

MC9S12VR 系列演示实验培训

编写者: Carlos Aceff
墨西哥汽车与工业化解决方案小组

1 简介

本文档介绍如何使用针对飞思卡尔 MC9S12VR 微控制器创建的实验示例套件中包含的代码。该实验示例套件包含的参考代码显示了如何配置和使用以下模块:

- S12FTMRG (用于闪存写入)
- S12PWM8B8CV2 (用于 PWM 生成)
- S12SCIV5 (用于 SCI 通信)
- BATSV2 (用于使用电池传感器)
- S12HSDRV1 (供高边驱动器使用)
- S12LSDRV1 (供低边驱动器使用)
- S12GMMCV1 (用于在不同的闪存页中导航)
- S12SPIV5 (供 SPI 使用)

可从 www.freescale.com 下载该实验示例套件。

目录

1	简介	1
2	设置	2
2.1	工具设置	2
2.2	板设置	2
3	演示实验示例	2
3.1	闪存编程示例	2
3.2	BAT 感应演示	3
3.3	PWM 模块	4
3.4	MMC 程序闪存分页窗口	5
3.5	SCI 通信	6
3.6	SPI 通信	7
3.7	LSDRV 模块	8
3.8	HSDRV 模块	9
4	结论	10
5	实用参考资料	10

2 设置

2.1 工具设置

应该在评估板 **S12VR64EVB** 上运行实验示例；可通过飞思卡尔网站购买此 EVB。

另外，有一个前提条件就是为 **S12VR** 安装评估版（或永久版）的 **CodeWarrior Development Studio**；**S12VR64EVB** 随附了该软件，飞思卡尔文档《**LL18UHVQSG 快速入门指南：S12VR64EVB**》（EVB 也随附了该文档）中包含了其安装指南。

2.2 评估板设置

要正确配置 **S12VR64EVB**，必须执行列表中列示的操作项。

1. 必须使用《快速入门指南：**S12VR64EVB**》中所示的默认跳线设置来配置 **S12VR64EVB** 板。
2. 可通过将 12 V 的电源接入 **J6** 或 **J4** 和 **J5** 中为板供电。
3. 将开关（**SW3**）按至打开位置。
4. 将 **A/B USB** 电缆连接至主机 **PC** 上开启的 **USB** 端口，并将 **USB** 迷你连接器连接至 **S12VR64EVB** 演示板。按照屏幕说明安装所需的 **USB** 驱动器（如果需要）。

3 演示实验示例

3.1 闪存编程示例

闪存模块主要用于在闪存中写入信息，并防止其发生变化。该操作可通过向闪存模块的寄存器区域写入命令来完成。因此，该程序将显示如何写入这些命令，以使用已知模式向闪存扇区写入数据。

文件 **flash.c** 中的过程 **LaunchFlashCommand** 是用于写入闪存命令的例程。查看此过程有助于了解如何操作寄存器。

3.1.1 设置

1. 打开 **CodeWarrior**。
2. 从 **CodeWarrior** 主菜单中选择“文件 -> 打开”并选择 **9S12VR64_FLASH_DEMO.mcp** 文件。
3. 单击“打开”。此时将打开项目窗口。
4. 从主菜单中选择“项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至演示板。
5. 此时将打开新的调试器环境。

请注意，该程序将加载到 **RAM** 中；这样做是必要的，因为当发生批量擦除或者程序闪存更新时，将无法读取程序闪存。

此外，闪存配置字段将不会发生变化。如果发生意外擦除（由于用户修改软件），则该部分将会被保护，并且不允许对其执行进一步的写入操作。有关如何解除保护 uC 的信息，请参考 **P&E Micro** 网站：<http://www.pemicro.com/>

3.1.2 说明

本实验练习的目的是展示如何为大部分可用程序闪存和所有数据闪存编程。

只需按 **F5** 即可执行程序指令；调试器将在插入 **BGND** 指令的编程位置自动停止。**BGND** 指令将激活 **BDM**，导致程序流在从调试器运行时停止。

3.1.2.1 第一个 BGND 位置

调试器将在执行闪存模块的分频器配置之前的那一刻停止（要写入闪存命令，必须先设置闪存分频器寄存器）。按 **F11** 可以逐步执行代码。检查此代码之后，您可再次按 **F5**，在下一个 **BGND** 指令处停止。

3.1.2.2 第二个 BGND 位置

现在，调试器将在写入闪存的函数处停止。不断地执行代码（按 **F11**）以查看此函数的详情。可随时按 **F5** 继续执行程序。

3.1.3 摘要

本实验练习显示如何初始化闪存模块以及如何在寄存器中写入命令，以使用已知数据模式向所有数据闪存和大多数程序闪存写入数据。

3.2 BAT 感应演示

BATS 模块提供测试电池电源引脚 **VSENSE** 电压或芯片电源引脚 **VSUP** 电压的功能。**VSENSE** 引脚或 **VSUP** 引脚的电压可通过内部分频器传输至内部模数转换器。用户可以不受 **ADC** 布线的限制，将其中一个电压传输至比较器，以生成低压或高压中断，向微控制器发出警报。

