

周翔的专栏

努力终有回报 <https://github.com/ZhouBox>

目录视图

摘要视图

RSS 订阅

个人资料



zhx6044

关注

发私信

访问：615054次

积分：8033

等级：BLOG > 5

排名：第2934名

原创：213篇

转载：18篇

译文：4篇

评论：260条

文章搜索

博客专栏



架构设计

文章：27篇

阅读：79397



C++

文章：35篇

阅读：51275



qt

文章：53篇

阅读：242491

文章分类

C/C++ (85)

qt (53)

设计模式 (13)

Linux (26)

Yotoc Project快速指南

标签：Yocto 嵌入式Linux 编译系统

2015-10-19 14:47

3099人阅读

评论(0)

收藏

举报

分类：

Linux (25) 嵌入式 (15) Yocto

目录(?)

[+]

原文路径: <http://www.yoctoproject.org/docs/1.8/mega-manual/mega-manual.html>

1.欢迎

欢迎来到Yocto项目!Yocto项目是一个专注于嵌入式Linux开发者的开源协作项目。在其他方面，Yocto项目使用一个基于OpenEmbedded项目的编译系统，这个编译系统使用BitBake工具来构造完整的linux系统镜像。BitBake和OE的组件被组合形成了Poky，一个参考的编译系统。

如果你没有一个运行linux的系统，但是你运行Yocto项目的测试例子，你可以考虑使用Yocto项目的Build Appliance。Build Appliance可以让你借助Yocto项目构建启动一个自定义的嵌入式linux系统，但是你可以使用一个不是linux的系统环境。您可以通过Yocto Project Build Appliance获得更多的信息。

另一方面，如果你了解所有关于开源开发，linux开发环境，git源码管理或者你仅仅想获得一个有用的信息在你的linux系统上试试Yocto项目，直接跳转到本指南的“Super User”章节。

在下面的时间中，这篇短短的文档将带给你一些关于环境的基础信息和让你通过一种最简单的形式来体验它。在读了这篇文章之后，你将对Yocto项目内容有个基本的了解和怎么使用它的核心组件。这篇文档通过一个怎么编译一个小型的系统和在QEMU上运行来的例子来一步步指导你。

为了获得更多关于Yocto的细节信息，你可以看看下面的资源：

* 网站：[Yocto Project Website](#) 提供了最新的构建，突发新闻，完善的开发文档和一个富有活力的社区。

* FAQs：列出了关于Yocto项目的问题和答案。你可以找到两个FAQs：[Yocto Priject FAQ](#)，其在wiki上，和“FAQ”章在Yocto Project的参考手册中。

* 开发者截屏视频：[The Getting Started with the Yocto Project - New Developer Screencast Tutorial](#) 提供了一个时长30分钟的视频，对于熟悉linux编译系统的用户。虽然这个视频有点过时，但熟悉linux编译系统的用户。虽然这个视频有点过时，但熟悉linux编译系统的用户。

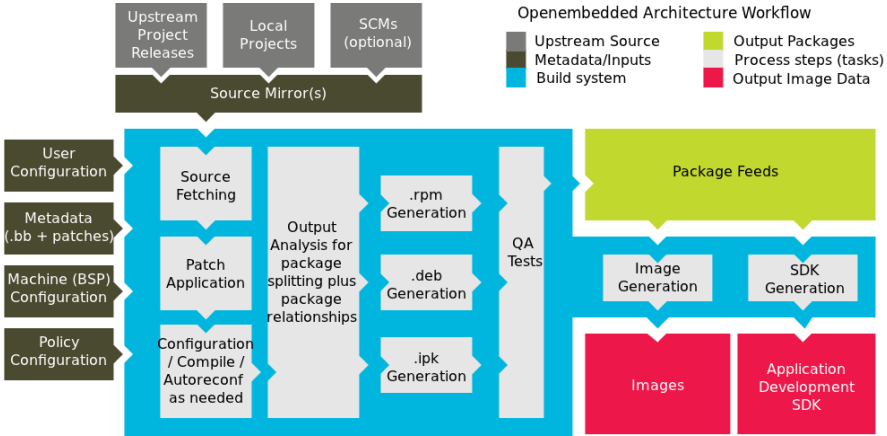
2.Yocto项目开发环境介绍

Yocto项目通过OpenEmbedded的编译系统提供了一ARM,MIPS,PowerPC和x86多种架构提供了多种平台使用Yocto项目的组件来设计，开发，构建，调试，模Window系统，GTK+框架和Qt框架。

