Nintendo DSi Applications

Justin Braach
Senior Software Engineer

Software Development Support Group

TWL Applications Overview

- Introduction to the TWL SDK
  - TWL SDK Features
  - TWL SDK Build System

- TWL Supported media types
  - Game Card
  - NAND

TWL Applications Overview

- TWL Application types
  - NITRO
  - Limited
  - Hybrid
- NAND Applications
- TWL Development suggestions

Introduction to the TWL SDK

- TWL is an extension of NITRO
  - Not a completely new hardware platform
  - New features for NITRO developers to leverage
- TWL SDK is the "new" NITRO SDK
  - Really just NITRO SDK with TWL support
  - Can be used for both NITRO and TWL development
  - No further NITRO SDK updates planned

TWL SDK Features

- Supports both NITRO and TWL hardware
  - Extends existing libraries for TWL support
  - Adds libraries for new TWL-only features
- Supports two media formats
  - Card and NAND
  - Retains support for DS Download Play
- Supports three different application types
  - NITRO (Nintendo DS)
  - Hybrid (Nintendo DSi Compatible)
  - Limited (Nintendo DSi Only)

TWL SDK Build System

- Updated from NITRO SDK Build System
  - Based on GNU Make 3.81 (works with 3.80)
  - See detailed explanations in SDKRules.html
- Flags used for selecting application type
  - TWLSDK_PLATFORM
  - TWL_ARCHGEN
- Flags for building NAND applications
  - TWL_NANDAPP
  - TARGET_BIN

TWL SDK Media : Game Card

- Card features
  - Can support all three application types
  - Card sizes up to 4Gbit
  - Built-in backup memory (up to 8Mbit)
- Card limitations
  - No support for Mask ROM for TWL applications
    - NITRO-only applications still support MROM
  - Card applications cannot access NAND

TWL SDK Media : NAND

- NAND features
  - Supports Limited and Hybrid application types
  - NAND applications can access shared data
  - Save game archives can be shared

- NAND limitations
  - NITRO application type is unsupported
  - Limited size (16MB maximum)
  - Some different/additional guidelines from Card

TWL Media Access

Application Type	DS Game Card		SD Memory Card	System NAND Memory
	ROM Region	Backup Memory		
Card application	Allowed	Allowed	Prohibited	Prohibited
DS Download Play	Access prohibited	*1	Not Possible	Not Possible
NAND applications	Access prohibited	*2	Prohibited in principle	Possible

*1: Read/write allowed if product of the same company. However, a DS Game Card must be inserted when a DS Download Play application starts.

*2: Read/write allowed if product of the same company. However, a DS Game Card must be inserted when a NAND application starts.

TWL SDK Application Types

	Supported Media Types		TWL Extension Support	DS Download Play
	Card	NAND		
NITRO	Supported	Not Possible	Not Supported	Supported
Hybrid	Supported	Supported	*1, *2	Supported
Limited	Supported	Supported	Supported	Not Supported

*1: Supported for Card and NAND applications.
*2: Not supported if transferred via DS Download Play

NITRO Applications

- Normal Nintendo DS Applications
  - Targets only NITRO hardware
  - From the code standpoint, almost identical to NITRO SDK
- TWL SDK Build System Settings
  - TWLSDK_PLATFORM=NITRO
  - TWL_ARCHGEN is not referenced

NITRO Application Cautions

- Application size differences
  - Same code built with TWL SDK is slightly larger than if built with NITRO SDK
  - Size difference depends on libraries used
    - See SDK documentation for details

- Some API changes from NITRO SDK
  - Very minor changes

Limited Applications

- "Nintendo DSi exclusive"
  - Targets TWL hardware only
  - Will not run on NITRO hardware
- TWL SDK Build System Settings
  - TWLSDK_PLATFORM=TWL
  - TWL_ARCHGEN=LIMITED

Limited Application Features

- Access to most TWL features
  - Some access based on media type
  - NAND is only accessible by NAND applications
  - SD Card usage is prohibited by guidelines
- Full-time use of available TWL features
  - Can code specifically for TWL hardware
  - No need to code to NITRO hardware bugs/limitations

Hybrid Applications

- "Nintendo DSi Compatible"
  - Runs as TWL application on TWL hardware
  - Runs as NITRO application on NITRO hardware
- TWL Build System Settings
  - TWLSDK_PLATFORM=TWL
  - TWL_ARCHGEN=HYBRID

Hybrid Application Features

- Compared to Limited applications
  - Same access to TWL extensions in TWL mode as Limited applications
  - Hybrid applications can support Clone boot
  - Limited applications are not subject to NITRO hardware restrictions

Hybrid Application Features

- Compared to NITRO applications
  - In NITRO mode almost identical to NITRO apps
  - TWL Hardware bug fixes and behavior are optional and can be disabled by program
  - This is done via System Configuration library (SCFG) calls

Hybrid Application Features (cont.)

