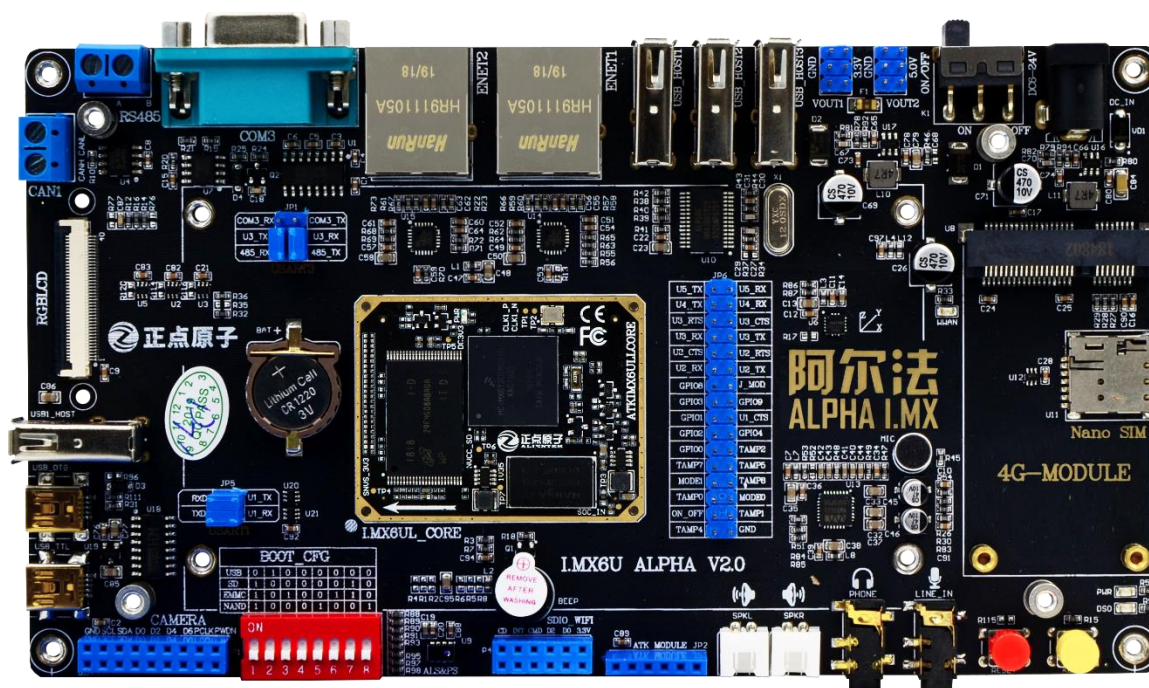


I.MX6U Yocto 构建根文件 系统 V1.0



正点原子 广州市星翼电子科技有限公司

淘宝店铺 1: <http://eboard.taobao.com>

淘宝店铺 2: <http://openedv.taobao.com>

技术支持论坛 (开源电子网) : www.openedv.com

原子哥在线教学: www.yuanzige.com

官方网站: www.alientek.com

最新资料下载链接: <http://www.openedv.com/posts/list/13912.htm>

E-mail: 389063473@qq.com QQ: [389063473](https://www.qq.com/389063473)

咨询电话: [020-38271790](tel:020-38271790)

传真号码: [020-36773971](tel:020-36773971)

团队: [正点原子团队](#)

正点原子, 做最全面、最优秀的嵌入式开发平台软硬件供应商。

友 情 提 示

如果您想及时免费获取“正点原子”最新资料, 敬请关注正点原子微信公众平台, 我们将及时给您发布最新消息和重要资料。



关注方法:

- (1) 微信“扫一扫”, 扫描右侧二维码, 添加关注
- (2) 微信→添加朋友→公众号→输入“正点原子”→关注
- (3) 微信→添加朋友→输入“alientek_stm32”→关注



文档更新说明

版本	版本更新说明	负责人	校审	发布日期
V1.0	初稿:	正点原子 linux 团队		2019.10.26

前言	5
一、在线构建根文件系统	5
1.1 安装与配置 git	5
1.1.1 安装 git	5
1.1.2 配置 git 用户信息	6
1.2 获取 yocto 项目	6
1.2.1 下载 repo	6
1.2.2 获取 yocto 项目	7
1.2.3 构建根文件系统	10
1.2.4 构建 SDK 工具	17
二、离线构建根文件系统	18
2.1 构建根文件系统	19
2.2 构建 SDK 工具	19

前言

yocto: yocto 是这个开源项目的名称, 该项目旨在帮助我们自定义 Linux 系统, 更多 yocto 介绍在此就不作解释了。相信网上的解释会比作者解释的更详细。以下文档内容仅供构建 yocto 根文件系统参考, yocto 是一个非常大的开源项目, 并非作者一个文档就能说的清楚, 作者也在与大家共同学习 yocto。所以作者写这个文档是让读者初步去了解如何构建根文件系统, 减少初学者在构建过程中可能遇到的错误, 更多 yocto 的详细与教程请参照正点原子的其他文档与官方文档。

NXP 官方已经有详细的 yocto 项目使用手册, 文档名字为《i.MX_Yocto_Project_Ubuntu's_Guide_Linux.pdf》, 光盘资料路径为\开发板光盘 A-基础资料\4、参考资料\。网上也有很多 yocto 相关教程, 同学们也可以借鉴参考。文档内容经过作者的实践与试验, 已经成功构建文件系统。若在学习以下文档内容有任何疑问, 请与作者或正点原子技术支持沟通与联系。谢谢!

备注: freescale 已经与 NXP 合并, 以下文档内容不区分他们的名字。

版本说明:

- ✚ Ubuntu 版本: ubuntu-14.04 或以上 (作者 ubuntu 环境是 ubuntu-14.04)
- ✚ Yocto 版本: 4.1.15-4.2.1.0
- ✚ 注意事项:
- ✚ yocto 构建根文件系统需要预留足够的硬盘空间, I.MX yocto 手册上标明至少要留 120G 的空间。
- ✚ yocto 构建根文件系统需要非常长的时间, 不建议用低配置的电脑进行构建。否则可能需要一两天的构建时间。
- ✚ yocto 构建的根文件系统包括了官方评估板所用的一切东西, 文件系统十分巨大, 需要读者自行裁剪。

作者考虑到了上面的因素, 为读者准备了一个在线构建根文件系统的方法, 也就是按照手册的步骤来构建根文件系统 (此方法耗时很长很长, 由读者亲自体验与学习)。另一种离线构建根文件系统 (快餐式构建根文件系统), 是作者准备好 yocto 项目源文件 (正点原子修改过的) 和下载好的相关资源文件, 直接给读者直接构建。可以节省读者在构建 yocto 根文件系统所用的时间, 直接构建出与正点原子 ALPHA 开发板一样功能的文件系统。

