

NS9360 ITRON SDK Make Guide



1.	はじめに.....	3
2.	SDKの概要.....	3
2.1	ITRON SDKの位置づけ.....	3
3.	インストール.....	3
4.	開発環境.....	4
4.1	フォルダ構成.....	4
4.2	ビルド.....	5
4.2.1	サンプルプログラムの内容.....	5
4.2.2	ビルドの実行.....	6
4.3	デバッガ起動.....	8
4.3.1	RAVEN.....	8
5.	タスクの追加.....	10

1. はじめに

本書は、Net Silicon NS9360 Net+Worksの環境でITRONの実現を行うSDKの開発環境および開発手順について記します。

2. SDKの概要

2.1 ITRON SDKの位置づけ

ITRON SDKは、NS9360評価ボードまたは、Connectcore9C評価ボード上で動作するITRONによる開発環境を提供します。

Boot Loaderは、NET+OSで作成するrom.binを使用し、SDKで作成するアプリケーションは、NET+Worksで作成するimage.binと同じ位置づけとなります。

3. インストール

インストールファイルを任意のフォルダへ解凍してください。

推奨フォルダは、

- /netos63_gnu/src/ (NET+OSがVer6.3の場合)

です。

4.2 ビルド

サンプルのタスクを使ってビルド手順について以下に説明します。

4.2.1 サンプルプログラムの内容

サンプルプログラムは以下の機能を実現します。

- **CSP (コンソールサービスプログラム)**
UART 経由で接続した PC からコマンドを入力し、コマンドの実行を行う。
ネットワークパラメータの設定やメモリの R/W などが可能です。
評価ボード (NS9360, connectcore9c) のシリアルポート A と PC を RS 232 C ケーブルで接続してください。

- **HTTP サーバ**
ysock の HTTP サーバが動作します。

- **FTP サーバ**
ysock の FTP サーバが動作します。
HTTP サーバで使用する、HTML ファイルをアップロードできます。

ユーザー名 : "user"
パスワード : "pass"

- **Filesystem**
yfile の Filesystem が動作します。

- **TELNET**
ysock の TELNET が動作します。
TELNET 経由で CSP の実行が出来ます。

4.2.2 ビルドの実行

- ① NET+WorksのBuild Environmentショートカットをダブルクリックし、CYGWINを起動します。



← クリック！

```
-----  
¥¥ //      |||||  |||  |||  ||  |||||  
¥¥//      |||  ||  ||  ||  ||  ||  
// ¥¥      |||  ||  ||  ||  ||  ||  
// ¥¥      ||  |||  |||  ||||  |||||  
-----  
GNU X-Tools v3.4 on CYGWIN_NT-5.1 1.5.13(0.118/4/2) |  
-----  
(c) Microcross, Inc.  
(c) Free Software Foundation  
(c) Cygnus Support  
(-: Hosts of Contributors  
-----  
Help on GNU Bash Shell and xtools commands... |  
-----  
Enter 'xtools <target-alias>' to set up for a cross target.  
      (i.e., xtools arm-elf)  
Enter 'xtools help' to get help on xtools commands.  
Enter 'help' to get Bash Shell help.  
Enter 'Ctrl-d' to exit.  
-----  
GNU X-Tools and Bash Shell ready... |  
-----  
Setting up Shell for arm-elf  
arm-elf GNU X-Tools Shell Ready...  
  
user /cygdrive/c/netos63_gnu  
$
```

- ② カレントディレクトリを "./src/itron/<CONFIG>"へ移動します。

```
user /cygdrive/c/netos63_gnu  
$ cd src/itron/9360EVA  
  
user /cygdrive/c/netos63_gnu/src/itron/9360EVA  
$
```


