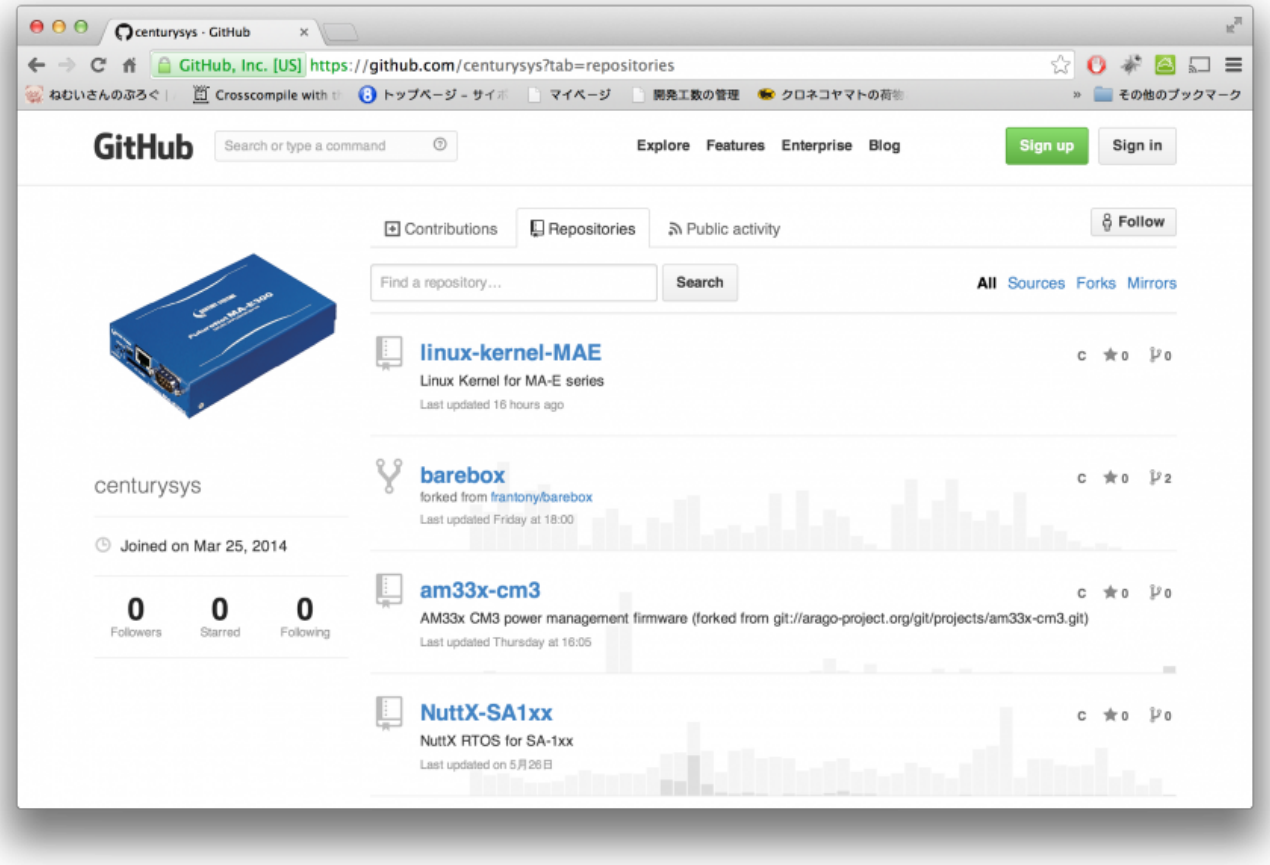
目次

カーネルのビルド

MA-E3xxシリーズ用のカーネルは、GitHubで公開しています[1)]