一、在线构建根文件系统

本节适用于 yocto 初学者, 作者将一步步与大家分享 NXP yocto 项目的搭建方法。如果读者使用离线构建根文件系统的方法, 可略过第一节 1.1.1~1.2.2 小节。

1.1 安装与配置 git

1.1.1 安装 git

因为 yocto 需要使用 git (用 git 来作版本控制), 如果您已经有安装过 git 并且已经设置过本地用户 git 信息就跳过此步。

Ubuntu# `sudo apt-get install git`

```
alientek@ubuntu:~$ sudo apt-get install git
[sudo] password for alientek:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会安装下列额外的软件包：
  git-man liberror-perl
建议安装的软件包：
  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
下列【新】软件包将被安装：
  git git-man liberror-perl
升级了 0 个软件包，新安装了 3 个软件包，要卸载 0 个软件包，有 453 个软件包未被升级。
需要下载 3,459 kB 的软件包。
解压缩后会消耗掉 22.0 MB 的额外空间。
您希望继续执行吗？ [Y/n] y
获取：1 http://mirrors.aliyun.com/ubuntu/ trusty/main liberror-perl all 0.17-1.1
[21.1 kB]
```

图 1.1.1.1 安装 git

1.1.2 配置 git 用户信息

```
Ubuntu# git config --global user.name "Your Name"
```

```
Ubuntu# git config --global user.email "Your Email"
```

```
Ubuntu# git config --list
```

参数解释：

- --global user.name : 添加用户名
- --global user.email : 添加用户的 email
- --list : 查看列表信息

比如作者设置个人的信息如下

```
Ubuntu# git config --global user.name "DZM@alientek"
```

```
Ubuntu# git config --global user.email "12345678@qq.com"
```

```
Ubuntu# git config --list
```

```
alientek@ubuntu:~$ git config --global user.name "DZM@alientek"
alientek@ubuntu:~$ git config --global user.email "12345678@qq.com"
alientek@ubuntu:~$ git config --list
user.name=DZM@alientek
user.email=12345678@qq.com
alientek@ubuntu:~$
```

图 1.1.2.1 查看配置的 git 用户信息

1.2 获取 yocto 项目

1.2.1 下载 repo

repo 简介：

repo 是一个构建在 Git 之上的工具，它可以更容易地管理包含多个存储库的项目，而这些

存储库不需要位于同一台服务器上。repo 很好地补充了 yocto 项目的分层特性，使用户可以更容易地将自己的层添加到 BSP 中。

简单来说，repo 用来管理多个 git 工程，git 是无法将 freescale 的在存储库里面的某个项目给 clone 下来，如果我们只需要 freescale 的 yocto 项目，(要想知道 freescale 可不只有 yocto 项目呢)，使用 Git 下载却把个存储库项目都下下来了。这就是为什么要用 repo 来管理更庞大的项目的原因了。

首先立工程项目文件夹(文件夹名称可随意)。

Ubuntu# `mkdir fsl-release-bsp`

Ubuntu# `cd fsl-release-bsp`

```
alientek@ubuntu:~$ mkdir fsl-release-bsp
alientek@ubuntu:~$ cd fsl-release-bsp/
alientek@ubuntu:~/fsl-release-bsp$
```

图 1.2.1.1 创建 fsl-release-bsp 文件夹

获取 repo，由于国内外网络环境的原因，官方手册里下载 repo 的链接需要翻墙。下面提供获取 repo 的两种方法，如果您的网络环境不能翻墙，建议使用方法二。(备注：网上下载的 repo 有很多种，经过作者实验网上有很多 repo 在 init 时会出错(需要修改 repo 里面的 REPO_URL 地址才能用)！所以作者已经把测试过 repo 上传到的 github 里。提供给读者参考与使用。)

方法一，官方提供的下载地址，需要翻墙

<http://commondatastorage.googleapis.com/git-repo-downloads/repo>

方法二，正点原子提供的下载链接

<https://raw.githubusercontent.com/dzm2018/alientek-git-repo-downloads/master/repo>

本次演示的是方法二：

Ubuntu# `curl https://raw.githubusercontent.com/dzm2018/alientek-git-repo-downloads/master/repo > ./repo`

```
alientek@ubuntu:~/fsl-release-bsp$ curl https://raw.githubusercontent.com/dzm2018/alientek-git-repo-downloads/master/repo > ./repo
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 17443 100 17443    0     0  17885      0 --:--:-- --:--:-- --:--:-- 17890
alientek@ubuntu:~/fsl-release-bsp$
```

图 1.2.1.2 下载 repo

修改 repo 文件的权限，注意不要使用 root 用户！（注：官方文档这里将 repo 导出环境变量里，作者这里并不这样做。）

Ubuntu# `chmod u+x repo`

```
alientek@ubuntu:~/fsl-release-bsp$ chmod u+x repo
alientek@ubuntu:~/fsl-release-bsp$ ls
repo
alientek@ubuntu:~/fsl-release-bsp$
```

图 1.2.1.3 赋予 repo 可执行权限

1.2.2 获取 yocto 项目

这里教大家选择 yocto 的分支，可以用浏览器打开网址 <http://git.freescale.com/git/cgit.cgi/imx/fsl-arm-yocto-bsp.git> 查看 freescale 的 yocto 项目的分支。根据我们源码内核发布的版本号是 4.1.15-2.10，可以从下图看出 imx-4.1-krogoth 分支就是对应内核 4.1.15-2.1.0 的版本，所以我们选择 imx-4.1-krogoth 分支。

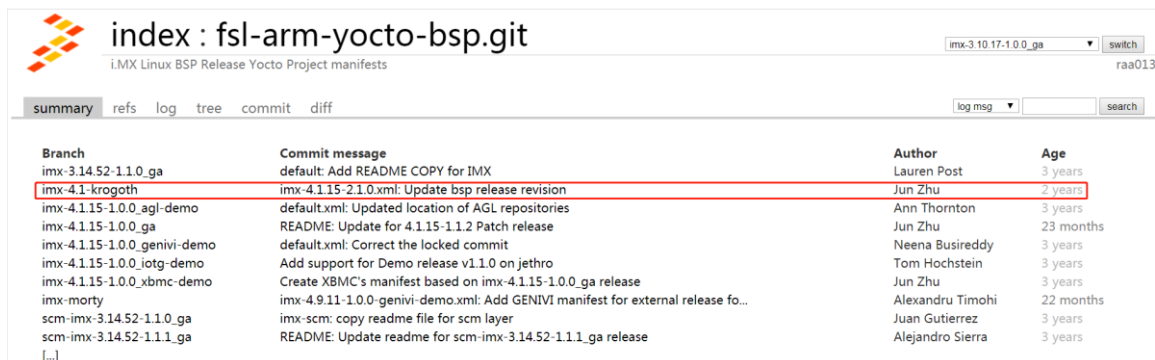


图 1.2.2.1 选择 imx-4.1-krogoth

获取 freescale yocto 项目的 imx-4.1-krogoth 分支，在当前目录下执行下面的指令，执行过程如下。过程中还会确认您的 git 用户信息是否是您在 1.1 节中配置的信息。

