

# 目次

<b>アプリケーションの自動起動</b> .....	3
<b>設定</b> .....	3
config の変更 .....	3
romfs image の作成 .....	4
ファームウェアのビルド .....	5
<b>ファームウェアの書き込みと実行</b> .....	5
<b>応用</b> .....	6



# アプリケーションの自動起動

ビルトインアプリケーションの追加で追加した、タスクなどを起動するビルトインアプリケーションを、機器電源 ON 時に自動起動させる方法を紹介します。

公式ドキュメントの該当箇所は [こちら](#) です。

例として、『[AD Converter の利用](#)』で紹介した `adc` というビルトインアプリケーションを自動起動させてみます。

## 設定

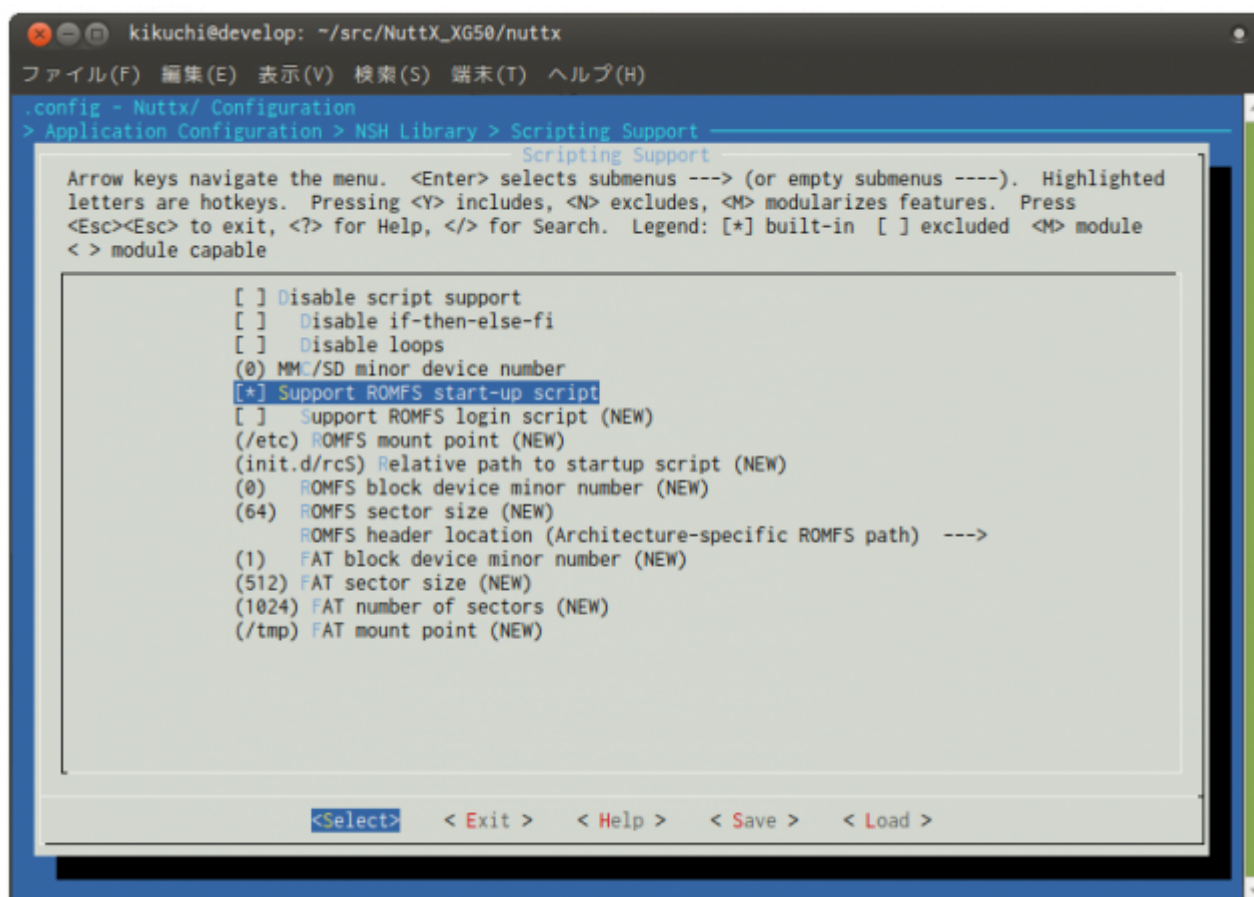
### config の変更

[ドキュメント](#) にある下記条件

- CONFIG\_DISABLE\_MOUNTPOINT がセットされていないこと
- CONFIG\_NFILE\_DESCRIPTOR が 4 より大きいこと
- CONFIG\_FS\_ROMFS が有効化されていること