必要なソフトウェアパッケージの導入

git[2)3)]で管理されていますので、ソースコードを入手するために git パッケージを導入します。

```
user1@lubuntu-vpc:~$ sudo apt-get install git
[sudo] password for user1:
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
以下の特別パッケージがインストールされます:
  git-man liberror-perl
提案パッケージ:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitk
gitweb git-arch git-bzr git-cvs
  git-mediawiki git-svn
以下のパッケージが新たにインストールされます:
  git git-man liberror-perl
アップグレード: 0 個、新規インストール: 3 個、削除: 0 個、保留: 0 個。
```

```
2,979 kB のアーカイブを取得する必要があります。
この操作後に追加で 21.5 MB のディスク容量が消費されます。
続行しますか? [Y/n] y
取得:1 http://jp.archive.ubuntu.com/ubuntu/ trusty/main liberror-perl all
0.17-1.1 [21.1 kB]
取得:2 http://jp.archive.ubuntu.com/ubuntu/ trusty/main git-man all
1:1.9.0-1 [692 kB]
取得:3 http://jp.archive.ubuntu.com/ubuntu/ trusty/main git amd64 1:1.9.0-1
[2,265 kB]
2,979 kB を 2秒 で取得しました (1,466 kB/s)
以前に未選択のパッケージ liberror-perl を選択しています。
(データベースを読み込んでいます … 現在 140805 個のファイルとディレクトリがインストールされています。)
Preparing to unpack .../liberror-perl_0.17-1.1_all.deb ...
Unpacking liberror-perl (0.17-1.1) ...
以前に未選択のパッケージ git-man を選択しています。
Preparing to unpack .../git-man_1%3a1.9.0-1_all.deb ...
Unpacking git-man (1:1.9.0-1) ...
以前に未選択のパッケージ git を選択しています。
Preparing to unpack .../git_1%3a1.9.0-1_amd64.deb ...
Unpacking git (1:1.9.0-1) ...
Processing triggers for man-db (2.6.6-1) ...
liberror-perl (0.17-1.1) を設定しています …
git-man (1:1.9.0-1) を設定しています …
git (1:1.9.0-1) を設定しています …
```

U-boot形式のイメージを作成するために□u-boot-tools を導入します。

```
user1@lubuntu-vpc:~$ sudo apt-get install u-boot-tools
[sudo] password for user1:
パッケージリストを読み込んでいます… 完了
依存関係ツリーを作成しています
状態情報を読み取っています… 完了
以下のパッケージが新たにインストールされます:
  u-boot-tools
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 0 個。
64.1 kB のアーカイブを取得する必要があります。
この操作後に追加で 208 kB のディスク容量が消費されます。
取得:1 http://jp.archive.ubuntu.com/ubuntu/ trusty/main u-boot-tools amd64
2013.10-3 [64.1 kB]
64.1 kB を 0秒 で取得しました (208 kB/s)
以前に未選択のパッケージ u-boot-tools を選択しています。
(データベースを読み込んでいます … 現在 146130 個のファイルとディレクトリがインストールされています。)
Preparing to unpack .../u-boot-tools_2013.10-3_amd64.deb ...
Unpacking u-boot-tools (2013.10-3) ...
Processing triggers for man-db (2.6.6-1) ...
u-boot-tools (2013.10-3) を設定しています …
user1@lubuntu-vpc:~$
```

ファームウェアファイルを作るときに必要となりますのでsquashfs-tools を導入します。

```
user1@lubuntu-vpc:~$ sudo apt-get install squashfs-tools
パッケージリストを読み込んでいます… 完了
依存関係ツリーを作成しています
状態情報を読み取っています… 完了
以下のパッケージが新たにインストールされます:
  squashfs-tools
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 0 個。
90.4 kB のアーカイブを取得する必要があります。
この操作後に追加で 275 kB のディスク容量が消費されます。
取得:1 http://jp.archive.ubuntu.com/ubuntu/ trusty/main squashfs-tools amd64
1:4.2+20130409-2 [90.4 kB]
90.4 kB を 0秒 で取得しました (290 kB/s)
以前に未選択のパッケージ squashfs-tools を選択しています。
(データベースを読み込んでいます … 現在 146153 個のファイルとディレクトリがインストールされています。)
Preparing to unpack .../squashfs-tools_1%3a4.2+20130409-2_amd64.deb ...
Unpacking squashfs-tools (1:4.2+20130409-2) ...
Processing triggers for man-db (2.6.6-1) ...
squashfs-tools (1:4.2+20130409-2) を設定しています …
user1@lubuntu-vpc:~$
```

