TWL-SDK

TWL-SDK Application Development Guide

Support for TWL

2009/01/07

TWL-06-0014-001-D
Released: August 26, 2010

Table of Contents

Code

Tables

Figures

Revision History

Revision Date	Description
2009/03/12	Added text and a Note about using SD cards to section 7.3.1 Reading and Writing Files.
2009/01/07	Changed the values specified with `WramMapping`.
2008/12/15	Added explanation of `libsyscall.a` in Chapter 8 Other.
2008/12/05	• Added explanation about use of SD Cards for the purpose of debugging. • Added description about the `ferret` component. • Revised the description of NAND application clone booting.
2008/12/04	Added cautions about use of inline functions.
2008/11/06	Revised reference to the *TWL-SDK Function Reference Manual* in section 6.1 Integrity Check.
2008/10/14	• Revised an explanation of the relationship between the CAMERA library and the CODEC mode. • Added a reference to the *TWL-SDK Function Reference Manual* in section 6.1 Integrity Check.
2008/09/16	Initial version.

1 Introduction

This document covers new items related to the development of applications for DS/TWL using the TWL-SDK. These items have been added because the SDK now supports TWL.

2 Application Types

2.1 ROM Types

In the NITRO-SDK, there was basically only one ROM type for Nintendo DS applications. Applications using this existing type of ROM are called "Nintendo DS software" in the TWL-SDK. In the TWL-SDK there are two additional types.

- TWL-compatible software
- TWL-exclusive software

There are different types of ROM for each of these three application types: NITRO, HYBRID, and LIMITED.

2.1.1 NITRO ROM

The NITRO ROM type is for developing Nintendo DS software. Nintendo DS software can run on either a DS or a TWL. However, although it runs on a TWL, Nintendo DS software cannot use the additional TWL devices or memory regions.

To create a NITRO ROM, link only the libraries that do not contain "`.TWL`." For components, use either the NITRO version of `mongoose` (`$TwlSDK/components/mongoose/ARM7-TS[.thumb]`) or `ichneumon`.

To create a NITRO ROM using the TWL-SDK `make` build system, specify `NITRO` as the `TWLSDK_PLATFORM` environment variable.

2.1.2 HYBRID ROM

The HYBRID ROM type is for developing TWL-compatible software. TWL-compatible software can run on either a DS or a TWL. If it is running on a TWL, it can use the additional TWL devices and the additional memory regions.

For details on developing HYBRID ROMs, see Chapter 3 HYBRID ROM.

2.1.3 LIMITED ROM

The LIMITED ROM type is for developing TWL-exclusive software. TWL-exclusive software only runs on TWL, not on Nintendo DS, and can use TWL's additional devices and memory regions.

To create a LIMITED ROM, link only the libraries that end with "`.TWL.LTD.a`". For components, use `racoon` or `ferret`, specifying `MAP2_TS_LTD` for the RSF `WramMapping` property. For details on `racoon` and `ferret`, see *AboutComponents.pdf*.

To create a LIMITED ROM with the TWL-SDK `make` build system, set the `TWLSDK_PLATFORM` environment variable to `TWL`, and specify `LIMITED` in `TWL_ARCHGEN` in the Makefile.

2.2 Media

With Nintendo DS, applications are stored only on DS Cards. However, with TWL it is possible to store applications in system memory.

2.2.1 Card Applications

Card applications are written to DS Cards or TWL Cards and run from them. In the NITRO-SDK, all applications are card applications.

To create a HYBRID or LIMITED ROM card application, set the `Media` property of the RSF provided in `makerom.TWL` to `GameCard`. The `Media` property is "`GameCard`" by default, so the application will be a card application even if you leave this unspecified.

Using NITRO ROMs you can only create card applications. Note that the presence of a `Media` property in the RSF when you create a NITRO ROM will cause an error.

2.2.2 NAND Applications

NAND applications are written to TWL system memory and run from there. Because NAND applications only run on TWL, basically they are created as LIMITED ROMs. Although it is possible to create a NAND application as a HYBRID ROM, this is intended as clone boot support. NAND application clone boot is supported beginning from TWL-SDK 5.1, PR Version. Please contact Nintendo if you want to create a HYBRID ROM for a NAND application.

To create a NAND application, specify "`NAND`" for the `Media` property in the RSF provided to `makerom.TWL`.