请阅读 `bats_hal.c` 文件中有关 `BATS_SetAdcConnectionChannel` 函数的信息；其中包含将所需基准电压传输至 **ADC** 所需的代码。

3.2.1 设置

1. 打开 CodeWarrior。
2. 从 CodeWarrior 主菜单中选择“文件 -> 打开”并选择 S12VR64_BATS_DEMO.mcp 文件。
3. 单击“打开”。此时将打开项目窗口。
4. 从主菜单中选择“项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至演示板。
5. 此时将打开新的调试器环境。
6. 在评估板和计算机之间连接串行电缆。
7. 打开串行终端并按照如下所示进行配置：
 - 57600 bps
 - 8 个数据位
 - 1 个停止位
 - 无奇偶校验

3.2.2 说明

在调试器上按 F5 以启动程序。终端将显示数字化的 VSENSE 值。串行终端将显示电源供电的变化，因此您需要更改电源电压以查看数字化 VSENSE 值的变化。

3.2.3 摘要

本实验示例展示如何配置 BATS 模块，以便将内部 VSENSE 连接到模拟转换器。串行终端显示数字化的值。

3.3 PWM 模块

S12 PWM 模块版本 2 是通道可缩放的模块，它已在 S12 PWM8B8C 版本 1 的基础上进行优化。PWM0 与 PWM7 之间通道可以成对伸缩。可用通道编号为 2、4、6 和 8。

本实验示例展示一种设置和使用脉冲宽度调制 (PWM) 模块创建具有不同极性设置的 50% 占空比输出的方法。此行为能够最好地说明双通道示波器上是否可同时显示所有 PWM 信号。

请查看 main.c 文件，以了解有关模块配置的详情。

3.3.1 设置

必须在运行实验示例之前完成这些步骤。

1. 在 Windows “开始”菜单中选择 CodeWarrior 将它启动。

2. 从 CodeWarrior 主菜单中选择“文件 -> 打开”并选择 9S12VR64_PWM_DEMO.mcp 文件。
3. 单击“打开”。此时将打开项目窗口。
4. main.c 文件中包含此演示的 C 代码。双击该文件将其打开。
5. 从主菜单中选择“项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至演示板。
6. 此时将打开新的调试器环境。下载至演示板完成之后，关闭调试器环境。

3.3.2 说明

1. 按 RESET（重置）按钮 SW4。代码必须以这种方式运行和配置 PWM 通道：50% 占空比，端口 PP0 和 PP1 上存在 24.7 kHz 的信号（可在 J11、引脚 2 和 5 处看到）。
2. 尝试同时探测两个信号（如果可能）。这将显示信号的极性差异。

3.3.3 摘要

PWM 是许多微控制器上的通用模块；可用于调光控制等应用（其中的闭合周期决定了应用的亮度）。本实验示例展示如何配置 PWM 模块，以生成极性互补的两种信号。

3.4 MMC 程序闪存分页窗口

此微控制器的内存是分页的，页在地址 0x8000 和 0xC000 之间映射。页寄存器被称为 PPAGE。

本实验示例展示如何通过更改内存页访问所有内存区域。有关如何更新 PPAGE 寄存器的信息，请参见 S12VR64_MMC_DEMO.mcp 中的 main.c 文件。

3.4.1 设置

必须在运行实验示例之前完成这些步骤：

1. 在 Windows “开始”菜单中选择 CodeWarrior 将它启动。
2. 从 CodeWarrior 主菜单中选择“文件 -> 打开”并选择 S12VR64_MMC_DEMO.mcp 文件。
3. 单击“打开”。此时将打开项目窗口。
4. main.c 文件中包含此演示的 C 代码。双击该文件将其打开。
5. 从主菜单中选择“项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至演示板。
6. 此时将打开新的调试器环境。请勿关闭调试器环境。

3.4.2 说明

此程序应该在用户预期看到内存页更新的位置停止。因此，用户需要在开始时按 F5 并在每次程序停止时再次按下。程序停止时，用户能够看到内存窗口并在寄存器区段看到 PPAGE 寄存器。每次更新 PPAGE 寄存器时，用户将可以看到地址 0x8000 处的信息发生变化。用户可以看到以下页：0xD、0xE、0xF、0xC。