作業用ディレクトリの作成

以降の作業用に、ディレクトリを作成します。
ここでは“src” というディレクトリを作成しています。

```
user1@lubuntu-vpc:~$ mkdir src
user1@lubuntu-vpc:~$ cd src/
user1@lubuntu-vpc:~/src$
```

リポジトリのclone

“git clone” により、カーネルソースをローカルにcloneします。

```
user1@lubuntu-vpc:~/src$ git clone
https://github.com/centurysys/linux-kernel-MAE.git linux-kernel
Cloning into 'linux-kernel'...
remote: Counting objects: 3502603, done.
remote: Compressing objects: 100% (550862/550862), done.
remote: Total 3502603 (delta 2944877), reused 3479150 (delta 2921481)
Receiving objects: 100% (3502603/3502603), 730.53 MiB | 8.10 MiB/s, done.
Resolving deltas: 100% (2944877/2944877), done.
Checking connectivity... done.
```

```
Checking out files: 100% (46038/46038), done.
```

ソースコードが下記の通り取得できました。

```
user1@lubuntu-vpc:~/src$ ls -l
合計 4
drwxrwxr-x 24 user1 user1 4096  3月 17 16:06 linux-kernel
```

git clone が失敗する場合

Linux Kernelのリポジトリは巨大なため“git clone” がメモリ不足やタイムアウトなどの原因で失敗することがあります。
その場合“-\-depth” を指定して少しずつ取得することでエラーの回避ができます。

```
user1@lubuntu-vpc:~/src$ git clone --depth 3
https://github.com/centurysys/linux-kernel-MAE.git linux-kernel
Cloning into 'linux-kernel'...
remote: Counting objects: 48961, done.
remote: Compressing objects: 100% (46439/46439), done.
remote: Total 48961 (delta 4416), reused 13494 (delta 1932)
Receiving objects: 100% (48961/48961), 131.94 MiB | 5.03 MiB/s, done.
Resolving deltas: 100% (4416/4416), done.
Checking connectivity... done.
Checking out files: 100% (46038/46038), done.
user1@lubuntu-vpc:~/src$
```

```
user1@lubuntu-vpc:~/src$ cd linux-kernel/
user1@lubuntu-vpc:~/src/linux-kernel$ git fetch --depth 10
remote: Counting objects: 52278, done.
remote: Compressing objects: 100% (8166/8166), done.
remote: Total 9541 (delta 4589), reused 3554 (delta 1170)
Receiving objects: 100% (9541/9541), 7.93 MiB | 4.39 MiB/s, done.
Resolving deltas: 100% (4589/4589), completed with 2461 local objects.
user1@lubuntu-vpc:~/src/linux-kernel$
```

カーネルのビルド

cloneしたソースのディレクトリに cd します。

```
user1@lubuntu-vpc:~/src$ cd linux-kernel/
user1@lubuntu-vpc:~/src/linux-kernel$
```

デフォルトブランチ “MA-E3xx/linux-WireGuard-4.19.y-20190601” になっています。(2019/06/11現在 v4.19.49 + WireGuard patch)

```
user1@lubuntu-vpc:~/src/linux-kernel$ git branch
* MA-E3xx/linux-WireGuard-4.19.y-20190601

user1@lubuntu-vpc:~/src/linux-kernel$ cat Makefile |head -6
# SPDX-License-Identifier: GPL-2.0
VERSION = 4
PATCHLEVEL = 19
SUBLEVEL = 49
EXTRAVERSION
NAME = "People's Front"
```

ブランチが異なっている場合、下記手順で変更することができます。

- ブランチの確認

```
user1@lubuntu-vpc:~/src/linux-kernel$ git branch
* MA-E3xx/linux-4.19.y  <-- "MA-E3xx/linux-WireGuard-4.19.y-20190601" ではない
```

