个 MSC 设备视作外部设备。不需要额外的应用程序来支持 MSC，因为 MSC 是由 Windows XP 内部支持的。

启动运行 RSK 上的 MSC 示例应用程序，用一根 USB 线把 RSK 跟 Windows PC 连接起来。

使用 Windows 资源管理器或类似工具，查看 Windows 新增加的驱动器。该驱动器显示 RSK 上的示例 RAM Disk 的内容。内容以 FAT 文件格式显示，卷标名称是“RENESAS”。数据的可用空间大小是 4KB。包括一个叫做 Renesas.txt 的例子文件，该文件可以打开、编辑和保存。就像一个正常的驱动器一样，你也可以向该驱动器拷入文件，但是注意当 RSK 断电时，RAM Disk 的内容就丢失了。

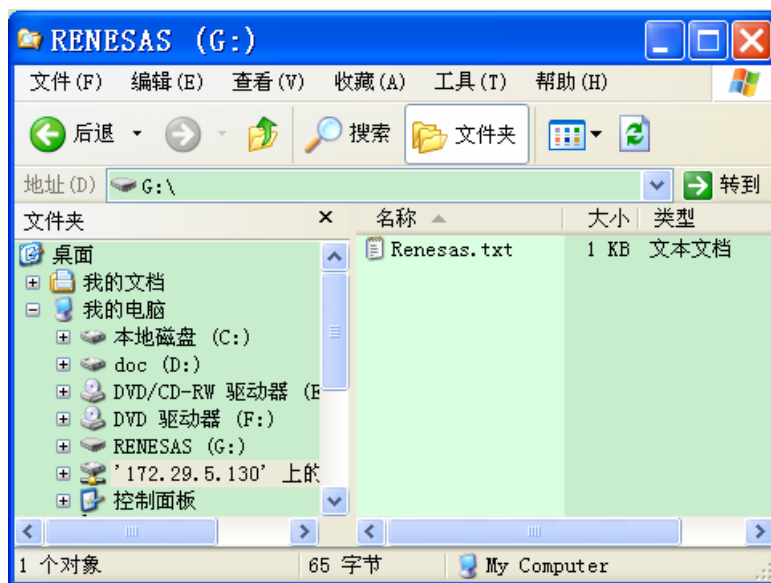


图4 Windows 资源管理器显示新的磁盘驱动器映射

“ram_disk.c”文件中有一个选项可以防止 RAM 磁盘使用文件系统来初始化自己。可以选择注释掉#define FORMAT_WITH_FAT_Example。在这种情况下，Windows 会报告该驱动器没有格式化然后给出格式化的选项。

5.5 LibUSB

LibUSB 示例应用程序功能上类似于前面介绍的 HID 应用程序。不同之处在于这个示例包含一个供 Windows 主机 PC 使用的叫做 RSK_LibUSB 的软件。这个开源的库目的是提供一个独立于操作系统接口的平台，可以在多个操作系统上采用通用的代码使用一个 USB 设备。

目标 RSK 代码不依赖于任何一个外部的库然而也要支持 LibUSB 功能。

若要使用提供的主机端软件需先下载 LibUSB-Win32，LibUSB-Win32 是应用于 Windows32 位开发环境的 LibUSB 代码的端口。可以从下面的 LibUSB32 网站下载：

<http://libusb-win32.sourceforge.net/#installation>

按照“Device Driver Installation”（不是“Filter Driver Installation”）的步骤下载一个“libusb-win32-device-bin-x.x.x.x.tar.gz”文件。一旦解压，你会得到想要的文件——不需要运行安装程序。

生成的 LibUSB 示例工程的 host/driver 文件夹下有一个示例 inf 文件。复制这个文件夹，粘贴到已下载的 LibUSB 文件夹“libusb-win32-device-bin-x.x.x.x\bin”。（注意：该操作可以省略“找到新硬件向导”询问驱动文件在什么位置。）

使用 RSK 指南手册里的 LibUSB 例程对 RSK 进行编程，然后运行该代码。用一根 USB 线连接主机 PC 和 RSK。设备第一次连接到某一 USB 端口时，Windows 会检测到新设备并请求适合的驱动。采用跟之前描述过的 CDC 工程类似的处理方法浏览到已下载的 LibUSB 文件夹，这个文件夹中已经拷入了示例 inf 文件（libusb-win32-device-bin-x.x.x.x\bin）。此时应该安装 LibUSB 驱动。

主机端软件：

主机端软件放在工程的 release 目录下，叫做 RSK_LibUSB.exe。另外也可以编译提供的源码。编译提供的源码需要下载的 LibUSB 文件——usb.h，LibUSB.lib 和 LibUSB0.dll。更新 MS Visual C++ 工程设置，按照要求指向这些文件。