はデフォルトでクリアするよう設定してあります。

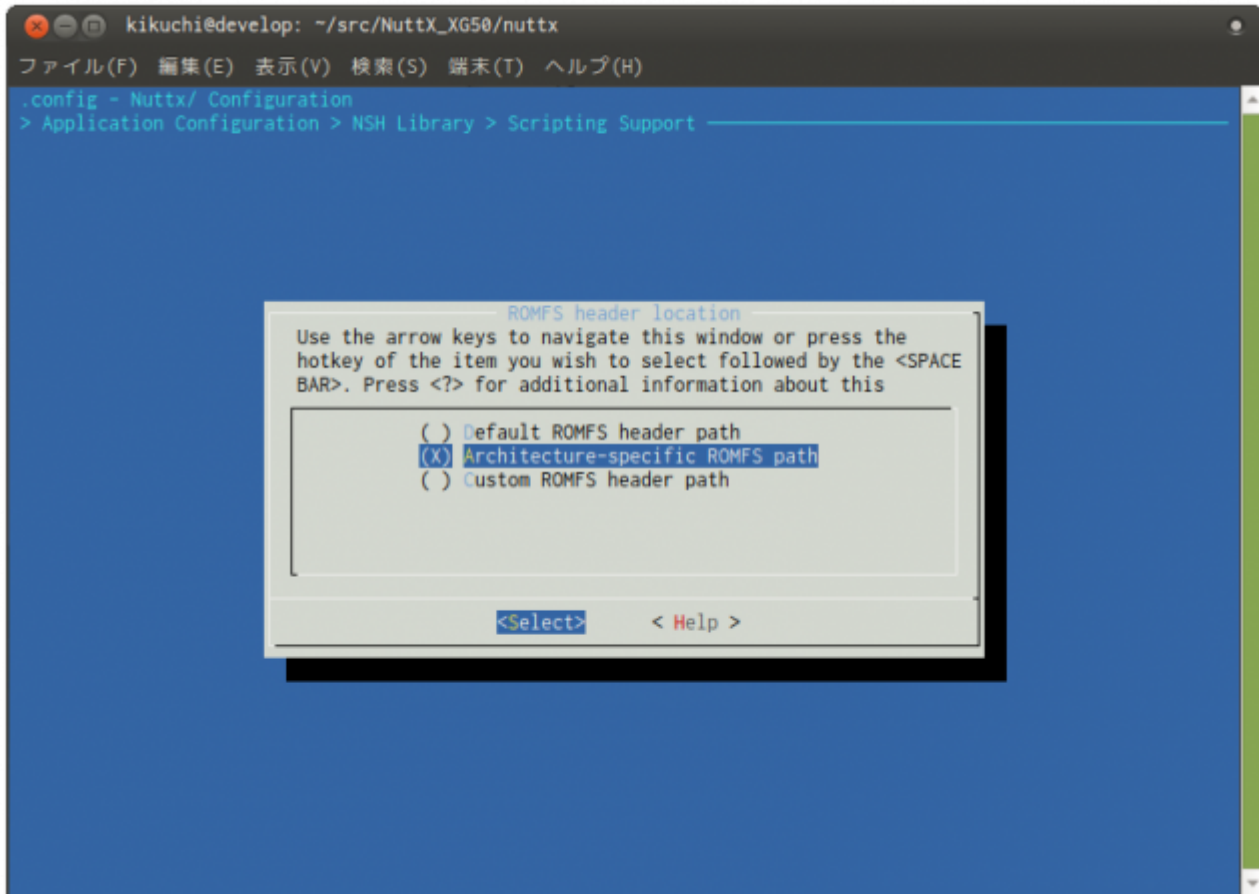
追加で、**CONFIG\_NSH\_ROMFSETC** と **CONFIG\_NSH\_ARCHROMFS** を有効にします。



```
kikuchi@develop: ~/src/NuttX_XG50/nuttx
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
.config - Nuttx/ Configuration
> Application Configuration > NSH Library > Scripting Support
Scripting Support
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted
letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module
< > module capable

[ ] Disable script support
[ ] Disable if-then-else-fi
[ ] Disable loops
(0) MMC/SD minor device number
[*] Support ROMFS start-up script
[ ] Support ROMFS login script (NEW)
(/etc) ROMFS mount point (NEW)
(init.d/rcS) Relative path to startup script (NEW)
(0) ROMFS block device minor number (NEW)
(64) ROMFS sector size (NEW)
ROMFS header location (Architecture-specific ROMFS path) --->
(1) FAT block device minor number (NEW)
(512) FAT sector size (NEW)
(1024) FAT number of sectors (NEW)
(/tmp) FAT mount point (NEW)

<Select> < Exit > < Help > < Save > < Load >
```