- ブランチをチェックアウト

```
user1@lubuntu-vpc:~/src/linux-kernel$ git checkout origin/MA-E3xx/linux-WireGuard-4.19.y-20190601 -b MA-E3xx/linux-WireGuard-4.19.y-20190601
Checking out files: 100% (11260/11260), done.
Branch MA-E3xx/linux-WireGuard-4.19.y-20190601 set up to track remote branch MA-E3xx/linux-WireGuard-4.19.y-20190601 from origin.
Switched to a new branch 'MA-E3xx/linux-WireGuard-4.19.y-20190601'
```

- 再びブランチの確認

```
user1@lubuntu-vpc:~/src/linux-kernel$ git branch
  MA-E3xx/linux-4.19.y
* MA-E3xx/linux-WireGuard-4.19.y-20190601 <-- 切り替わっている
```

デフォルトの config でセットアップします。

```
user1@lubuntu-vpc:~/src/linux-kernel$ make plum_MA_defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  SHIPPED scripts/kconfig/zconf.tab.c
  SHIPPED scripts/kconfig/zconf.lex.c
  SHIPPED scripts/kconfig/zconf.hash.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
user1@lubuntu-vpc:~/src/linux-kernel$
```

カーネルのconfigがこのままでよければ、下記コマンドによりビルドを行います。
最後の "-j5" の部分は、ビルドの並列数指定です。仮想マシンに割り当てたCPU数により増減させてください。
CPU数+1が目安となります。

```
user1@lubuntu-vpc:~/src/linux-kernel$ make CROSS_COMPILE=arm-linux-
gnueabihf- -j5
  CHK     include/config/kernel.release
  CHK     include/generated/uapi/linux/version.h
  HOSTCC  scripts/basic/fixdep
  CHK     include/generated/utsrelease.h
  HOSTCC  scripts/dtc/checks.o
  HOSTCC  scripts/dtc/data.o
  HOSTCC  scripts/genksyms/genksyms.o
  HOSTCC  scripts/dtc/dtc-lexer.lex.o
make[1]: `include/generated/mach-types.h' は更新済みです
  HOSTCC  scripts/genksyms/lex.lex.o
  CC      kernel/bounds.s
  GEN     include/generated/bounds.h
  HOSTCC  scripts/genksyms/parse.tab.o
  CC      arch/arm/kernel/asm-offsets.s
  HOSTCC  scripts/dtc/dtc-parser.tab.o
  GEN     include/generated/asm-offsets.h
  CALL    scripts/checksyscalls.sh
・・・・・中略・・・・・
  LD [M]  net/netfilter/xt_set.ko
  LD [M]  net/netfilter/xt_socket.ko
  LD [M]  net/netfilter/xt_state.ko
  LD [M]  net/netfilter/xt_statistic.ko
  LD [M]  net/netfilter/xt_string.ko
  LD [M]  net/netfilter/xt_tcpmss.ko
  LD [M]  net/netfilter/xt_time.ko
  LD [M]  net/netfilter/xt_u32.ko
user1@lubuntu-vpc:~/src/linux-kernel$
```

uImage形式のカーネルイメージを作成します。

```
user1@lubuntu-vpc:~/src/linux-kernel$ make CROSS_COMPILE=arm-linux-
gnueabihf- LOADADDR=0x80008000 uImage
  CALL    scripts/checksyscalls.sh
  CHK     include/generated/compile.h
  OBJCOPY arch/arm/boot/Image
  Kernel: arch/arm/boot/Image is ready
  XZKERN  arch/arm/boot/compressed/piggy_data
  AS      arch/arm/boot/compressed/piggy.o
  LD      arch/arm/boot/compressed/vmlinux
  OBJCOPY arch/arm/boot/zImage
  Kernel: arch/arm/boot/zImage is ready
  UIMAGE  arch/arm/boot/uImage
```

```
Image Name:   Linux-4.19.49+
Created:      Tue Jun 11 09:02:00 2019
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    4047096 Bytes = 3952.24 KiB = 3.86 MiB
Load Address: 80008000
Entry Point:  80008000
  Kernel: arch/arm/boot/uImage is ready
user1@lubuntu-vpc:~/src/linux-kernel$
```

1) https://github.com/centurysys/linux-kernel-MAE

2) http://git-scm.com/

3) http://ja.wikipedia.org/wiki/Git