For details on developing NAND applications, see *NandAppManual.pdf*.

2.2.3 Feature Restrictions

Some features are restricted depending on whether the application is a Card application or a NAND application.

Features not available on Card applications:

- Sub-banners
- System internal fonts
- Access to NAND application save data
- Access to the photograph database

Features not available on NAND applications:

- Access to Card ROM or Card backup memory

3 HYBRID ROM

3.1 Overview

HYBRID ROM is one of the ROM types added in the TWL-SDK.

HYBRID ROMs can run on Nintendo DS, in which case they only use Nintendo DS features. HYBRID ROMs can also run on TWL, in which case they can use the devices and memory space that have been added to TWL.

The HYBRID ROM type is provided to make it possible to develop applications for Nintendo DS that when run on TWL can use its added features. For example, when one of these DS applications is running on TWL, it could use photos taken with the camera as insignia in DS Multi-Card Play.

3.2 Creating a HYBRID ROM

To create a HYBRID ROM, link only the libraries that end with "`.TWL.HYB.a`." For components, use the HYBRID version of `mongoose` (`$TwlSDK/components/mongoose/ARM7-TS.HYB[.thumb]/`), specifying `MAP2_TS_HYB` for in the RSF's `WramMapping` property.

To create a HYBRID ROM with the TWL-SDK `make` build system, set the `TWLSDK_PLATFORM` environment variable to "`TWL`," and specify `HYBRID` for `TWL_ARCHGEN` in the Makefile.

3.3 Operation

On Nintendo DS, applications are constructed from an ARM9 executable binary, an ARM7 executable binary, and overlays. In the Nintendo DS IPL, the two binaries are loaded into memory for use by the two respective processors, after which processing begins.

HYBRID ROMs include these two binaries as well as ARM9 TWL-exclusive and ARM7 TWL-exclusive executable binaries, for a total of four binaries, along with overlays and/or relocatable modules. When a HYBRID ROM runs on a Nintendo DS, the Nintendo DS IPL loads only the ARM9 executable binary and the ARM7 executable binary into memory before beginning processing. When a HYBRID ROM runs on TWL, the TWL IPL loads all four binaries, then begins processing. Using this mechanism, the application can run on both Nintendo DS and TWL; furthermore, when running on TWL it can use TWL's added features.

3.4 Development

There are two main items of note when developing HYBRID ROMs.

- Nintendo DS main memory size
- TWL mode-exclusive APIs

3.4.1 Nintendo DS Main Memory Size

When a HYBRID ROM runs on TWL, it can use a 16MB main memory space. However, when it runs on a Nintendo DS it can only use a 4MB main memory space. For this reason, when a hybrid application is running on a Nintendo DS, developers must do their best not to load code that is only required for TWL.

A simple solution is to store this code in the ARM9 TWL-exclusive executable binary. By default, the application code is stored in the ARM9 executable. To store application code in the ARM9 TWL-exclusive executable, use the pair of headers `twl/ltdmain_begin.h` and `twl/ltdmain_end.h`. When you include these two headers in the source code, all code and data defined in the region between the two headers will be stored in the ARM9 TWL-exclusive executable binary.

Code 3-1 Using twl/ltdmain_begin.h and twl/ltdmain_end.h

```
// Code contained in the ARM9 executable
#include <twl/ltdmain_begin.h>
// Code contained in the ARM9 TWL-exclusive executable
#include <twl/ltdmain_end.h>
// Code contained in the ARM9 executable
```

However, when using this solution, exercise caution when using inline functions. When an inline function that is referenced from both the ARM9 executable binary and the ARM9 TWL-exclusive executable binary is not made inline, there is no guarantee that the code will be stored in the ARM9 executable binary. If it is not stored in the ARM9 executable, it cannot be referenced when running in NITRO-compatible mode. In this case, either make the code smaller until it is made inline, or use a normal function instead.

Another solution is the use of overlays or relocatable modules. This solution provides a great deal of flexibility. With this method, code needed only on TWL is collected in overlays or relocatable modules, and is only loaded when the application is run on TWL.

3.4.2 Using TWL Mode-Exclusive APIs

None of the TWL mode-exclusive APIs added in the TWL-SDK run on Nintendo DS. Therefore, your implementation must ensure that when the HYBRID ROM runs on Nintendo DS, the application does not use these functions.