## romfs image の作成

ここでビルドを試みると、下記のエラーになります。

```
./nsh_romfsetc.c:57:39: fatal error: arch/board/nsh_romfsimg.h: No such file
or directory
# include <arch/board/nsh_romfsimg.h>
^
compilation terminated.
ERROR: arm-none-eabi-gcc failed: 1
```

nuttX/include/arch/board/nsh\_romfsimg.h を作成する必要があるとわかります。

romfs image は rcS.template というテンプレートファイルから生成します。  
今回は **adc** というアプリケーションを起動させたいので、下記のようなファイルを作成します。

### rcS.template

```
#!/bin/nsh

echo "auto execute adc from rcS!"
```

```
adc
```

nuttx/tools/mkromfsimg.sh というスクリプトで、romfs image を作成します。

```
develop:~/src/NuttX_XG50/nuttx/include/arch/board$
../../tools/mkromfsimg.sh ../../../../
develop:~/src/NuttX_XG50/nuttx/include/arch/board$ ls -ln
total 48
-rw-rw-r-- 1 1000 1000 9551  3月13 13:45 board.h
-rw-rw-r-- 1 1000 1000 2755  3月13 13:49 boardctl.h
-rw-rw-r-- 1 1000 1000 19504 3月12 11:10 centurysys-xg50.h
-rw-rw-r-- 1 1000 1000 6383  3月14 15:38 nsh_romfsimg.h <--- 生成された
-rw-rw-r-- 1 1000 1000  51  3月14 15:38 rcS.template
develop:~/src/NuttX_XG50/nuttx/include/arch/board$
```

## ファームウェアのビルド

ビルドを行います。場合にもよりますが、まずは make clean したほうが良いかもしれません。

## ファームウェアの書き込みと実行

ファームウェアを書き込み、実行してみます。

```
ABCDF
auto execute adc from rcS!
adc_main: g_adcstate.count: 1
adc_main: Hardware initialized. Opening the ADC device: /dev/adc0
Sample:
1: channel: 0 value: 1498
2: channel: 1 value: 3493
3: channel: 2 value: 0
4: channel: 17 value: 926

NuttShell (NSH)
nsh>
```

rcS により自動実行されました。

rcS も以下のように見えています。

```
nsh> ls -lR /etc
/etc:
```

```
dr-xr-xr-x    0  .
dr-xr-xr-x    0  init.d/
/etc/init.d:
-r-xr-xr-x    51  rcS
nsh> cat /etc/init.d/rcS
#!/bin/nsh

echo "auto execute adc from rcS!"
adc

nsh>
```

## 応用

NuttShell には、`if` `then` `else` `fi` や `while` `do` `done` など Linux の shell script のような制御文も使えますので、設定をチェックして起動するプログラムを変えるなどもスクリプトで柔軟に行うことができます。

参照: [NuttShell Conditional Command Execution](#)

From:

<https://wiki.centurysys.jp/> - MA-X/MA-S/MA-E/IP-K Developers' Wiki

Permanent link:

[https://wiki.centurysys.jp/doku.php?id=xg\\_series\\_devel:autostart\\_rcs:start](https://wiki.centurysys.jp/doku.php?id=xg_series_devel:autostart_rcs:start)

Last update: **2020/11/30 09:58**