程序第一次停止时，寄存器 PPAGE 将显示 0x0D 值。在位置 0x8000 处，用户将看到消息 0D PPAGE。看到该消息之后，用户应再次按 F5 以查看下一页；在这种情况下，PPAGE 将指向 0x0E，内存区域将显示 0E PPAGE。

请重复此过程以检查所有内存区域。

3.4.3 摘要

S12GMMC 能够将多达 64 KB 的闪存、512 字节的 EEPROM 和 2 KB 的 RAM 映射成全局内存映射。可通过页寄存器访问此全局内存的一部分。

本实验示例展示了如何更改 PPAGE 寄存器以访问所有分页内存。

3.5 SCI 通信

SCI 允许与外围设备和其他 CPU 进行异步串行通信。

本实验示例展示如何配置 SCI 模块，以使用以下配置传输和接收数据：

- 57600 bps
- 8 个数据位
- 无奇偶校验
- 1 个停止位

有关硬件配置的信息，请参阅本项目中的 main.c 文件。

3.5.1 设置

必须在运行实验示例之前完成这些步骤。

1. 在 Windows “开始” 菜单中选择 CodeWarrior 将它启动。
2. 从 CodeWarrior 主菜单中选择 “文件 -> 打开” 并选择 9S12VR64_SCI_DEMO.mcp 文件。
3. 单击 “打开”。此时将打开项目窗口。
4. 从主菜单中选择 “项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至演示板。
5. 此时将打开新的调试器环境。

6. 软件使用 RS-232 端口与用户交互。打开终端窗口（波特率 = 57600，数据位数 = 8，奇偶校验 = N，停止位数 = 1，流控制 = 无）以查看 RS-232 端口数据。
7. 使用一根串行电缆将 EVB 中的 DB9 与您的主机连接起来。

3.5.2 说明

按 F5 执行程序；代码开始执行并将 SCI 配置为所选波特率。串行终端将显示 SCI 模块传输的信息。您可根据串行终端中显示的信息将数据发送回 SCI。选择一些选项并观察寄存器的配置。

3.5.3 摘要

在本实验示例中，SCI 模块用于提供有关硬件状态的信息，显示如何将 SCI 配置为输出和输入信息。

3.6 SPI 通信

SPI 模块允许 MCU 和外围设备之间进行双工同步串行通信。本实验示例展示如何将模块配置为主机，以不断地传输数据字节。

有关模块配置的信息，请参阅本项目中的 `main.c` 文件。

3.6.1 设置

此演示中需要一台示波器和三个示波探针。必须在运行实验示例之前完成这些步骤：

1. 在 Windows “开始” 菜单中选择 CodeWarrior 将它启动。
2. 从 CodeWarrior 主菜单中选择 “文件 -> 打开” 并选择 `S12SPIV5.mcp` 文件。
3. 单击 “打开”。此时将打开项目窗口。
4. `main.c` 文件中包含此演示的 C 代码。双击该文件将其打开。
5. 从主菜单中选择 “项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至演示板。
6. 此时将打开新的调试器环境。下载至演示板完成之后，关闭调试器环境。
7. 将示波探针连接至端子 J16 上的信号 `CLK_SPI`、`SPI_MISO` 和 `SPI_SS`。有关此跳线的信息，请查阅 www.freescale.com 上提供的评估板线路图。
8. 将示波器配置为在 `CLK_SPI` 的下降沿上触发。

3.6.2 说明

请遵照以下说明来运行本实验示例。

1. 按 RESET（重置）按钮 SW4。代码开始执行，并将 SPI 配置为以 3.125 Mbit/s 的波特率传输数据字节。
2. 监视示波器上的 SPI 传输信息，以查看从属设备选择、MOSI 上传输的数据以及串行时钟信号之间的关系。

3.6.3 摘要

可以使用 SPI 模块在外围设备和 MCU 之间进行同步串行通信。本实验示例展示如何在主控模式下设置和使用 SPI 模块以传输恒定的信息。

3.7 LSDRV 模块

LSDRV 模块提供两个低边驱动器，这些驱动器通常用于驱动感应负载（继电器）。本实验示例展示如何配置硬件以显示以下功能：低边驱动器启用和过流检测。

可在本项目中的 main.c 文件中找到有关使用这些驱动器的详情。

3.7.1 设置

1. 在 Windows “开始”菜单中选择 CodeWarrior 将它启动。
2. 从 CodeWarrior 主菜单中选择“文件 -> 打开”并选择 S12VR64_LS_DEMO.mcp 文件。
3. 单击“打开”。此时将打开项目窗口。
4. main.c 文件中包含此演示的 C 代码。双击该文件将其打开。
5. 从主菜单中选择“项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至评估板。

3.7.2 说明

在需要用户输入参数的位置，本程序将插入一条 BGND 指令。因此，用户可以通过按 F5 开始执行程序。程序停止后，用户需要更改变量“test”——此变量显示在数据窗口中。此变量的可能值如下：

