

1 Yocto Project について

- Linux Foundation 傘下の組み込み Linux 開発支援プロジェクトです。
- 開発者の負荷を軽減、開発の複雑性を緩和するという目的があります。
- 幅広いハードウェア向けにカスタム Linux ディストリビューションを構築するための開発基盤・ツールを提供しています。

1.1 Yocto Project の開発体制

- 世界中の企業やプロジェクトが開発に参加しています。
 - Intel, TI, Freescale, Mentor, Windriver など
- Yocto は半年に一度リリースされます。
- 2012 年 11 月末に Yocto 1.3 がリリースされました。
- 2013 年 6 月に Yocto 1.4 がリリースされる（予定）。
- これらを決める権限があるのは Yocto Project に参加している企業のメンバーにあります。

1.2 Yocto Project の開発環境構成

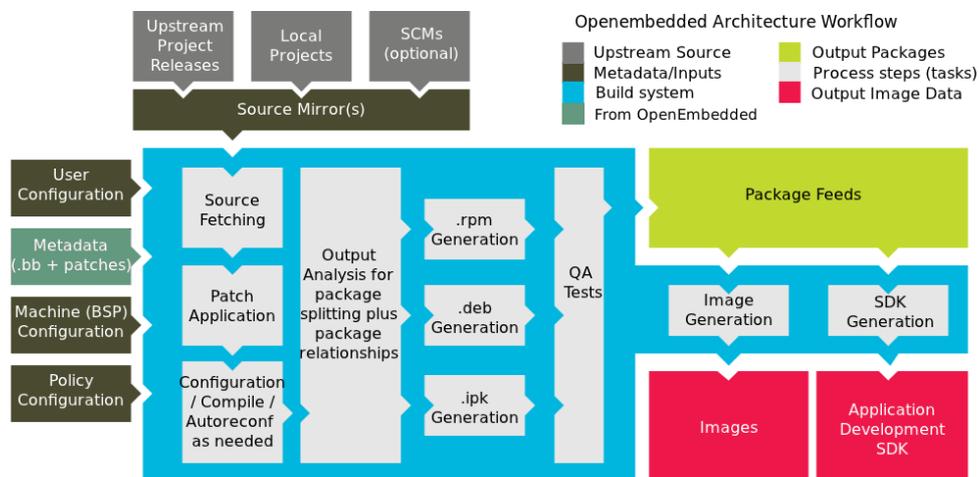


図 1 Yocto Project の開発環境構成図

2 Yocto Project と他のディストリビューションとの比較

2.1 その他のディストリビューションと比べてのメリット

- LTSI カーネルを Yocto のビルドシステム内で利用できます。
- ソースコードベースのディストリビューションです。

- バイナリファイルのチューニングができます。
- ユーザランドのサイズが小さい。
- メンター・グラフィックス、MontaVista、Windriver などの Sier と連携しやすい。
よって、ルネサスの BSP から移行が容易になると考えられます。
(もちろん、各 Sier と話をしておく必要があります)^{*1 *2}

2.2 その他のディストリビューションと比べてデメリット

- ソースからのビルドが必要です。
 - 目的のソフトウェア使うまでに時間がかかります。
 - 別のリソースが必要 (速い CPU, スレッド、ディスク、ネットワーク)
 - * 原因: native 用ツール、クロスツールチェーンを含むフルビルド
 - * 原因: 依存関係の冗長性
 - 同じバイナリが作成されない事があるので、問題の再現が難しい場合があります。
- 必要なソフトウェアが提供されていない場合があります。
- 必要な機能が提供されていない場合があります。

3 提供されるソフトウェア

- Linux カーネル
 - Linux カーネル 3.4 (LTSI)
Yocot Project で提供している Linux カーネルは LTSI 3.4.25 をベースにしています。もちろん A1 (armadillo800eva) の修正もてきようされています。
- ユーザアプリケーションは以下の URL から参照できます。 <https://git.yoctoproject.org/cgit/cgit.cgi/poky/tree/meta>

