

Quick Start Guide

NITRO-SDK

2005/03/11
任天堂株式会社

0. はじめに

本ドキュメントでは NITRO-SDK のインストールと簡単なアプリケーションの作成方法を説明します。

以下のような順に説明します。

1. 開発ツールの用意
2. NITRO-SDK bulb パッケージの展開
3. 環境変数の設定
4. NITRO-SDK ツリーのビルド
5. サンプルを動かしてみる
6. 簡単なプログラムを書いてみる
7. ビルドスイッチ

1. 開発ツールの用意

現在 NITRO-SDK では以下の Windows 環境でビルドできることを確認しています。

- Microsoft Windows 2000 Professional

NitorSDK のビルド(コンパイルなど)を行なうためには下記のツールが必要です。

- CodeWarrior for NITRO
- Cygwin または MinGW(MSYSツール)

またデバッグを行なうためには下記のいずれかのツールが必要です。

- ensata NITRO Software Emulator
- IS-NITRO-EMULATOR

ensata NITRO Software Emulator についてはNITRO-SDKに収録されておりますが、その他の各ツールについては配布元にお問い合わせください。

Cygwin のインストールについては、SDK 展開後の NitroSDK/docs/SDKTools ディレクトリ内の CygwinPackageList.pdf をご覧ください。

インストールについての実作業については各ツールのドキュメントをご覧ください

2. NITRO-SDK パッケージの展開

NITRO-SDK のパッケージをディスクに展開します。zip 形式で圧縮されていますので各種解凍ツールを使われるか cygwin などに付属している unzip コマンドで展開してください。

unzip コマンドを使った場合の展開方法は以下のとおりです。

```
% cd <展開するディレクトリ>
% unzip e:/NITRO-SDK/NitroSDK-2.0-xxxxxxx.zip
```

NitroSDK という名のディレクトリが作成されます。(% はプロンプトです)

3. 環境変数の設定

環境変数 NITROSDK_ROOT に展開されたディレクトリ NITRO-SDK の絶対パスを設定します。何も指定されていない場合は C:\NitroSDK が設定されます。以後このディレクトリのことを \$NitroSDK と表記します。

4. NITRO-SDK ツリーのビルド


Cygwin または MSYS でシェル(bash など)を立ち上げ、\$NitroSDK に移動します。
\$NitroSDK で

```
% make
```

と入力することでビルドが開始されます。もし途中でエラーを出力して停止するようでしたら、ここまでの設定に間違いがある、あるいは本パッケージにバグがあることが考えられます。お手数ですが、ここまでの設定を見直された後で配布元へご連絡ください。