Ubuntu# ./repo init -u git://git.freescale.com/imx/fsl-arm-yocto-bsp.git -b imx-4.1-krogoth

```
alientek@ubuntu:~/fsl-release-bsp$ ./repo init -u git://git.freescale.com/imx/fsl-arm-yocto-bsp.git -b imx-4.1-krogoth
Get git://codeaurora.org/tools/repo.git
remote: Counting objects: 4466, done.
remote: Compressing objects: 100% (333/333), done.
remote: Total 4466 (delta 191), reused 0 (delta 0)
接收对象中: 100% (4466/4466), 1.03 MiB | 269.00 KiB/s, done.
处理 delta 中: 100% (3014/3014), done.
来自 git://codeaurora.org/tools/repo
* [新分支] aosp-new/maint -> origin/aosp-new/maint
* [新分支] aosp-new/master -> origin/aosp-new/master
* [新分支] aosp-new/stable -> origin/aosp-new/stable
* [新分支] caf-stable -> origin/caf-stable
* [新分支] master -> origin/master
* [新分支] stable -> origin/stable
* [新标签] v1.0 -> v1.0
* [新标签] v1.0.1 -> v1.0.1
* [新标签] v1.0.2 -> v1.0.2
* [新标签] v1.0.3 -> v1.0.3
* [新标签] v1.0.4 -> v1.0.4
* [新标签] v1.0.5 -> v1.0.5
* [新标签] v1.0.6 -> v1.0.6
* [新标签] v1.0.7 -> v1.0.7
* [新标签] v1.0.8 -> v1.0.8
* [新标签] v1.0.9 -> v1.0.9
* [新标签] v1.1 -> v1.1
* [新标签] v1.10.0 -> v1.10.0
* [新标签] v1.10.1 -> v1.10.1
* [新标签] v1.10.2 -> v1.10.2
* [新标签] v1.10.3 -> v1.10.3
* [新标签] v1.10.4 -> v1.10.4
```

图 1.2.2.2 克隆 imx-4.1-krogoth 分支

连续输入三次 y，完成后可以看到 repo 初始化在当前目录下，使用 ls 指令可以查看隐藏文件夹.repo。

```
* [新分支] imx-4.1.15-1.0.0_ga -> origin/imx-4.1.15-1.0.0_ga
* [新分支] imx-4.1.15-1.0.0_genivi-demo -> origin/imx-4.1.15-1.0.0_genivi-demo
* [新分支] imx-4.1.15-1.0.0_iotg-demo -> origin/imx-4.1.15-1.0.0_iotg-demo
* [新分支] imx-4.1.15-1.0.0_xbmc-demo -> origin/imx-4.1.15-1.0.0_xbmc-demo
* [新分支] imx-morty -> origin/imx-morty
* [新分支] scm-imx-3.14.52-1.1.0_ga -> origin/scm-imx-3.14.52-1.1.0_ga
* [新分支] scm-imx-3.14.52-1.1.1_ga -> origin/scm-imx-3.14.52-1.1.1_ga

Your Name [DZM@alientek]: y
Your Email [12345678@qq.com]: y 连续输入三次 'y'

Your identity is: y <y>
is this correct [y/n]? y

repo initialized in /home/alientek/fsl-release-bsp
alientek@ubuntu:~/fsl-release-bsp$
```

图 1.2.2.3 确认 git 用户信息

Ubuntu# ls -a

```
alientek@ubuntu:~/fsl-release-bsp$ ls -a
.  ..  repo  .repo
alientek@ubuntu:~/fsl-release-bsp$
```

图 1.2.2.4 “.repo” 隐藏文件夹

使用 repo sync 指令同步数据，同步的过程中会克隆 NXP 官方 fsl-arm-yocto-bsp.git 项目的 imx-4.1-krogoth 分支。（注：经过作者实测，上午执行 repo sync 同步下载网速会很快，下午执行 repo sync 同步，下载网速非常慢且有可能服务器无法响应您的下载请求而同步出错。也有可能不同地方会有不同的下载情况，请读者自行选择适合的时间段下载！可能不同地方有不同的情况！）

Ubuntu# repo sync

```
alientek@ubuntu:~/fsl-release-bsp$ ./repo sync
Initializing project fsl-community-bsp-base ...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 271 (delta 1), reused 6 (delta 1), pack-reused 261
接收对象中: 100% (271/271), 60.54 KiB | 0 bytes/s, done.
处理 delta 中: 100% (144/144), done.
来自 git://github.com/Freescale/fsl-community-bsp-base
* [新分支] daisy -> freescale/daisy
* [新分支] danny -> freescale/danny
* [新分支] denzil -> freescale/denzil
* [新分支] dizzy -> freescale/dizzy
* [新分支] dora -> freescale/dora
* [新分支] dylan -> freescale/dylan
* [新分支] fido -> freescale/fido
* [新分支] jethro -> freescale/jethro
* [新分支] krogoth -> freescale/krogoth
* [新分支] master -> freescale/master
* [新分支] master-next -> freescale/master-next
* [新分支] morty -> freescale/morty
* [新分支] pyro -> freescale/pyro
* [新分支] rocko -> freescale/rocko
* [新分支] sumo -> freescale/sumo
* [新分支] thud -> freescale/thud
* [新分支] warrior -> freescale/warrior
* [新标签] 1.2 -> 1.2
* [新标签] 1.2-rc1 -> 1.2-rc1
* [新标签] 1.3 -> 1.3
* [新标签] 1.6 -> 1.6
* [新标签] 1.7 -> 1.7
* [新标签] 1.8 -> 1.8
```

图 1.2.2.5 repo 同步

同步完成如下图，可以看到目录下多了几个文件，其中 sources 文件夹是项目源文件。

```

* [新标签] yocto-2.4.2 -> yocto-2.4.2
* [新标签] yocto-2.4.3 -> yocto-2.4.3
* [新标签] yocto-2.4.4 -> yocto-2.4.4
* [新标签] yocto-2.5 -> yocto-2.5
* [新标签] yocto-2.5.1 -> yocto-2.5.1
* [新标签] yocto-2.5.2 -> yocto-2.5.2
* [新标签] yocto-2.5.3 -> yocto-2.5.3
* [新标签] yocto-2.6 -> yocto-2.6
* [新标签] yocto-2.6.1 -> yocto-2.6.1
* [新标签] yocto-2.6.2 -> yocto-2.6.2
* [新标签] yocto-2.7 -> yocto-2.7
* [新标签] yocto-2.7.1 -> yocto-2.7.1
* [新标签] yocto_1.5_M5.rc8 -> yocto_1.5_M5.rc8
Fetching projects: 100% (9/9), done.

alientek@ubuntu:~/fsl-release-bsp$ ls
fsl-setup-release.sh  README  README-IMXBSP  repo  setup-environment  sources
alientek@ubuntu:~/fsl-release-bsp$