That said, the function instances of functions from the TWL mode-exclusive APIs are defined so that they are included in the ARM9 TWL-exclusive executable. They will not be loaded into main memory on Nintendo DS. In addition, these APIs are wrapped with inline functions that determine whether the API is running on a TWL, so you can call them when running on a Nintendo DS with no problem. If you use functions from these APIs on a Nintendo DS, they will either return an error code indicating the call was invalid or failed, or they will simply do nothing.

Code 3-2 Example of TWL Mode-Exclusive API Implementation

```
SDK_INLINE CAMERAResult CAMERA_Init(void)
{
    if (OS_IsRunOnTwl() == TRUE)
    {
        CAMERA_InitCore();
        return CAMERA_I2CInitCore(CAMERA_SELECT_BOTH);
    }
    return CAMERA_RESULT_FATAL_ERROR;
}
```

3.5 Debug

To run a HYBRID ROM in NITRO-compatible mode, run it on a Nintendo DS. To run it in TWL mode, run it on a development TWL console.

If you use IS-TWL-DEBUGGER you can switch between NITRO-compatible mode and TWL mode on one machine.

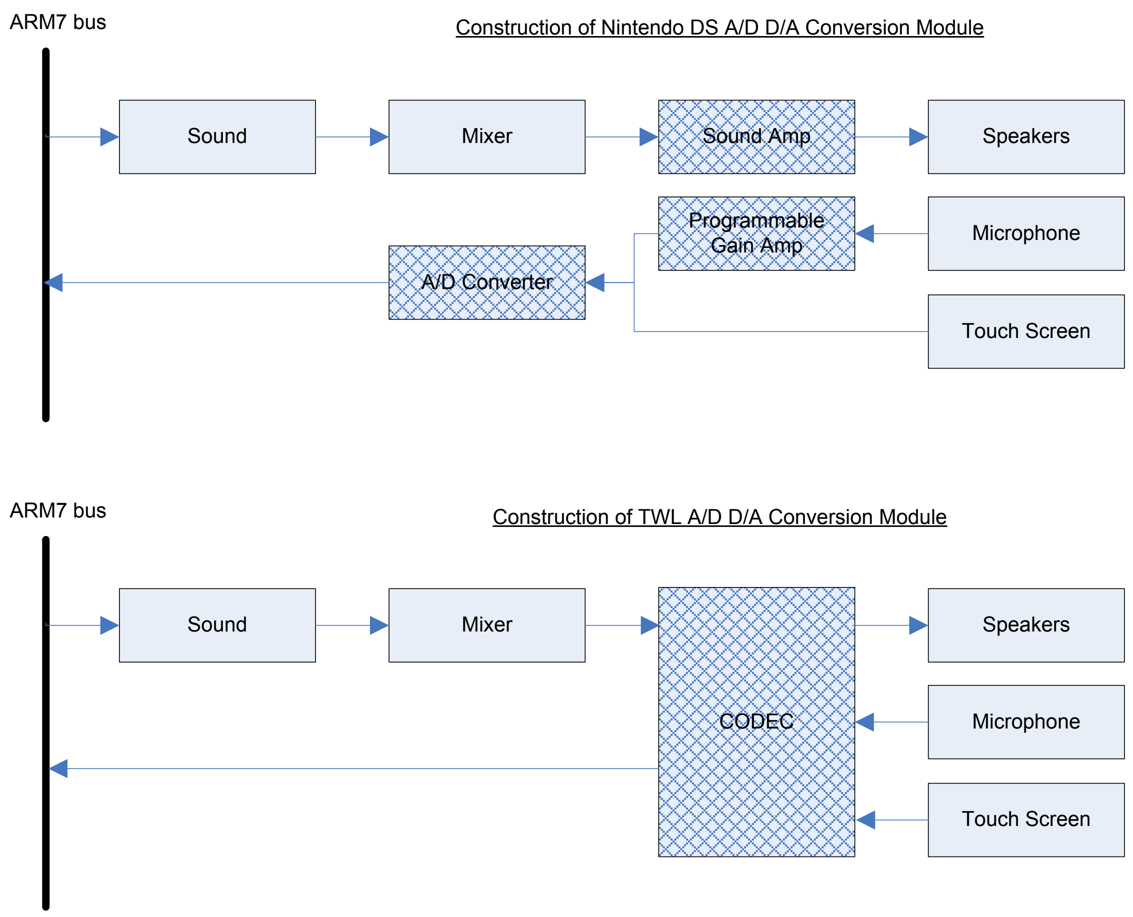
4 CODEC Mode

4.1 CODEC

On TWL, the hardware module that performs D/A and A/D conversion has been replaced with a module called CODEC.

