Revolution SDK

Reset and Shutdown Programming Manual

Version 1.02

The content of this document is highly confidential and should be handled accordingly.

RVL-06-0162-001-D
Released: October 17, 2008

Table of Contents

1 Introduction 5

2 Reset 6
2.1 Using the Restart Function 6
2.2 Restart Implemented by the Application 6

3 Shutdown 7

4 Reset and Shutdown Preprocessing 8
4.1 Terminate Audio Playback 8
4.2 Darken the Screen 8
4.3 Wait for Writes to Internal Flash Memory to Complete 8
4.4 Wait for SD Card Processing to Complete 8
4.5 Wait for Nintendo GameCube Memory Card Processing to Complete 9
4.6 Shut Down Wii Remote 9
4.7 Reset Nintendo GameCube Controller Origin and Deactivate Rumble Motor 9
4.8 Perform Disc Check (Only When Application Implements Restart) 9

5 Precautions 11
5.1 Functions Called from Callbacks or Handlers 11
5.2 Optical Disc Drive Functions 11
5.3 Stopping the Scheduler 11
5.4 Converting Segment Addresses 11
5.5 Data in Memory 12
5.6 Location for Calling Reset or Shutdown 12

Revision History

Version	Revision Date	Description
1.02	2007/12/28	Section 4.6: Updated the information about shutdown processing for the Wii Remote accessory.
1.01	2007/07/26	Section 4.8: Updated sentences by replacing the `OSRebootSystem` function with the `OSReturnToMenu` function.
1.00	2006/09/01	Initial version.

1 Introduction

This document provides descriptions and precautions about the reset and shutdown operations of the Wii system.

2 Reset

Reset refers to the process of returning to the game startup screen (restart). Applications must execute a reset when RESET is pressed on the Wii Console or when Reset is selected in the HOME Menu. With the Nintendo GameCube, reset was executed after a player pressed and then released RESET. However, on the Wii console a reset is executed immediately when RESET is pressed.

Reset can be implemented in two different ways.

- Using the restart function
- Restarting using an implementation defined by the application

Note: Reset preprocessing (see Chapter 4 Reset and Shutdown Preprocessing) is required regardless of which implementation technique is used.

2.1 Using the Restart Function

The Wii console OS reloads the application from the Game Disc and restarts operations from the beginning of the application.

The function `OSRestart` has been created for this purpose. For details, see the `OSRestart` page of *Revolution SDK Function Reference*.

2.2 Restart Implemented by the Application

An application can reset itself at any point. The restart point, such as an opening movie sequence or game startup screen, can be freely selected by the programmer.

There is no special function corresponding to `OSRestart`. Instead, the implementation of this functionality is left up to the programmer.

3 Shutdown

Shutdown refers to the process of shutting off power to the Wii console. Applications must execute a shutdown when the Power Button on the Wii console or Wii Remote is pressed.

The function `OSShutdownSystem` is provided for this purpose. For details, see the `OSShutdownSystem` page of *Revolution SDK Function Reference*.

4 Reset and Shutdown Preprocessing

In order to prevent the system from locking up and other possible problems, certain preprocessing is required when executing a reset or shutdown. Information on the preprocessing required for reset and shutdown is presented in this section.

If a restart is implemented by an application, it can be implemented in the same manner as a standard scene transition back to the game startup screen. However, be absolutely sure to perform the processing described in sections 4.8 Perform Disc Check (Only When Application Implements Restart) and 4.7 Reset Nintendo GameCube Controller Origin and Deactivate Rumble Motor.

4.1 Terminate Audio Playback

Before executing a reset or shutdown, we recommend that you first terminate all audio playback by the application for the following two reasons. First, even though the reset and shutdown functions both stop audio hardware internally, it may take a long time for the hardware to stop if audio playback is not terminated before initiating the reset or shutdown. Second, it is possible for some noise to be generated during the reset or shutdown.

4.2 Darken the Screen

Before executing a reset or shutdown, we recommend that the screen be darkened by calling the function `VISetBlack` (and the functions `VIFlush` and `VIWaitForRetrace`). If a reset or shutdown is executed without darkening the screen, the frame buffer may be updated inappropriately during the reset or shutdown process and the on-screen display may be appear strange.

4.3 Wait for Writes to Internal Flash Memory to Complete

If it becomes necessary to execute a reset or shutdown while writing data to internal flash memory, we recommend that you wait for all data to be written first, if possible, before executing the reset or shutdown. The data written cannot be guaranteed if the reset or shutdown is executed while data is still being written to internal flash memory.

4.4 Wait for SD Card Processing to Complete

We recommend that you allow all SD Card processing to complete before executing the reset or shutdown. Because both reset and shutdown functions unmount the SD Card internally within the API, the reset or shutdown operation may take some time if SD Card processing has not yet finished.