ソフトウェア名	バージョン
gcc	4.7.2
binutils	2.23.1
eglibc	2.16
gstreamer	0.10.36
directfb	1.6.1

表 1 現時点で提供されるソフトウェアのバージョンの一部

^{*1} http://www.windriver.com/japan/news/press/20120830_Yocto_Project.html

^{*2} http://www.mentorg.co.jp/products/embedded_software/news/2013/130109.html

4 RSO のアクティビティについて

- Yocto 1.4 をターゲットにして A1 / M1 向けのレシピを作成し、デバッグしています。→ meta-renesas
- クロスコンパイラ、開発用のユーザランドの提供を行います（今日、コンパイラとユーザランドを持って来ました）。

5 その他情報

5.1 ルネサスが Yocto Project のメンバになった場合の利点

- ルネサスの CPU・ボードに関するレシピ、メタ情報を管理するデータを Yocto Project のサーバに置くことができます。
- ルネサス向けの BSP バイナリが定期的にリリースされ、Yocto Project のウェブサーバからダウンロードできるようになります。

5.2 Yocto Project に関する情報など

- Yocto の開発、拡張に関するドキュメント（英語）がまとまっています。
<https://www.yoctoproject.org/documentation/current>

6 本日お渡しするクロスツールチェーン、ユーザランドについて

6.1 クロスツールチェーン

6.1.1 注意点

1. ツールチェーンのビルドのホスト環境には Ubuntu-10.10 を使っています。他の環境でも動作すると思いますが、動かない可能性があることをご注意ください。
2. ターゲットを 1.4 にしているので、1.3+snapshot というバージョンになっています。今後これは 1.4 がリリースされた後、またはされる前に、1.4 に変わる予定です。

6.1.2 クロスツールチェーンの使い方

1. poky-eglibc-x86i_64-arm-toolchain-1.3+snapshot-20130123.sh を実行します。
特に指定しない場合、/opt/poky/1.3+snapshot 以下に クロスコンパイラが展開されます。
2. /opt/poky/1.3+snapshot/environment-setup-armv7a-vfp-neon-poky-linux-gnueabi を実行すると、必要な設定を環境変数に登録してくれます。
CC や CXX といった環境変数も上書きするので、実行マシン上で他のアーキテクチャのソフトウェアをコンパイル等する場合は注意が必要です。PATH だけ更新したほうが問題が少なくてよいと思います。

3. クロスツールチェーンによって提供されるコマンドは以下のとおりです。

- arm-poky-linux-gnueabi-addr2line
- arm-poky-linux-gnueabi-gcc-nm
- arm-poky-linux-gnueabi-objdump
- arm-poky-linux-gnueabi-ar
- arm-poky-linux-gnueabi-gcc-ranlib
- arm-poky-linux-gnueabi-ranlib
- arm-poky-linux-gnueabi-as
- arm-poky-linux-gnueabi-gcov
- arm-poky-linux-gnueabi-readelf
- arm-poky-linux-gnueabi-c++filt
- arm-poky-linux-gnueabi-gdb
- arm-poky-linux-gnueabi-size
- arm-poky-linux-gnueabi-cpp
- arm-poky-linux-gnueabi-gprof
- arm-poky-linux-gnueabi-strings
- arm-poky-linux-gnueabi-elfedit
- arm-poky-linux-gnueabi-ld
- arm-poky-linux-gnueabi-strip
- arm-poky-linux-gnueabi-g++
- arm-poky-linux-gnueabi-ld.bfd
- arm-poky-linux-gnueabi-gcc
- arm-poky-linux-gnueabi-nm
- arm-poky-linux-gnueabi-gcc-ar
- arm-poky-linux-gnueabi-objcopy

6.2 ユーザランド

ユーザランドは `core-image-sato-sdk-armadillo800eva-20130123025727.rootfs.tar.bz2` になります。展開して、NFS ルートなどで利用できます。