```

图 1.2.2.6 查看项目源文件

1.2.3 构建根文件系统

下面是发行版（DISTRO）的四种配置：

- fsl-imx-x11 -只有 X11 图形
- fsl-imx-wayland - Wayland weston graphics
- fsl-imx-xwayland - Wayland graphics and X11。不支持使用 EGL 的 X11 应用程序
- fsl-imx-fb -帧缓冲图形-没有 X11 或 Wayland

支持的 machine 有很多种，以下是按官方文档列出的几种。

- imx6qpsabreauto
- imx6qpsabresd
- imx6ulevk
- imx6ull14x14evk
- imx6ull9x9evk
- imx6dlsabreauto
- imx6dlsabresd
- imx6qsabreauto
- imx6qsabresd
- imx6slevk
- imx6solosabreauto
- imx6solosabresd
- imx6xsabresd
- imx6xsabreauto
- imx7dsabresd

更多 machine 支持，可以进入 `sources/meta-fsl-bsp-release/imx/meta-bsp/conf/machine` 查看，或者直接用 `ls` 指令查看

```
USER# ls sources/meta-fsl-bsp-release/imx/meta-bsp/conf/machine
```



图 1.2.3.1 查看支持的 machine

构建 yocto 的文件系统镜像，freescale (NXP) 提供了几种镜像供读者构建，文件系统镜像支持的功能越多，文件系统就越大，读者可自行选择一个来构建。

如下表格简单介绍了几种镜像

core-image-minimal	仅支持设备启动的小镜像
core-image-base	目标设备硬件的只支持控制台的镜像
core-image-sato	镜像支持 X11 与 Sato 主题和使用 Pimlico 应用程序。它包含一个终端、一个编辑器和一个文件管理器。
fsl-image-machine-test	FSL 社区的核心镜像，支持控制台环境，没有 GUI 界面
fsl-image-gui	一个支持 GUI 但没有 Qt5 的镜像
fsl-image-qt5	一个开源 Qt 5 的镜像

官方提供了 fsl-setup-release.sh 脚本来构建项目，脚本语法如下：

```
DISTRO=<distro name> MACHINE=<machine name> source fsl-setup-release.sh -b <build dir>
```

参数解释：

DISTRO：发行版类型

MACHINE：机器类型

-b:指定工程构建输出的目录

发行版我们选择 fsl-imx-x11，正点原子的发布的系统需要运行 Qt5 程序，所以选择 fsl-image-qt5 这个文件系统镜像来构建。我们开发板是 IMX6ULL 芯片，所以 MACHINE 选择是 imx6ull14x14evk。fsl-setup-release.sh 这个脚本会解释用户配置参数。

所以我们最终的指令是

```
USER# DISTRO=fsl-imx-x11 MACHINE=imx6ull14x14evk source fsl-setup-release.sh -b build
```

图 1.2.3.2 执行配置指令

这里要特别的注意：构建 yocto 时不要加载任何工具链的环境变量（构建时会使用 yocto 项目的编译器进行编译），请创建新的 shell 或打开新的终端窗口。如果您在 /etc/profile 下导出了与 gcc 相关的环境变量，请把它用 “#” 号注释后重启或者注销 Ubuntu 再执行上面的指令，因为 /etc/profile 下的环境变量会开机就加载，并且在新的 shell 或者终端里始终生效！

配置过程中请你阅读 EULA 内容，输入 q 退出阅读内容后再按 y 确认已经阅读，再继续下一步。

IMPORTANT. Read the following NXP Semiconductors Software License Agreement ("Agreement") completely. By selecting the "I Accept" button at the end of this page, you indicate that you accept the terms of the Agreement and you acknowledge that you have the authority, for yourself or on behalf of your company, to bind your company to these terms. You may then download or install the file.

NXP SEMICONDUCTORS SOFTWARE LICENSE AGREEMENT

This is a legal agreement between you, as an authorized representative of your employer, or if you have no employer, as an individual (together "you"), and Freescale Semiconductor, Inc., a 100% affiliated company of NXP Semiconductors N.V. ("NXP"). It concerns your rights to use the software identified in the Software Content Register and provided to you in binary or source code form and any accompanying written materials (the "Licensed Software"). The Licensed Software may include any updates or error corrections or documentation relating to the Licensed Software provided to you by NXP under this License. In consideration for NXP allowing you to access the Licensed Software, you are agreeing to be bound by the terms of this Agreement. If you do not agree to all of the terms of this Agreement, do not download or install the Licensed Software. If you change your mind later, stop using the Licensed Software and delete all copies of the Licensed Software in your possession or control. Any copies of the Licensed Software that you have already distributed, where permitted, and do not destroy will continue to be governed by this Agreement. Your prior use will also continue to be governed by this Agreement.

1. DEFINITIONS

1.1. "Affiliates" means, any corporation, or entity directly or indirectly controlled by, controlling, or under common control with NXP Semiconductors N.V.

1.2. "Essential Patent" means a patent to the limited extent

Do you accept the EULA you just read? (y/n)

图 1.2.3.3 阅读 EULA 内容

配置完成后会直接进入 build 目录

```

1.1.          "Affiliates" means, any corporation, or entity directly
or indirectly controlled by, controlling, or under common control with NXP
Semiconductors N.V.

1.2.          "Essential Patent" means a patent to the limited extent

Do you accept the EULA you just read? (y/n) y
EULA has been accepted.

Welcome to Freescale Community BSP

The Yocto Project has extensive documentation about OE including a
reference manual which can be found at:
    http://yoctoproject.org/documentation

For more information about OpenEmbedded see their website:
    http://www.openembedded.org/

You can now run 'bitbake <target>'

Common targets are:
    core-image-minimal
    meta-toolchain
    meta-toolchain-sdk
    adt-installer
    meta-ide-support

Your build environment has been configured with:

    MACHINE=imx6ull14x14evk
    SDKMACHINE=i686
    DISTRO=fsl-imx-x11
    EULA=
    BSPDIR=
    BUILD_DIR=.
alientek@ubuntu:~/fsl-release-bsp/build$