嵌入式	(16)
数据结构与算法	(15)
菜鸟的日子	(8)
UML	(1)
java	(6)
C++模板	(26)
杂项	(7)
openCV	(9)
linux服务器	(15)
perl学习	(9)
感悟	(3)
hadoop	(1)
linux驱动学习	(6)
51单片机	(1)
Closure学习	(0)
mongoDB	(4)
IT疯人疯语	(5)
windows编程	(1)
写给女朋友看的C++教程	(0)
PHP	(2)
IOS	(3)
android	(8)
boost.asio	(3)
物联网平台构建	(0)
scrapy	(2)
zigbee	(1)
python	(5)
docker	(1)
nodejs	(7)
express	(5)
nginx	(5)
fastcgi	(2)
Yocto	(1)
架构学习	(4)

文章存档	
2016年06月	(1)
2016年05月	(3)
2016年04月	(2)
2016年03月	(6)
2016年02月	(2)
展开	

阅读排行	
使用nginx作为websocket的...	(34924)
移植qt5.3.1到arm	(24779)
qwt学习一	(23066)
Linux 文件锁	(19777)
qt并行计算框架QtConcurren...	(12397)
Qt mvc学习一	(12235)
qt外部数据传入实现动态的折...	(11992)
使用Scrapy来爬取自己的CSD...	(11641)
ZiqBee组网	(11368)
Qt5摄像头截图	(11239)



The Yocto Project Development Environment

下面是关于Yocto项目的一些集锦：

- * 提供了适合于嵌入式环境的包含一系列命令集和库的较新的linux内核
- * 可以编译一些系统组件比如X11，GTK+，Qt，Clutter和SDL等，你可以获得好的用户体验在有显示器件的设备上。对于没有显示器件的设备上或者你希望使用其他的UI框架，这些组件就不需要包含。
- * 构建一个专注和稳定的兼容OpenEmbedded项目的核心，通过它你可以容易和可靠的构建和开发。
- * 通过QEMU模拟来可以充分的支持大范围的硬件和设备。

Yocto项目能够生成多种不同设备的镜像。然而，基于QEMU的全系统模拟器适用于x86，x86-64，ARM，MIPS和PPC-base架构，以及一些特殊的硬件，比如因特尔的桌面板-DH55TC。因为一个用Yocto项目开发的镜像能够用一个QEMU模拟器启动，开发环境能够像在一个测试的开发嵌入式软件平台一样工作良好。

一个重要的Yocto项目的重要特性是Sato用户界面。这个基于GTK+的可选的UI主要面向有屏幕限制的设备。这个UI整齐的排在设备显示的顶部通过使用GTK+的堆栈，提供了良好的用户体验。其实现在自己的层，它明确开发者怎么能够实现他们自己的用户借口在由Yocto生成的镜像顶部。

3.你需要的和你怎么得到它们

你需要以下这些东西在Yocto环境下开发项目：

- * 一个主机，至少有50G空间，运行一个支持Yocto的linux发行版(Fedora, openSUSE, CentOS, Debian或者Ubuntu)。如果你的主机支持多核多线程，那么你可以修改Yocot的配置来加快编译速度。
- * 需要在你的系统上安装一些软件包。
- * Yocto项目的一个发行版

3.1linux发行版

Yocto项目组正在持续确认越来越多的linux发行版可以支持Yocto项目，你应该不会遇到问题：

- * Ubuntu
- * Fedora
- * openSUSE
- * CentOS
- * Debian

一份更详细的清单关于支持Yocto项目，你可以察看Yocto Project的“Distributions”章节。

JD.COM 京东

skyworth 创维



40"

JD.COM 京东

¥ 2799.00

6/6

评论排行

Qt5摄像头截图	(32)
移植qt5.3.1到arm	(24)
qt外部数据传入实现动态的折...	(19)
利用可变模板参数实现log功能	(18)
基于qt的多线程视频采集与传输	(17)
qt折线图显示数据	(17)
在Qt5的QML中使用QZXing...	(12)
qwt学习一	(12)
Qt和Zbar的结合使用识别二...	(11)
基于TCP协议的视频传输	(10)