- Two run modes
  - NITRO mode: 4MB RAM, NITRO features only
  - TWL mode: 16MB RAM, TWL extensions
  - Run mode is selected by hardware type at boot
- Executable is split into sections
  - NITRO region (old security)
  - Extended region (new security)

Hybrid Application Features (cont.)

- TWL-only code/data in Extended region
  - NITRO hardware will only load NITRO region
  - Extended region is "invisible" to NITRO hardware
- Code/data location is developer controlled
  - Similar mechanism to [I/D]TCM placement
  - #include <ltdmain_[begin|end].h>

Hybrid App Cautions

- Size of Hybrid apps vs. NITRO apps
  - App built as Hybrid is larger than NITRO
  - Difference depends on number of libraries used
    - See SDK documentation for details

- Most complex application type of the three
  - May require extra planning
  - Possibly longer development time than NITRO-only or Limited-only applications

Creating Hybrid Applications: Approach #1

- Nintendo DS-only application
  - Build a great NITRO game then add TWL features
    - Extra/Extended levels
    - Better sound assets, output
    - Mini-games
  - Lowest risk, Lowest overhead
    - Already familiar with NITRO applications
    - TWL-specific coding is "extra" and not integral
  - May not make best use of TWL features
    - TWL features may feel "tacked on"

Creating Hybrid Applications: Approach #2

- Approach as two applications
  - One driver function, two code paths
    - Optimized NITRO version
    - Optimized TWL version
  - Higher risk, higher overhead
    - Potentially long development time
    - Could run into ROM size issues in extreme cases
  - Optimize TWL path without NITRO limitations
    - However 2x the code can mean 2x the bugs…

Creating Hybrid Applications: Approach #3

- Approach as a mixed NITRO/TWL app
  - Treat like cross-platform development
    - Shared generic code base (game logic)
    - Optimized "platform specific" code
  - Moderate risk, moderate overhead
    - Requires good up-front planning
    - Need to be careful of NITRO limitations in shared code
  - Good integration of TWL features
    - Extend NITRO functionality without separate code
    - Can plan for TWL integration during design phase

NAND Applications

- "DSiWare"
  - Small sized applications
  - Purchased from Nintendo DSi Shop
  - Stored to system NAND memory
  - Can be backed up to SD Card
- For TWL Hardware only
  - However, can include DS Download Play
    - Clone boot requires Hybrid executable

NAND Applications (cont.)

- TWL SDK Build System settings
  - TARGET_BIN=<application name>.tad
  - TWL_NANDAPP=TRUE
- No card or backup memory
  - Runs from NAND, saves to NAND
  - Requires special RSF settings
- Not very different from building Card app
  - Must be linked against NA library
  - SRL converted to TAD file with MakeTad tool

NAND-only Features

- ROM and Save Data archives
  - All file access is done via FS library
  - This includes save data R/W access
  - 2 save archives available (public and private)

- Save data can be shared with other apps
  - Only allowed if both apps share company code
  - Access is read/write

NAND App-only Features (cont.)

- Access to shared data
  - Contains Nintendo DSi System Font
    - S, M, and L sizes
  - Does not count against TAD size (use it!)
- Possible to access stored photo data
  - Requires a special library
  - Gives read/write access with restrictions
- Sub-banners
  - Displayed banner can be changed dynamically from application

NAND Application Cautions

- Size restricted by business agreement
  - Maximum allowed TAD size is 16MB
  - Total Size includes:
    - Executable
    - Game Data
    - e-manual
    - Save Data archives

- DSiWare e-manual
  - Binary data and library built into application
  - Size depends on data, number of pages
  - Estimate roughly 100KB for first language
    - Estimate 20~100KB for each additional language

NAND Application Cautions

- NAND access is frequency restricted
  - Guideline restriction
  - Don't access small ranges of NAND repeatedly
    - Many times per second
  - Intended to limit wear on system NAND

- SD Card backup/restore
  - Private archive is not copied during backup
  - During restore, private archive overwritten with empty data
  - Game must function with empty private archive

Taking advantage of TWL extensions

- File cache in Extended memory
  - Pre-fetch data from card to cache
  - Program NDMA to transfer when bus is idle
- Use improved VRAM transfer speed
  - More/larger VRAM cell animations
  - Dynamic textures

Taking advantage of TWL extensions (cont.)

- Improved audio
  - Larger sound data in extended memory
  - Better mixing/effects using DSP features (SNDEX)
- WRAM banks available to program
  - Larger scratch area than DTCM, 1 cycle access
  - Fast WRAM -> VRAM transfers for animation and/or dynamic data

What fun things can you do with the TWL?

- Use camera to customize game assets
  - Modify/replace textures with camera images
  - Use camera data to generate random data
  - Modify in-game mood using average luminance
- Use camera as an input device
  - Face recognition library
- Animated Nintendo DSi Menu Banners

Questions?

- Contact Support@noa.com

- Thanks for listening