```

这里例举了一些可构建的项目内容，并非全部

这里是您配置的内容

图 1.2.3.4 打印配置的内容

输入 `bitbake fsl-image-qt5` 开始构建根文件系统，看到下面报错的内容，这是因为构建 yocto 项目时需要安装一些软件。

USER# `bitbake fsl-image-qt5`

```

alientek@ubuntu:~/fsl-release-bsp/build$ bitbake fsl-image-qt5
NOTE: Your conf/bblayers.conf has been automatically updated.
ERROR: OE-core's config sanity checker detected a potential misconfiguration.
       Either fix the cause of this error or at your own risk disable the checker (see sanity.conf).
       Following is the list of potential problems / advisories:

       Please install the following missing utilities: C++ Compiler (g++),makeinfo,gawk,chrpath

Summary: There was 1 ERROR message shown, returning a non-zero exit code.
alientek@ubuntu:~/fsl-release-bsp/build$

```

图 1.2.3.5 安装必须的软件

执行下面的指令安装构建项目所需要的软件(这里与官方的构建 yocto 项目的顺序不一样，官方的是先安装软件包，作者的是遇到需要安装软件包时再去安装这些软件包)

USER# `sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libssl1.2-dev curl`


```

allientek@ubuntu:~/fsl-release-bsp/build$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libstd1.2-dev curl
[sudo] password for allientek:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
diffstat 已经是最新的版本。
diffstat 被设置为手动安装。
unzip 已经是最新的版本。
unzip 被设置为手动安装。
将会安装下列额外的软件包：
  cpp-4.8 dpkg-dev fakeroot g++ g++-4.8 gcc-4.8 gcc-4.8-base gcc-4.8-multilib
  lib32asan0 lib32atomic1 lib32gcc-4.8-dev lib32gcc1 lib32gomp1 lib32itm1
  lib32quadmath0 libalgorithm-diff-perl libalgorithm-diff-xs-perl

```

图 1.2.3.6 执行安装软件指令

安装完成以上的软件后，再执行一次 bitbake fsl-image-qt5 指令重新构建。如下图：

USER# bitbake fsl-image-qt5

```

Creating config file /etc/perl/XML/SAX/ParserDetails.ini with new version
正在设置 libxml-libxml-perl (2.0108+dfsg-1ubuntu0.2) ...
update-perl-sax-parsers: Registering Perl SAX parser XML::LibXML::SAX::Parser with priority 50...
update-perl-sax-parsers: Registering Perl SAX parser XML::LibXML::SAX with priority 50...
update-perl-sax-parsers: Updating overall Perl SAX parser modules info file...
Replacing config file /etc/perl/XML/SAX/ParserDetails.ini with new version
正在设置 texinfo (5.2.0.dfsg.1-2) ...
正在设置 libwww-perl (6.05-2) ...
正在设置 libxml-parser-perl (2.41-1build3) ...
正在设置 libxml-sax-expat-perl (0.40-2) ...
update-perl-sax-parsers: Registering Perl SAX parser XML::SAX::Expat with priority 50...
update-perl-sax-parsers: Updating overall Perl SAX parser modules info file...
Replacing config file /etc/perl/XML/SAX/ParserDetails.ini with new version
正在处理用于 libc-bin (2.19-0ubuntu6.9) 的触发器 ...
allientek@ubuntu:~/fsl-release-bsp/build$ bitbake fsl-image-qt5

```

图 1.2.3.7 执行指令构建 fsl-image-qt5

构建的时长的因素与以下几点有关：计算机的性能（硬盘写入速度，CPU 的核心数），网络下载速度，还与您的 Ubuntu 虚拟机在安装时分配的核心数与分配的内存有关！**构建过程中可能会出错，可按 Ctrl +c 终止，再次输入 bitbake fsl-image-qt5 构建。**yocto 支持从断点处重新构建。因为构建过程中已经生成缓存文件，重复构建时会跳过已经构建过的任务。

可以看我们配置的信息如下图

```

update-perl-sax-parsers: Registering Perl SAX parser XML::SAX::Expat with priority 50...
update-perl-sax-parsers: Updating overall Perl SAX parser modules info file...
Replacing config file /etc/perl/XML/SAX/ParserDetails.ini with new version
正在处理用于 libc-bin (2.19-0ubuntu6.9) 的触发器 ...
allientek@ubuntu:~/fsl-release-bsp/build$ bitbake fsl-image-qt5
NOTE: Your conf/bblayers.conf has been automatically updated.
Parsing recipes: 100% [#####] Time: 00:00:23
Parsing of 2216 .bb files complete (0 cached, 2216 parsed). 2791 targets, 212 skipped, 5 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION      = "1.30.0"
BUILD_SYS       = "x86_64-linux"
NATIVELSBSTRING = "Ubuntu-14.04"
TARGET_SYS      = "arm-poky-linux-gnueabi"
MACHINE         = "imx6ull14x14evk"
DISTRO          = "fsl-imx-x11"
DISTRO_VERSION  = "4.1.15-2.1.0"
TUNE_FEATURES   = "arm armv7ve vfp neon          callconvention-hard          cortexa7"
TARGET_FPU      = "hard"
meta
meta-poky       = "HEAD:f5da2a5913319ad6ac2141438ba1aa17576326ab"
meta-oe         = "HEAD:247b1267bbe95719cd4877d2d3cfbaf2a2f4865a"
meta-multimedia = "HEAD:be78894e4682f111575470fb23e51e6ba523508d"
meta-fsl-arm    = "HEAD:3dfb82fc7e703eae9891b3ffda0e9393701f2396"
meta-fsl-arm-extra = "HEAD:a165068f8a0d1cf29aabe4b4053f28be1c2aa492"
meta-bsp        = "HEAD:823b26a67261270d2bf22d511e6190641a8a90cf"
meta-sdk        = "HEAD:77736988073a5d90fcff9d0005c8477332ede387"
meta-browser    = "HEAD:77736988073a5d90fcff9d0005c8477332ede387"
meta-gnome
meta-networking
meta-python
meta-filesystems = "HEAD:247b1267bbe95719cd4877d2d3cfbaf2a2f4865a"
meta-qt5        = "HEAD:ccae79be69c5268df3b47e4e14cea0591c39a531"

NOTE: Preparing RunQueue
NOTE: Executing SetScene Tasks
NOTE: Executing RunQueue Tasks

```

图 1.2.3.8 构建内容的信息

构建过程中会下载很多软件包，下载目录为../downloads。也就是 build 目录的前一级目录下。第一次构建时需要下载这些软件包，所以耗时很长。如果再次构建，那么无需重复下载，所以再次构建项目时速度会快很多。下图黄色的警告是构建任务中下载软件包失败时所提示的，并不会影响构建过程。