The following diagram shows the pertinent sections extracted from the Nintendo DS and TWL system block diagrams. Irrelevant items are not shown. As you can see from the diagram, the CODEC processes sound output as well as microphone and touch-screen input.

Figure 4-1 The A/D D/A Conversion Modules on Nintendo DS and TWL



4.2 CODEC Modes

There are two CODEC modes: CODEC-DS and CODEC-TWL.

The CODEC-DS mode is a mode provided to assure maximum compatibility with Nintendo DS. In CODEC-DS mode, the CODEC emulates the operation of the various individual modules that were provided for DS and yields the same behavior.

The CODEC-TWL mode exhibits the full capabilities of the CODEC. CODEC-TWL mode can use a number of extended features not available on the Nintendo DS.

The CODEC mode is specified in the RSF's `CodecMode` property. The CODEC mode is changed based on this value before an application starts. You cannot change the CODEC mode while an application is running.

In the case of NITRO ROMs, the CODEC mode is set to CODEC-DS mode and no other options are available. In the TWL-SDK build system, the default is CODEC-TWL mode.

4.3 CODEC-TWL Mode

CODEC-TWL mode provides the following additional features.

- Sound output and microphone sampling at 47.61KHz
- Microphone sampling at 16-bit precision
- 120-level microphone amplifier
- IIR filters for microphone input and sound output
- Forced audio output
- Automatic CODEC-driven microphone and touch screen sampling

CODEC-TWL mode also has the following additional restrictions.

- The CODEC processing frequency limits the usable microphone sampling frequencies to four types.
- You cannot use 8-bit or 12-bit precision microphone sampling.

In CODEC-TWL mode you can select either 32.73 KHz or 47.61 KHz as the CODEC processing frequency. The CODEC processing frequency simultaneously affects both microphone sampling and sound output. Therefore, you cannot sample the microphone at 32.73 KHz and at the same time output sound at 47.61 KHz. However, it is possible to continue using these features while switching the processing frequency.

Forced sound output makes it possible to output audio at any volume from the TWL system speaker, regardless of the user-specified volume and whether headphones are inserted in the headphone jack. This is used to play the camera shutter sound when taking a photo.

With automatic CODEC-driven microphone and touch-screen sampling, the automatic sampling that ARM7 formerly performed using software is now performed by the CODEC hardware. In particular, this reduces the load on ARM7 during microphone sampling, which reduces the dropped processing that occurs when ARM7 is used to simultaneously process microphone and wireless or other features.

4.4 Mode Determination

Be careful when selecting a CODEC mode. Although you can freely select a mode for each application, you cannot switch the CODEC mode while an application is running. The following elements will affect this decision.

- Will you use the camera?

  The CAMERA library can only be used if the CODEC mode is CODEC-TWL. You must therefore use CODEC-TWL mode whenever you use the camera.

- Is the application a HYBRID ROM?

  HYBRID ROMs must be able to run on both Nintendo DS and TWL. If you do not need the additional features provided in CODEC-TWL mode, use CODEC-DS mode because it equalizes the behavior of Nintendo DS and TWL hardware, and makes development easier.

5 Restrictions on Starting Applications

5.1 Region

With Nintendo DS, with the exception of applications made for China, all applications could run on every Nintendo DS. In contrast, HYBRID ROMs and LIMITED ROMs running on TWL use the concept of *regions*.

TWL has four regions: Japan, America, Europe, and Australia. Application regions are specified by the RSF's `CardRegion` property for both card applications and NAND applications. Set the region that is appropriate to the application.

5.2 Parental Controls

HYBRID ROMs and LIMITED ROMs running on TWL are subject to Parental Controls based on the application's rating.

Each application must attach its own ratings information, which is set by ratings organizations in each region. Use *MasterEditor* to attach ratings information.

6 ROM Authentication

6.1 Integrity Check

The contents of HYBRID ROMs and LIMITED ROMs running on TWL are checked for completeness during ROM read access.