4.3 デバッガ起動

サンプルのタスクを使ってデバッガ起動手順について以下に説明します。

4.3.1 RAVEN

RAVENを使用した場合の手順を以下に記します。

- ① カレントディレクトリを “. /src/itron/jsp/obj”へ移動します。

```
user /cygdrive/c/netos63_gnu/src/itron/9360EVA
$ cd ../jsp/obj

user /cygdrive/c/netos63_gnu/src/itron/jsp/obj
$
```

- ② ターゲット、ICEの電源を入れます。

- ③ Ocd Remote for ARM926EJ-Sを起動します。



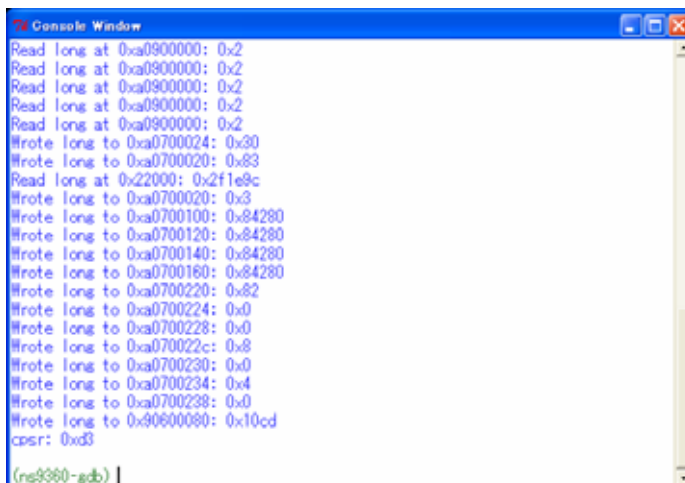
← クリック！

Ocd Remote のウィンドウ

```
ocdremote 2.08: RAVEN via LPT 1 at speed : 1
JTAG SDO <-| CPU(1) ARM926EJ-S : listening on port 8888 |<- JTAG SDI
```

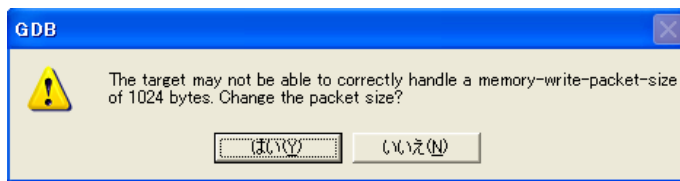
- ④ デバッガを起動します。

```
user /cygdrive/c/netos63_gnu/src/itron/jsp/obj
$ gdbtk -se jsp.elf
```