请读者耐心等待文件系统的构建完成，下图的任务总数为 7756 个，不同的构建内容，任务总数可能是不相同的。

```
meta-bsp
meta-sdk      = "HEAD:823b26a67261270d2bf22d511e6190641a8a90cf"
meta-browser  = "HEAD:77736988073a5d90fcff9d0005c8477332ede387"
meta-gnome
meta-networking
meta-python
meta-filessystems = "HEAD:247b1267bbe95719cd4877d2d3cfbaf2a2f4865a"
meta-qt5       = "HEAD:ccae79be69c5268df3b47e4e14cea0591c39a531"

NOTE: Preparing RunQueue
NOTE: Executing SetScene Tasks
NOTE: Executing RunQueue Tasks
WARNING: zlib-native-1.2.8-r0 do_fetch: Failed to fetch URL http://www.zlib.net/zlib-1.2.8.tar.xz, attempting MIRRORS if available
WARNING: libmpc-native-1.0.3-r0 do_fetch: Failed to fetch URL http://www.multiprecision.org/mpc/download/mpc-1.0.3.tar.gz, attempting MIRRORS if available
WARNING: bzip2-1.0.6-r5 do_fetch: Checksum mismatch for local file /home/alientek/fsl-release-bsp/downloads/bzip2-1.0.6.tar.gz
Cleaning and trying again.
WARNING: bzip2-1.0.6-r5 do_fetch: Renaming /home/alientek/fsl-release-bsp/downloads/bzip2-1.0.6.tar.gz to /home/alientek/fsl-release-bsp/downloads/bzip2-1.0.6.tar.gz_bad-checksum_527819af097ee3014e4d442315b6f283
WARNING: bzip2-1.0.6-r5 do_fetch: Checksum failure encountered with download of http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz - will attempt other sources if available
WARNING: pigz-native-2.3.3-r0 do_fetch: Failed to fetch URL http://zlib.net/pigz/pigz-2.3.3.tar.gz, attempting MIRRORS if available
WARNING: cryptodev-linux-native-1.8-r0 do_fetch: Failed to fetch URL http://download.gna.org/cryptodev-linux/cryptodev-linux-1.8.tar.gz, attempting MIRRORS if available
WARNING: popt-native-1.16-r3 do_fetch: Failed to fetch URL http://rpm5.org/files/popt/popt-1.16.tar.gz, attempting MIRRORS if available
WARNING: rpm-native-5.4.16-r0 do_fetch: Failed to fetch URL http://www.rpm5.org/files/rpm/rpm-5.4/rpm-5.4.15-0.20140824.src.rpm;name=src.rpm;extract=rpm-5.4.15.tar.gz, attempting MIRRORS if available
Currently 16 running tasks (170 of 7756):
0: gcc-source-5.3.0-5.3.0-r0 do_fetch (pid 28235)
1: binutils-cross-arm-2.26-r0 do_fetch (pid 28339)
2: linux-libc-headers-4.1-r0 do_fetch (pid 28392)
3: glibc-initial-2.23-r0 do_fetch (pid 28402)
4: glibc-2.23-r0 do_fetch (pid 28418)
5: gettext-native-0.19.6-r0 do_fetch (pid 28644)
6: openssl-native-1.0.2h-r0 do_fetch (pid 34410)
7: gettext-0.19.6-r0 do_fetch (pid 34654)
8: libpcre-native-8.38-r0 do_fetch (pid 35370)
9: python-native-2.7.11-r1.1 do_fetch (pid 36198)
10: libpcre-8.38-r0 do_fetch (pid 36509)
11: file-native-5.25-r0 do_fetch (pid 39704)
12: byacc-native-20150711-r0 do_fetch (pid 45418)
13: rpm-native-5.4.16-r0 do_fetch (pid 48375)
14: xz-native-5.2.2-r0 do_configure (pid 54832)
15: gmp-native-6.1.0-r0 do_configure (pid 54833)
```

图 1.2.3.9 构建过程截图

构建成功如下图，没有打印错误信息。（备注：作者 Ubuntu 分配 16GB 内存，16 个核心数，网速为 100M，大约构建了 9 个小时！）

```

meta-networking
meta-python
meta-fileSystems = "HEAD:247b1267bbe95719cd4877d2d3cfbaf2a2f4865a"
meta-qt5         = "HEAD:ccae79be69c5268df3b47e4e14cea0591c39a531"

NOTE: Preparing RunQueue
NOTE: Executing SetScene Tasks
NOTE: Executing RunQueue Tasks
WARNING: minicon-2.7-r0 do_fetch: Failed to fetch URL https://alioth.debian.org/frs/download.php/latestfile/3/minicon-2.7.tar.gz, attempting MIRRORS if available
WARNING: nano-2.2.5-r3.0 do_fetch: Failed to fetch URL ftp://nano-editor.org/pub/nano/v2.2/nano-2.2.5.tar.gz, attempting MIRRORS if available
WARNING: vlan-1.9-r0 do_fetch: Failed to fetch URL http://vlan.sourceforge.net/downloads/1.9-3ubuntu9/vlan-1.9.orig.tar.gz, attempting MIRRORS if available
WARNING: bonnie++-1.03e-r0 do_fetch: Failed to fetch URL http://www.coker.com.au/bonnie++/bonnie++-1.03e.tgz, attempting MIRRORS if available
WARNING: ed-1.9-r0 do_fetch: Failed to fetch URL http://ftp.gnu.org/gnu/ed/ed-1.9.tar.gz, attempting MIRRORS if available
WARNING: logrotate-3.9.1-r0 do_fetch: Checksum mismatch for local file /home/allientek/fsl-release-bsp/downloads/logrotate-3.9.1.tar.gz. Cleaning and trying again.
WARNING: logrotate-3.9.1-r0 do_fetch: Renaming /home/allientek/fsl-release-bsp/downloads/logrotate-3.9.1.tar.gz to /home/allientek/fsl-release-bsp/downloads/logrotate-3.9.1.tar.gz_bad-checksum_687e2c1b54ef6caa6ceaca586b388e78
WARNING: logrotate-3.9.1-r0 do_fetch: Checksum failure encountered with download of https://fedorahosted.org/releases/l/o/logrotate/logrotate-3.9.1.tar.gz - will attempt other sources if available
WARNING: cronie-1.5.0-r0 do_fetch: Checksum mismatch for local file /home/allientek/fsl-release-bsp/downloads/cronie-1.5.0.tar.gz. Cleaning and trying again.
WARNING: cronie-1.5.0-r0 do_fetch: Renaming /home/allientek/fsl-release-bsp/downloads/cronie-1.5.0.tar.gz to /home/allientek/fsl-release-bsp/downloads/cronie-1.5.0.tar.gz_bad-checksum_9116b617b65b1161be95d8782d57b641
WARNING: cronie-1.5.0-r0 do_fetch: Checksum failure encountered with download of https://fedorahosted.org/releases/c/r/cronie/cronie-1.5.0.tar.gz - will attempt other sources if available
WARNING: mktemp-1.7-r3 do_fetch: Failed to fetch URL ftp://ftp.mktemp.org/pub/mktemp/mktemp-1.7.tar.gz, attempting MIRRORS if available
WARNING: xserver-common-1.34-r8 do_fetch: Failed to fetch URL http://gpe.linuxtogo.org/download/source/xserver-common-1.34.tar.gz, attempting MIRRORS if available
WARNING: xtscale-0.6.3-r13 do_fetch: Failed to fetch URL http://gpe.linuxtogo.org/download/source/xtscale-0.6.3.tar.bz2, attempting MIRRORS if available
WARNING: matchbox-theme-sato-0.1-r1 do_fetch: Failed to fetch URL http://pokylinux.org/releases/sato/matchbox-theme-sato-0.1.tar.gz, attempting MIRRORS if available
WARNING: xset-1.1.2.3-r0 do_fetch: Failed to fetch URL http://xorg.freedesktop.org/releases/individual/app/xset-1.2.3.tar.bz2, attempting MIRRORS if available
WARNING: lttng-modules-2.7.1+gitAUTOINC+45e0ebd91f-r0 do_package: lttng-modules: no modules were created; this may be due to CONFIG_TRACEPOINTS not being enabled in your kernel.
NOTE: Tasks Summary: Attempted 7756 tasks of which 2105 didn't need to be rerun and all succeeded.

Summary: There were 17 WARNING messages shown.
allientek@ubuntu:~/fsl-release-bsp/build$