4.5 Wait for Nintendo GameCube Memory Card Processing to Complete

We recommend that you allow all Nintendo GameCube Memory Card processing to complete before executing the reset or shutdown. Because both the reset and shutdown functions unmount the Memory Card internally within the API, the reset or shutdown operation may take some time if Memory Card processing has not yet finished.

4.6 Shut Down Wii Remote

With Revolution SDK 2.2 Patch 10 and later versions, shutdown processing for the Wii Remote is handled by the system. When the system is shut down, the connection between the Wii console and the Wii Remotes is terminated. In addition, when the system is reset, the system disconnects the Wii Remotes temporarily, then places them in an automatic reconnect state (in which they reconnect without any button input) for about 15 to 20 seconds after disconnection. This allows the Wii Remote to be restored to its status prior to the reset without any button input from the user when restarting a game application or moving to the Wii Menu.

4.7 Reset Nintendo GameCube Controller Origin and Deactivate Rumble Motor

At the time of the reset operation, applications that support the Nintendo GameCube Controller must reset the Nintendo GameCube Controller origin using the function `PADRecalibrate`. This is because users sometimes press RESET to correct the origin when it becomes misaligned.

The Rumble Motor must be deactivated in order to prevent a Controller that is not being used from falling from a desk or other surface after the reset operation. The `PADRecalibrate` function also deactivates the Rumble Motor.

4.8 Perform Disc Check (Only When Application Implements Restart)

A disc check must be performed before an application executes its own restart operation. A disc check determines if the most recently confirmed correct disc has ever been removed (even once). If the application detects this during reset processing, do not restart the application. Instead, call the `OSReturnToMenu` function to return execution to the Wii Menu.

For example, assume that the user is playing Game A, switches discs to play Game B, then presses RESET. The player naturally expects Game B to restart. However, if Game A is designed to restart itself using a restart implemented in the application when RESET is pressed and it does not perform a disc check, Game A will restart instead. This is not what the player expects. A disc check is therefore essential when executing a restart that is implemented by the application.

Even if the disc check detects that the disc has been removed, the correct disc may have been reinserted. However, because it takes several seconds to determine that the correct disc has been reinserted once it has been removed, it is essential in such cases to return execution to the Wii Menu using the `OSReturnToMenu` function without attempting to identify the disc. If disc identification is performed and it is found that the loaded disc is not the correct one, a system restart will be applied at that point and it will take several more seconds to start the disc.

In addition, even if a disc is removed during game play, the device driver will automatically check whether the correct disc is inserted the next time the disc is accessed. (Naturally, this takes several seconds.) If RESET is pressed right after the correct disc was confirmed by this automatic check, the disc check returns “already recognized” and no system reboot is required. A disc check is used to check whether the disc has been removed (even once) since the last time it was detected.

Use the `DVDCheckDiskAsync` function when performing disc checks.

When restarting using `OSRestart`, a disc check is performed within the `OSRestart` function. If the wrong disc is detected as a result, the system is automatically rebooted. There is therefore no reason to perform a disc check before executing `OSRestart`.

Note: We are not recommending that the system always be rebooted whenever a disc has been removed. Instead, we are recommending that the system be rebooted whenever RESET is pressed and the disc has yet to be recognized.

5 Precautions

5.1 Functions Called from Callbacks or Handlers

Problems may occur if there are any active user callbacks or user handlers that might call one of the following functions after the reset or shutdown function has been called:

- A function that initializes an audio-related library (such as `AXInit` or `AIInit`)
- A `GX` function

Both the reset function and shutdown function first shuts down all subsystems, then disables all interrupts and cancels all user alarms. If there are any active alarms or asynchronous functions that have not completed when the reset or shutdown function is called, there is a chance that a related user callback or handler may be called after the subsystem in question has been shut down but before interrupts and alarms have been disabled. Depending on the subsystem, a fatal error may occur if a callback or handler tries to call the subsystem after shutdown and fails.

The functions just listed might present this type of problem. If the system locks up after the reset or shutdown function is called, be sure to check if there is a callback or handler that is calling one of the functions just listed.

5.2 Optical Disc Drive Functions

We recommend that you not call any optical disc drive functions before calling the reset or shutdown functions. `DVDCancel` poses a particular problem because the reset and shutdown functions cancel all optical disc drive requests. Not only is calling `DVDCancel` unnecessary, doing so may cause the optical disc drive to be reset twice depending on the exact timing of the reset.

5.3 Stopping the Scheduler

The reset and shutdown functions temporarily stop the thread scheduler within the function. User threads are not executed once one of these functions has been called.

5.4 Converting Segment Addresses

The Revolution SDK does not use the Broadway's MMU for segment address conversion and will not access related registers, including SDR1 or SR*n*. If an application implements its own mechanism for segment address conversion, the registers in question must be restored to their original state before executing restart using the `OSRestart` function. If these registers are not restored, applications may behave unexpectedly.

5.5 Data in Memory

There is no guarantee that all data in memory will match after an application is restarted.

5.6 Location for Calling Reset or Shutdown

The reset or shutdown function must not be called from within a callback.