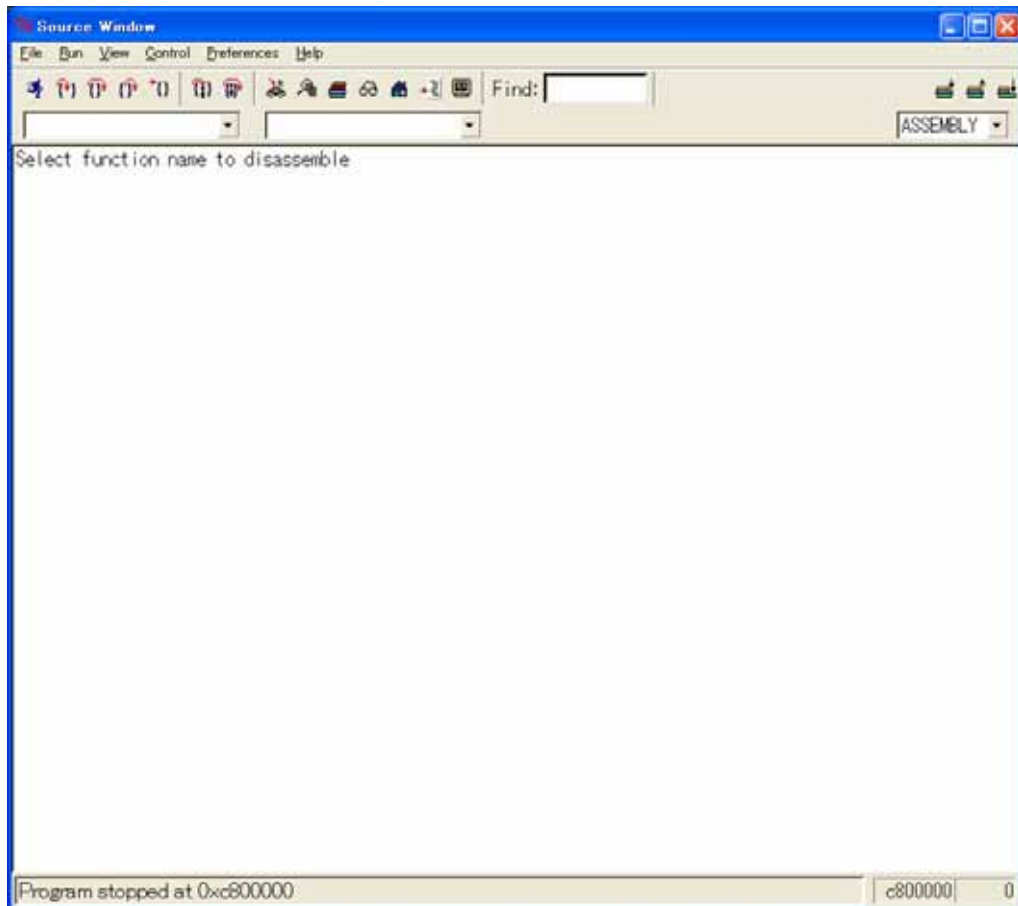


← コンソールウィンドウが表示されます。

- ⑤ 以下の警告が出た場合は、「はい」をクリックしてください。



- ⑥ プログラム (jsp.elf) のロードが終了すると、デバッガのメインウィンドウが表示されます。



- ⑦ 以降の操作は、NET+Worksと同じです。

5. タスクの追加

タスクを追加する場合は、以下の作業が必要となります。

(1) nsl_itron.cfg(itron/nsl_itron)の変更

作成したタスクや使用するリソースの登録を行います

Toppers/JSPでは、タスクの登録は、静的APIの”CRE_TSK”を使用します。

CRE_TSKの宣言は、コンフィグレーション・ファイルに

```
CRE_TSK(ID tskid, {ATR tskatr, VP_INT exinf, FP task, PRI itskpri,  
                SIZE stksz, VP stk} );
```

と記述します。

```
CRE_TSK(APL1, { TA_HLNG, (VP_INT) 1, apl1, MID_PRIORITY, STACK_SIZE, NULL });  
CRE_TSK(APL2, { TA_HLNG, (VP_INT) 2, apl2, MID_PRIORITY, STACK_SIZE, NULL });  
:  
CRE_TSK(TASK3, { TA_HLNG, (VP_INT) 3, task, MID_PRIORITY, STACK_SIZE, NULL });  
CRE_TSK(MAIN_TASK, { TA_HLNG|TA_ACT, 0, main_task, MAIN_PRIORITY, STACK_SIZE, NULL });
```

(2) make.sh(itron/obj)の変更

追加したファイルをmake.sh内の”APP_OBJ”変数に追加します。

```
APP_OBJ="mnt.o   ¥  
        apl1    ¥  
        apl2    "
```

(3) makefile(itron/nsl_itron)の変更

追加したファイルをmakefile内の”UTASK_COBJS”変数に追加します。

```
UTASK_COBJS = $(UNAME).o CPU_IO.o PS_TELNET.o HTTPD_CGI.o CSP.o PS_NIC.o ¥  
              mib_if.o mib_com.o mib_Priv.o mib_Sys.o mib_table.o ¥  
              PS_USER.o halt.o PS_SNMP_USER.o EF_USER.o apl1 apl2
```

(4) ファイルの追加

追加するファイルを”itron/apps”の下に追加します。