```

图 1.2.3. 10 构建完成截图

构建完成后进入../build/tmp/deploy/images/imx6ull14x14evk 目录下用 ls 指令查看生成的文件。

```

allientek@ubuntu:~/fsl-release-bsp/build/tmp/deploy/images/imx6ull14x14evk$ ls
fsl-image-qt5-imx6ull14x14evk-20190904030729.rootfs.ext4      u-boot-sd-2016.03-r0.imx
fsl-image-qt5-imx6ull14x14evk-20190904030729.rootfs.manifest  zImage
fsl-image-qt5-imx6ull14x14evk-20190904030729.rootfs.sdcard    zImage--4.1.15-r0-imx6ull14x14evk-20190904030729.bin
fsl-image-qt5-imx6ull14x14evk-20190904030729.rootfs.tar.bz2   zImage--4.1.15-r0-imx6ull-14x14-evk-20190904030729.dtb
fsl-image-qt5-imx6ull14x14evk.ext4                             zImage--4.1.15-r0-imx6ull-14x14-evk-btwifi-20190904030729.dtb
fsl-image-qt5-imx6ull14x14evk.manifest                         zImage--4.1.15-r0-imx6ull-14x14-evk-emmc-20190904030729.dtb
fsl-image-qt5-imx6ull14x14evk.sdcard                           zImage--4.1.15-r0-imx6ull-14x14-evk-gpmi-weim-20190904030729.dtb
fsl-image-qt5-imx6ull14x14evk.tar.bz2                         zImage--4.1.15-r0-imx6ull-14x14-evk-usb-cert-20190904030729.dtb
modules--4.1.15-r0-imx6ull14x14evk-20190904030729.tgz        zImage-imx6ull14x14evk.bin
modules-imx6ull14x14evk.tgz                                    zImage-imx6ull-14x14-evk-btwifi.dtb
README--DO_NOT_DELETE_FILES_IN_THIS_DIRECTORY.txt            zImage-imx6ull-14x14-evk.dtb
u-boot.imx                                                     zImage-imx6ull-14x14-evk-emmc.dtb
u-boot-imx6ull14x14evk.imx                                     zImage-imx6ull-14x14-evk-gpmi-weim.dtb
u-boot-imx6ull14x14evk.imx-sd                                  zImage-imx6ull-14x14-evk-usb-cert.dtb
u-boot.imx-sd
allientek@ubuntu:~/fsl-release-bsp/build/tmp/deploy/images/imx6ull14x14evk$

```

图 1.2.3. 11 构建完成的“产物”

生成的文件解释:

生成的文件中, 其中蓝色文字内容的是链接文件, 这里就不列出了。

文件名	用途
*.rootfs.manifest	文件系统内的软件列表
*.rootfs.ext4	ext4 格式的文件系统压缩包
*.rootfs.sdcard	可直接写入 SD 卡, 从 SD 卡启动的镜像
*.rootfs.tar.bz2	tar.bz2 格式的文件系统压缩包
modules*.tgz	ttgz 格式的内核模块压缩包
u-boot*.imx	官方多版本类型 u-boot 镜像
zImage*/zImage*.bin	内核设备树与 bin 文件

我们需要的文件系统镜像压缩包是 fsl-image-qt5-imx6ull14x14evk-20190904030729.rootfs.tar.

bz2, 文件系统比较大, 读者按需裁剪, 或者按需修改 yocto 项目源文件再构建。

备注: 因存储设备的大小限制, 正点原子还构建了 fsl-image-gui 这个不含 Qt5 但含 GUI 的镜像。

1.2.4 构建 SDK 工具

构建完成我们含 Qt5 的文件系统后, 还需要构建 SDK 工具。SDK 工具里面含交叉编译工具链, 我们需要含 Qt5 的交叉编译工具链。

在 build 目录下执行下面的指令 (注: 请在构建根文件系统后不要退出当前终端, 否则还需要重新输入 DISTRO=fsl-imx-x11 MACHINE=imx6ull14x14evk source fsl-setup-release.sh -b build 来使用环境变量才能使用 bitbake 指令)