During this check, the read data is internally hashed and compared with a hash table that was verified at application startup. For card applications, these reads occur during ROM accesses by the CARD library and `rom` archive accesses by the FS library. For NAND applications, these reads occur during `rom` archive accesses by the FS library. See **FS → Overview → ROM Archives** in the *TWL-SDK Function Reference Manual* for more information.

It is important to note that, in particular, access speeds for a HYBRID ROM are slower when run on a TWL system when compared to a Nintendo DS.

If the integrity check fails, application execution will stop on `OS_TPanic`.

6.2 Development Systems

A TWL development system is required to run TWL applications that are developed using the TWL-SDK.

You use a commercial Nintendo DS to develop for the NITRO-compatible mode of NITRO ROMs and HYBRID ROMs. In contrast, you cannot run in-development SRL files of any ROM type on commercial TWL systems. A development TWL system is required to develop HYBRID ROMs and LIMITED ROMs.

The reverse is true of production HYBRID ROMs and LIMITED ROMs, which will not run on a development TWL system. The tables below show which types of applications will run on each type of console.

Table 6-1 Development Applications and the Debugger and Systems They Run On

	NITRO ROM	HYBRID ROM		LIMITED ROM
		NITRO-Compatible Mode	**TWL Mode**	
Nintendo DS system	O	O	X	X
IS-NITRO-EMULATOR	O [*1]	O	X	X
IS-NITRO-CAPTURE	O	O	X	X
IS-TWL-DEBUGGER	O [*2]	O [*2]	O [*2]	O [*2]
IS-NITRO-CAPTURE	O	X	O	O
Development TWL system	O	X	O	O
Commercial TWL system	X	X	X	X

Table 6-2 Production Applications and the Debugger and Systems They Run On

	NITRO ROM	HYBRID ROM		LIMITED ROM
		NITRO-Compatible Mode	**TWL Mode**	
Nintendo DS Console	O	O	X	X
IS-NITRO-EMULATOR	X [*1]	X	X	X
IS-NITRO-CAPTURE	O	X	X	X
IS-TWL-DEBUGGER	O [*2]	X	X	X
IS-TWL-CAPTURE	O	X	X	X
Development TWL System	O	X	X	X
Commercial TWL System	O	X	O	O

* 1 It is not possible to run DS/TWL cards using IS-NITRO-EMULATOR.

* 2 It is not possible to run DS/TWL cards that use special devices using IS-TWL-DEBUGGER.

7 Using SD Cards for Debugging Purposes

7.1 Overview

Access to SD cards in the final product is prohibited by the guidelines. However, use of SD cards for debugging purposes during development is permitted.

7.2 Use Conditions

SD cards can be used for debugging purposes for the HYBRID ROM or LIMITED ROM types and only for the DEBUG or RELEASE builds. They cannot be used for NITRO ROMs or for FINALROM builds. Be aware that only FINALROM builds of ROMs can be submitted.

Table 7-1 Conditions for Use of SD Cards for Debugging Purposes

	NITRO ROM	HYBRID ROM	LIMITED ROM
DEBUG Build	×	○	○
RELEASE Build	×	○	○
FINALROM Build	×	×	×

7.3 Usage

7.3.1 Reading and Writing Files

SD cards are mounted as `sdmc` archives in the FS library. For this reason, by using the FS library with `sdmc` as the archive name, files can be read or written on the SD Card. (See the note below when using SD cards with HYBRID build card-boot applications.) For example, the `log.txt` file directly under the root directory on the SD card can be indicated with the "`sdmc:/log.txt`" path.

Note: In addition to the `FS_Init` function, you must also run the `FS_InitFatDriver` function when initializing the FS library if you are using SD cards with HYBRID build card-boot applications.

7.3.2 Inserting and Removing SD Cards

Unlike the system save memory, SD cards can be inserted or removed at any time. The `FS_RegisterEventHook` function is used to notify when an SD card has been inserted or removed. By registering a callback with this function, the callback can be called when an SD card is inserted or removed.

Note: When using a mini SD card with an SD card adapter, insertion/removal of the mini SD card is not detected if the SD card adapter is left inserted in the TWL console.

8 Other

8.1 libsyscall.a and libsyscall.twl.a

You must obtain `libsyscall.a` and `rom_header.template.sbin` from Nintendo. For HYBRID ROMs and LIMITED ROMs, `libsyscall.twl.a` and `rom_header.LTD.sbin` are also necessary; for these, use the files provided in the SDK.