运行 RSK_LibUSB.exe，则会显示如下窗口：

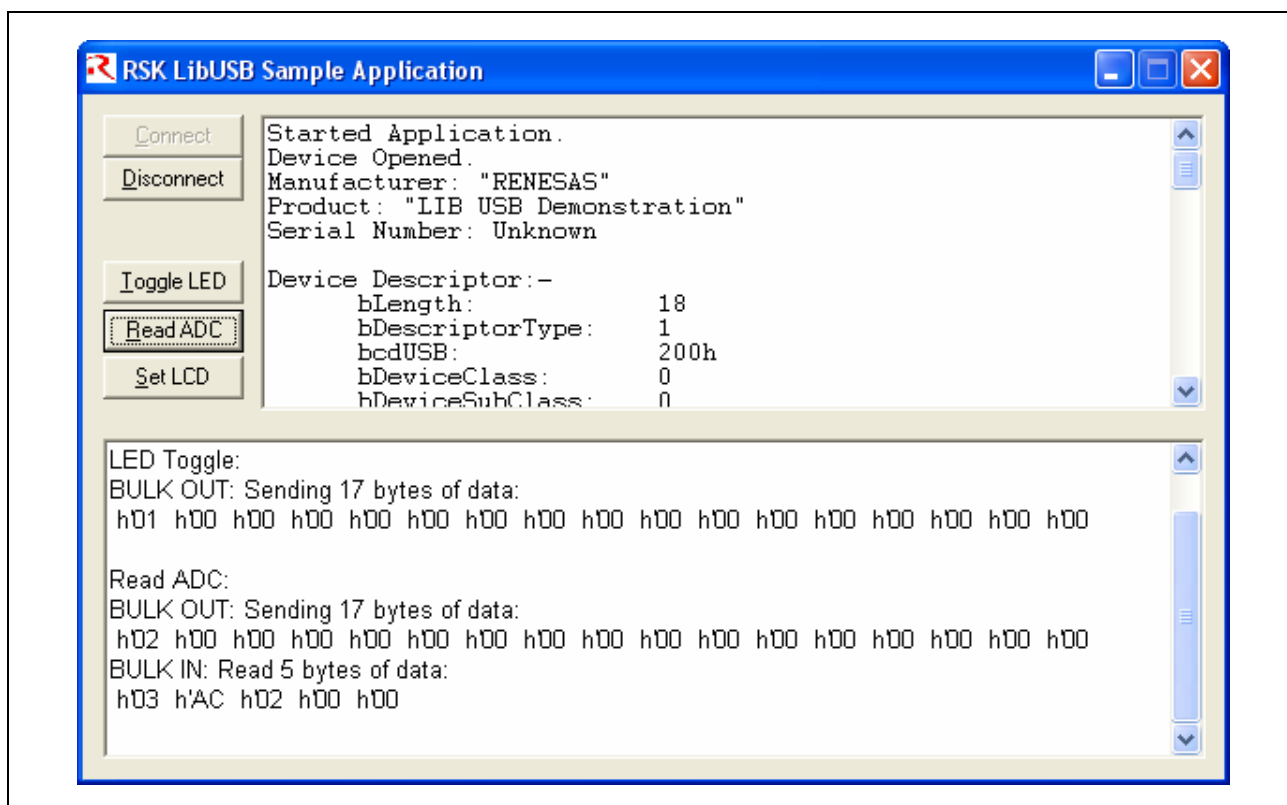


图5 LibUSB 应用界面

注意：这是连接后按下了“Set LCD”按钮和“Read ADC”按钮的一个截屏。

现在可以跟应用程序进行连接了。点击“Connect”按钮，会提示确认所连接设备的 VID 和 PID。如果没有把 RSK 固件的 VID 和 PID 改成自己专门指定的，那么缺省值就是正确的。当成功连接后，设备信息就会显示出来，剩下的按钮被使能。

“Toggle LED”按钮使 RSK 上的 LED 灯闪烁。

“Read ADC”按钮使 RSK 读取 ADC 并把值返回到主机端显示出来。

“Set LCD”按钮可以变换 RSK 的 LCD 上的字符。

为了演示 RSK 产生通信请求，可以按下 RSK 上的按键，这个信息将会返回到主机端并显示在对话框里。

这表明 RSK 跟 PC 之间可以成功的传送数据。所有信息都采用了大小固定的数据格式，一个 OUT 和一个 IN:

IN Message:(RSK 至 PC)

Byte 1

Bit 0 = LED status.

Bit 1 = ADC value valid indicator.

Bit 2 = Switch pressed indicator.

Byte 2-5 = 32 bit, little endian ADC Value.

OUT Message:(PC 至 RSK)

Byte 1

Bit 0 = LED toggle request.

Bit 1 = ADC read request.

Bit 2 = LCD set request.

Byte 2-17 = 16 ASCII Characters for LCD.

只要按下 RSK 上的一个按键，RSK 就会采用中断 IN 传输发送一个 IN 信息。对应于某一 OUT 信息，RSK 采用 BULK IN 传输方式发送 IN 信息进行响应。

每当用户点击对话框上的按钮时，RSK 就采用 BULK OUT 传输发送一个 OUT 信息。

LibUSB 应用程序包含下列文件：

Target:

libusb_app.c

libusb_app.h

usbdescriptors.c

Host:

\\PC\\RSK_LibUSB\\...

6. 附加信息

如何使用 High-performance Embedded Workshop (HEW) 的细节, 请参照 CD 上的 HEW 手册或者安装的 Manual Navigator。

M16C/6C 系列微处理器的相关信息, 请参照《M16C/6C Group Hardware Manual》。

M16C/6C 汇编语言的相关信息, 请参照《M16C Series Programming Manual》。

E8a 仿真器的相关信息, 请参照《E8a 仿真器用户手册》。

更多本产品的相关信息, 请参照瑞萨网站:

http://www.renesas.com/renesas_starter_kits

瑞萨微处理器的概要信息, 请参照如下网站:

全球: <http://www.renesas.com/>

瑞萨 M16C/6C Starter Kit

USB 例程 用户手册

Publication Date: Rev.1.00 May 31, 2010

Published by: Renesas Electronics Corporation.

瑞萨 M16C/6C Starter Kit
USB 例程 用户手册



瑞萨电子株式会社