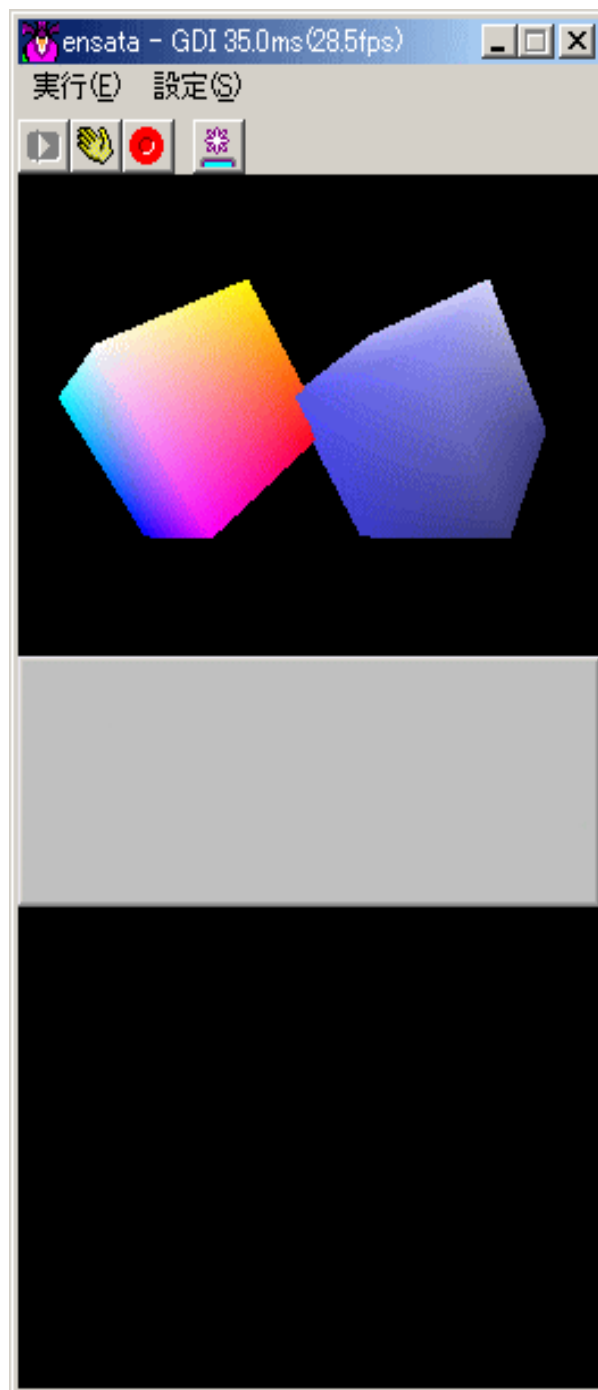
5. サンプルを動かしてみる

ビルドが正常に行なわれたかを確認するためにサンプルを動かしてみます。
ここでは ensata がお手元にあると仮定します。

- 1) ensata を起動します。
- 2) [実行(E)] → [NITRO ファイルの読み込み(O)] を選びます。
- 3) ダイアログで bin ファイルを指定します。
ここでは先ほどのビルドで作成された下記のファイルを指定します。
\$NitroSDK/build/demos/gx/UnitTours/3D_Pol_LightColor/bin/ARM9-TEG/Release/main.srl
- 4) ensata のツールバーの実行ボタン[下図  印] を押します。



5) スクリーンウィンドウに以下のような2つの立方体が表示されればOKです。ちなみに右側の立方体の色は次々に切り替わります。



- 6) 停止ボタン[下図 ○印]を押すと bin ファイルのエミュレーションによる実行を停止します。



6. 簡単なプログラムを書いてみる

簡単なプログラムを書いてみましょう。

- 1) 適当な作業ディレクトリを作成し、以下のファイルをコピーします。

```
$NitroSDK/build/demos/template/src/main.c
$NitroSDK/build/demos/template/makefile
```

- 2) main.c を以下のように編集します。

16行目以降

```
#include <nitro.h>

void NitroMain(void)
{
    OS_Init( );
    OS_Printf( "Hello World of NITROid.¥n" );    ← この行を追加
    while (1){ }
}
```

- 3) make コマンドを実行します。以下のような出力（SDK の修正などによって表示が若干変更されている場合があります）が出ます。もし error や warning などの表示があったときは 2) での編集作業に誤りがないかを見直してください。

```
% make
==== /c/NitroSDK/xxx
C:/Program¥ Files/Metrowerks/CodeWarrior¥ for¥ NITRO¥ V0.3/ARM_Tools/Command_Line
_Tools/mwccarm.exe -lang c -proc arm946e -nothumb -nopic -nopid -interworking -O
4 -opt speed -inline on,noauto -g -msgstyle std -w all -enc SJIS -char unsigned
-stdinc -enum int -stdkeywords off -avoid_strb all,err -DSDK_CWBUG_INLINE -DSDK
_TEG -DSDK_4M -DSDK_ARM9 -DSDK_CW -DSDK_RELEASE -DSDK_CODE_ARM -gccinc -I./inclu
de -I./src -Id:/dev/NitroSDK/include -cwd proj -c main.c -o obj/ARM9-TEG/Release/
main.o
C:/Program¥ Files/Metrowerks/CodeWarrior¥ for¥ NITRO¥ V0.3/ARM_Tools/Command_Line
_Tools/mwldarm.exe -proc arm946e -nothumb -nopic -nopid -interworking -g -msgsty
le std -w on -stdlib -map closure -main _start -L./lib/ARM9-TEG/Release -Ld:/dev
/NitroSDK/lib/ARM9-TEG/Release ./obj/ARM9-TEG/Release/main.o -llibfx.a -llibgx.a
-llibos.a -llibsyscall.a -lcr0.o d:/dev/NitroSDK/include/nitro/ARM9-TEG.lcf -
```

```
o bin/ARM9-TEG/Release/main.nef
C:/Program Files/Metrowerks/CodeWarrior for NITRO V0.3/ARM_Tools/Command_Line
_Tools/elftobin.exe bin/ARM9-TEG/Release/main.nef d:/dev/NitroSDK/tools/elftobin/
spIdle.elf d:/dev/NitroSDK/tools/elftobin/romHeader.bin -o bin/ARM9-TEG/Release/
main.bin
%
```

4) Ensata を起動し、前章と同様にウィンドウ上で右クリックを行ない[実行(E)] → [NITRO ファイルの読み込み(O)] から、作業ディレクトリの下の bin/ARM9-TEG/Release/main.srl を指定します。

5) [設定(S)] → [デバッグ出力(C)] を選びます。

6) 実行ボタンを押すとデバッグ出力ウィンドウに以下のような表示がでます。

Hello World of NITROid.

7) 停止ボタンを押すと bin ファイルのエミュレーションによる実行を停止します。

7. ビルドスイッチ

SDK にはいろいろなビルドの設定があります。

デフォルトでは IS-NITRO-EMULATOR (または TS ボード)用の Release 版ライブラリがリンクされますが、ビルド時のマクロの設定によって Debug 版のビルドを行なうことができます。これをビルドスイッチと呼びます。

ビルドスイッチについての詳しい情報は \$NitroSDK/docs/SDKRules/Rules-Defines.html をご覧ください。