6.2.1 ビルド方法

ホームディレクトリが `/home/foo` で、`/home/foo/yocto/build` でビルドする場合、以下のように実行します。

1. <http://www.yoctoproject.org/docs/latest/yocto-project-qs/yocto-project-qs.html> を読んで、開発環境を整えます。
2. ホームディレクトリに `yocto` ディレクトリを作成し、移動します。

```
$ cd ; mkdir yocto ; cd yocto
```

3. DVDROM 中にある `poky` と `meta-renesas` を `yocto` ディレクトリにコピーします。

```
$ cp -rf /dvdrom/poky .
$ cp -rf /dvdrom/meta-renesas .
```

4. DVDROM 中にある `bsp-src-20130124.tar.gz` をカレントディレクトリに展開し、`build` という名前に変更します。そして移動します。

```
$ tar -xjf /dvdrom/bsp-src-20130124.tar.bz2 -C .
$ mv bsp-src-20130124 build
$ cd build
```

5. `conf/bblayers.conf` を以下のように修正します。

```
# LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
# changes incompatibly
LCONF_VERSION = "6"

BBPATH = "${TOPDIR}"
BBFILES ?= ""

BBLAYERS ?= " \
    /home/foo/yocto/poky/meta \
    /home/foo/yocto/poky/meta-yocto \
    /home/foo/yocto/poky/meta-yocto-bsp \
    /home/foo/yocto/meta-renesas \
"
BBLAYERS_NON_REMOVABLE ?= " \
    /home/foo/yocto/poky/meta \
    /home/foo/yocto/poky/meta-yocto \
"

```

6. 以下のコマンドを実行し、環境変数を設定します。

```
$ export PATH=/home/foo/yocto/poky/scripts:/home/foo/yocto/poky/bitbake/bin/:$PATH
$ export BB_ENV_EXTRAWHITE="MACHINE DISTRO TCMODE TCLIBC http_proxy ftp_proxy https_proxy \
    all_proxy ALL_PROXY no_proxy SSH_AGENT_PID SSH_AUTH_SOCK BB_SRCREV_POLICY SDKMACHINE \
    BB_NUMBER_THREADS BB_NO_NETWORK PARALLEL_MAKE GIT_PROXY_COMMAND GIT_PROXY_IGNORE \
    SOCKS5_PASSWD SOCKS5_USER SCREENDIR"
$ export BUILDDIR=/home/foo/yocto/build
$ rm pseudodone
$ echo `pwd`/tmp > tmp/saved_tmpdir

```

7. 以下のコマンドを実行し、`core-image-sato-sdk meta-toolchain` を作成します。

```
$ bitbake core-image-sato-sdk meta-toolchain

```

8. `tmp/deploy/images/` と `tmp/deploy/sdk` に各イメージが作成されます。

6.2.2 現時点わかっている問題点

- `/dev/ttySCX` が生成されない

`/dev/ttySCX` が生成されません。udev か、その他設定ファイルの問題だと思っており、調査中です。対応方法としてはユーザランド展開後 `dev` ディレクトリに移動し、以下のコマンドを実行します。

```
$ sudo mknod ttySC1 c 204 9

```

- NFS ブートが失敗する

Yocto Project のユーザランドはディスクからの起動をベースにしている設定になっており、NFS ブートを行うと、正常に起動しません。udev の設定にパッチを適用することで問題は解決します。CD 内に入っているソースにはパッチが適用されています。

```
--- a/meta/recipes-core/udev/udev/init
+++ b/meta/recipes-core/udev/udev/init
@@ -37,7 +37,7 @@ export ACTION=add
 echo "Starting udev"

 # mount the tmpfs on /dev, if not already done
 -LANG=C awk '$2 == "/dev" && ($3 == "tmpfs" || $3 == "devtmpfs") { exit 1 }' /proc/mounts && {
 +LANG=C awk '$2 == "/dev" && ($3 == "tmpfs" || $3 == "devtmpfs") || ($2 == "/" && $3 == "nfs") { exit 1 }' /proc/mounts && {
     mount -n -o mode=0755 -t tmpfs none "/dev"
 }
 [ -e /dev/pts ] || mkdir -m 0755 /dev/pts

```