USER# bitbake meta-toolchain-qt5

```
allientek@ubuntu:~/fsl-release-bsp/build$ bitbake meta-toolchain-qt5
Loading cache: 100% |#####| ETA: 00:00:00
Loaded 2792 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION      = "1.30.0"
BUILD_SYS       = "x86_64-linux"
NATIVELSBSTRING = "Ubuntu-14.04"
TARGET_SYS      = "arm-poky-linux-gnueabi"
MACHINE         = "imx6ull14x14evk"
DISTRO          = "fsl-imx-x11"
DISTRO_VERSION  = "4.1.15-2.1.0"
TUNE_FEATURES   = "arm armv7ve vfp neon          callconvention-hard          cortexa7"
TARGET_FPU      = "hard"
meta            = "HEAD:f5da2a5913319ad6ac2141438ba1aa17576326ab"
meta-poky       = "HEAD:f5da2a5913319ad6ac2141438ba1aa17576326ab"
meta-oe         = "HEAD:f5da2a5913319ad6ac2141438ba1aa17576326ab"
meta-multimedia = "HEAD:247b1267bbe95719cd4877d2d3cfbaf2a2f4865a"
meta-fsl-arm    = "HEAD:be78894e4682f111575470fb23e51e6ba523508d"
meta-fsl-arm-extra = "HEAD:3dfb82fc7e703eae9891b3ffda0e9393701f2396"
meta-fsl-demos  = "HEAD:a165068f8a0d1cf29aabe4b4053f28be1c2aa492"
meta-bsp        = "HEAD:823b26a67261270d2bf22d511e6190641a8a90cf"
meta-sdk        = "HEAD:823b26a67261270d2bf22d511e6190641a8a90cf"
meta-browser    = "HEAD:77736988073a5d90cfff9d0005c8477332ede387"
meta-gnome      = "HEAD:77736988073a5d90cfff9d0005c8477332ede387"
meta-networking = "HEAD:77736988073a5d90cfff9d0005c8477332ede387"
meta-python     = "HEAD:77736988073a5d90cfff9d0005c8477332ede387"
meta-filesystems = "HEAD:247b1267bbe95719cd4877d2d3cfbaf2a2f4865a"
meta-qt5        = "HEAD:ccae79be69c5268df3b47e4e14cea0591c39a531"

NOTE: Preparing RunQueue
```

图 1.2.4.1 构建 meta-toolchain-qt5

下图为构建过程截图, 可以看到构建的速度快了很多, 因为我们已经构建过文件系统, 已经下载过软件包, 同时在构建时生成了缓存文件。

```
Currently 14 running tasks (2907 of 4598):
0: nativesdk-linux-libc-headers-4.4-r0 do_fetch (pid 99565)
1: nativesdk-glibc-initial-2.23-r0 do_unpack (pid 100431)
2: nativesdk-glibc-2.23-r0 do_unpack (pid 100450)
3: binutils-crosssdk-x86_64-2.26-r0 do_unpack (pid 103484)
4: nativesdk-qtbase-5.6.1+gitAUTOINC+7bf002c3b3-r0 do_unpack (pid 108818)
5: nativesdk-qtdeclarative-5.6.1+gitAUTOINC+a05f59388e-r0 do_unpack (pid 109774)
6: nativesdk-qemu-2.5.0-r1 do_unpack (pid 111008)
7: nativesdk-elftosb-10.12.01-r4 do_fetch (pid 112357)
8: nativesdk-u-boot-mkimage-v2016.03+gitAUTOINC+df61a74e68-r0 do_fetch (pid 112489)
9: nativesdk-imx-usb-loader-1.0+AUTOINC+349286e25c-r0 do_fetch (pid 112584)
10: nativesdk-rpm-5.4.16-r0 do_patch (pid 114892)
11: binutils-cross-canadian-arm-2.26-r0 do_fetch (pid 120019)
12: gdb-cross-canadian-arm-7.10.1-r0 do_unpack (pid 120566)
13: pixz-native-1.0.6-r0 do_fetch (pid 123161)
```

图 1.2.4.2 构建过程截图

构建成功如下图

```
WARNING: nativesdk-linux-libc-headers-4.4-r0 do_fetch: Failed to fetch URL http://kernel.org/pub/linux/kernel/v4.x/linux-4.4.tar.xz, a
ttempting MIRRORS if available
NOTE: Tasks Summary: Attempted 4598 tasks of which 3394 didn't need to be rerun and all succeeded.

Summary: There was 1 WARNING message shown.
alientek@ubuntu:~/fsl-release-bsp/build$
```

图 1.2.4.3 构建成功截图

构建成功后，进入../build/tmp/deploy/sdk 目录，查看生成的文件。其中 fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh 是我们需要的文件，如下图

```
alientek@ubuntu:~/fsl-release-bsp/build/tmp/deploy/sdk$ ls
fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.host.manifest
fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.target.manifest
alientek@ubuntu:~/fsl-release-bsp/build/tmp/deploy/sdk$
```

图 1.2.4.4 构建得到的 SDK 工具

编译出来的 fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh 的文件可直接执行安装，默认安装在/opt 目录下。使能工具链后可用于编译内核、uboot、c 文件与 Qt 应用程序。

使用方法请参考《【正点原子】I.MX6U 用户快速体验》第四章来搭建交叉编译环境。

二、离线构建根文件系统

小节前言

本节适用于个人快速构建根文件系统。建议读者请先阅读在线构建根文件系统的方法再阅读以下内容，避免在构建时出现错误。

由于 yocto 构建前需要下载约 8GB 的软件包到本地，为了快速构建，作者已经把相关的软件和 yocto 资源文件打包成 fsl-release-bsp.tar.xz，因为光盘只能存放 4GB 以下的文件，所以这个文件放不到光盘，我们把它放到百度云了，下面提供百度的下载链接地址 <https://pan.baidu.com/s/1vRrFTPRlpxhw56F02purNQ>。下载后，直接解压使用，减少重复下载软件包的时间。

读者可基于正点原子提供已经配置与测试的 yocto 项目源文件搭配下载好的资源文件进行二次开发。或者用于直接构建与正点原子发布的根文件系统一样功能的根文件系统。

构建前准备，请 fsl-release-bsp.tar.xz 拷贝到 ubuntu 虚拟机，本文拷贝到~（家目录）。如下图：

```
alientek@ubuntu:~$ ls
examples.desktop fsl-release-bsp.tar.xz github test yocto 公共的 模板 视频 图片 文档 下载 音乐 桌面
alientek@ubuntu:~$
```

图 2.1.1 拷贝 fsl-release-bsp.tar.xz 到虚拟机

执行下面的指令解压，因为文件比较大，解压时间需要几分钟，读者请耐心等待。

USER# tar -Jvxf fsl-release-bsp.tar.xz

解压结果如下图

```
alientek@ubuntu:~$ ls
examples.desktop fsl-release-bsp fsl-release-bsp.tar.xz
alientek@ubuntu:~$
```

图 2.1.2 解压 fsl-release-bsp.tar.xz

使用 cd 指令进入 fsl-release-bsp 文件夹用 ls 查看有如下文件

```
downloads fsl-setup-release.sh README README-IMXBSP setup-environment sources
```

图 2.1.3 进入 fsl-release-bsp 文件夹查看内容

2.1 构建根文件系统

方法与在线构建根文件系统的方法一样,请读者从第一节在线构建根文件系统的 1.2.3 小节构建根文件系统的方法阅读(作者离线构建根文件系统时长约 2 个小时仅供参考)。

2.2 构建 SDK 工具

方法与在线构建 SDK 工具的方法一样,请读者从第一节的 1.2.4 小节构建 SDK 工具的方法开始阅读。