要激活 LS0	1
要激活 LS1	2
要停用 LS0	3
要停用 LS1	4
要测试 LS0 中的过流功能	7
要测试 LS1 中的过流功能	8

更改变量 “test” 之后，请按 F5 以执行该测试。

要测试过流功能，您必须：

1. 将变量 “test” 更改为 7 或 8。
2. 运行程序。
3. 将 20 Ω 电阻器连接在 J21 和 R12（对于 LS0）之间或 J22 和 R13（对于 LS1）之间。
4. 研究如何设置变量 LS0OC_Flag 或 LS1OC_Flag 以及如何关闭输出。

要退出此测试模式，请按 Ctrl+R 重置微控制器。

3.7.3 摘要

LSDRV 模块提供两个低边驱动器，这些驱动器通常用于驱动感应负载（继电器）。本实验示例展示如何配置低边驱动器，以激活继电器并检测过流情况。请注意，这些驱动器可处理的最大电流为 150 mA。

3.8 HSDRV 模块

HSDRV 模块提供两个高边驱动器，这些驱动器通常用于驱动 LED 或电阻负载（通常为 240 Ω ）。此处未将白炽灯或卤素灯视为可能的负载。

本实验示例展示如何配置硬件以启用以下功能：高边驱动器启用和过流检测。

请阅读文件 main.c 以了解有关模块硬件配置的信息。

3.8.1 设置

1. 在 Windows “开始” 菜单中选择 CodeWarrior 将它启动。
2. 从 CodeWarrior 主菜单中选择 “文件 -> 打开” 并选择 S12VR64_HS_DEMO.mcp 文件。
3. 单击 “打开”。此时将打开项目窗口。
4. main.c 文件中包含此演示的 C 代码。双击该文件将其打开。
5. 从主菜单中选择 “项目 -> 调试”。此时将会编译源代码，生成可执行文件并将其下载至评估板。

3.8.2 说明

在需要用户输入参数的位置，本程序将插入一条 BGND 指令。因此，用户可以通过按 F5 开始执行程序。程序停止后，用户需要更改变量 “test” — 此变量显示在数据窗口中。此变量的可能值如下：

要激活 HS0	1
要激活 HS1	2
要停用 HS0	3
要停用 HS1	4
要测试 HS0 中的过流功能	7
要测试 HS1 中的过流功能	8

更改变量 “test” 之后，请按 F5 以执行该测试。

要测试过流功能，您必须：

1. 将变量 “test” 更改为 7 或 8。
2. 运行程序。
3. 在 D7（对于 HS0）的阳极和阴极或 D8（对于 HS1）的阳极和阴极之间连接 20 Ω 电阻器。
4. 研究如何设置变量 HS0OC_Flag 或 HS1OC_Flag 以及如何关闭输出。

要退出此测试模式，请按 Ctrl+R 重置微控制器。

3.8.3 摘要

MCS12VR64 中的高边驱动器可用于驱动小型负载。本实验示例展示如何对其进行配置以驱动 LED 和继电器；此外，还介绍开环和过流检测功能。

4 结论

MCS12VR 系列微控制器 (MCU) 提供 16 位微控制器的增强功能；此系列随附高边和低边驱动器，用于直接驱动继电器和 LED。此系列是各种中央车体控制应用（如下端车窗升降器或天窗控制器）的理想之选。

应用笔记附带一个 AN4448SW.zip 文件，其中包含实验示例所需的全部 CodeWarrior 项目。可从 www.freescale.com 下载该文件。

5 实用参考资料

可从 www.freescale.com 获取以下资料。

软件开发工具:

适用于 HCS12(X) 微控制器的 CodeWarrior Development Studio 版本 5.9

应用说明:

AN2612 — 使用 HCS12 定时器通道发生 PWM, 修订版 1,1/2006

AN2428 — HCS12 ATD 模块概述, 修订版 0,01/2003

AN2883 — 使用 UART 串行通信接口作为 HCS12 MCU 上的 UART, 修订版 1.0, 05/2005

参考手册:

MC9S12VRRMV2、MC9S12VR 系列参考手册, 修订版 ev. 2.2 版, 2011 年 6 月 20 日

How to Reach Us:**Home Page:**freescale.com**Web Support:**freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和 / 或规格中所提供的 "典型" 参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括 "经典值" 在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：freescale.com/SalesTermsandConditions。

Freescale, the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. MagniV is trademark of Freescale Semiconductor Inc. All other product or service names are the property of their respective owners.

© 2012, 2013 Freescale Semiconductor, Inc.

© 2012, 2013 飞思卡尔半导体有限公司。

Document Number: AN4448
Rev 0, 01/2012