最新评论

qwt学习一 Rolen_luoren : 我要把连接数据库的数据显示到曲线上,。请指教下呗
基于C++11的事件驱动框架 小乌龟在大乌龟背上 : 为什么不用vs开发,使用GDI+? 渲染使用OGL的话,不如直接撸一个游戏引擎了
QTabWidget+QToolBar实现浏览器标签... u010759651 : 可以用QAction触发信号,然后处理槽事件,在槽事件中用tabbar的位置来确定弹出菜单的位置。
qml+QZXing实现实时QRCode识别和给... zjq19930322 : 您好,我想问一下我也出现这个问题了bindTextureImage: clearing GL err...
QTabWidget+QToolBar实现浏览器标签... Yeah_tian : 楼主大大,请问解决点击好多下菜单才出现标签列表的问题了么?希望可以指点下
基于TCP协议的视频传输 漫9612 : 博主有没有不需要opencv的用QT+VS基于TCP协议的视频传输
Qt中unicode转utf-8 MoreBoldness : qDebug()<<QString::fromLocal8Bit("\u4F60\u597D");/...
基于TCP协议的视频传输 qq_38492405 : 给我留一份582591439@qq.com
基于TCP协议的视频传输 qq_38492405 : 给我留一份582591439@qq.com
基于TCP协议的视频传输 qq_38367446 : 651661297@qq.com谢谢

Test

JD.COM 京东

OpenEmbedded编译系统能够运行在任何版本,它们必须如下版本的git, tar和python程序。

- * Git 1.7.8或者更高版本
- * tar 1.24或者更高版本
- * Python 2.7.3或者更高版本,但是不包括Python3,它是不受支持的。

更早版本的Python是不能工作的,并且系统目前不支持python3。如果你的系统当前的环境不能满足上述的三个条件,那么你必须按步准备好你的系统,以便你能够使用编译系统。在手册中查看 [“Required Git, tar, and Python Version”](#) 章节。

3.2软件包

软件包和包的安装不同依赖于你的开发系统和你的意图。比如,如果你想构建一个能够在QEMU上以图形化运行的系统镜像(最小,基础编译要求),软件包的数量是和如果你想构建一个无图形的系统或者包含Yocto文档是不同的。总之,如果你想包含所有的方面,需要安装数量巨大的软件包。记得在安装的时候使用sudo命令。

下面的几小节的内容将告诉你,在支持的linux中,那些软件包需要被安装,如果你想编译一个能在QEMU上以图形化运行的系统镜像。
对于在其他情况下,需要的软件包列表,可以在Yocto手册的 [“Required Packages for the Host Development System”](#) 章节看到。

3.2.1 Ubuntu和Debian

在Ubuntu和Debian发行版中,你需要的必要和图形化的支持软件包包含在下面的命令中:

```
1 $ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential
```

Note

如果你的环境安装了 oss4-dev 软件包,你可能因为其安装在它的自定义目录/usr/include/linux/soundcard.h在Debian系统上,QEMU运行失败。如果你遇到了上述的问题,下面两个的任何一个可以帮你解决:

```
1 $ sudo apt-get build-dep qemu
2 $ sudo apt-get remove oss4-dev
```

3.2.2 Fedora

在Fedora发行版中,你需要的必要和图形化的支持软件包包含在下面的命令中:

```
1 $ sudo yum install gawk make wget tar bzip2 gzip python unzip perl patch diffutils diffutils-devel
```

3.2.3 OpenSUSE

在OpenSUSE发行版中,你需要的必要和图形化的支持软件包包含在下面的命令中:

```
1 $ sudo zypper install python gcc gcc-c++ git coreutils
```

3.2.4 CentOS

在CentOS发行版中,你需要的必要和图形化的支持软件包包含在下面的命令中:

```
1 $ sudo yum install gawk make wget tar bzip2 gzip python unzip perl patch diffutils diffutils-devel
```

JD.COM 京东

¥ 2799.00

6/6



Note
根据你使用的CentOS版本不同，可能还有其他的要求和依赖u。想要知道更多的细节，你可以查看[Poky/GettingStarted/Dependencies](#)维基的CentOS章节。

3.3 Yocto项目发行版

我们推荐你使用git clone操作来从**poky**git仓库克隆拷贝到本地系统。这样你可以为Yocto项目做出自己的贡献。
你也可以从Yocto项目的网站下载Yocto项目的发行版文件。在那个网站，你只需要点击位于导航栏的 *Downloads*，就会带你到一个列出了所有下载项的页面，当前的和以前的都可以下载。每日和开发构建版本被维护在 <http://autobuilder.yoctoproject.org/pub/nightly/> 。

4 运行一个测试的例子

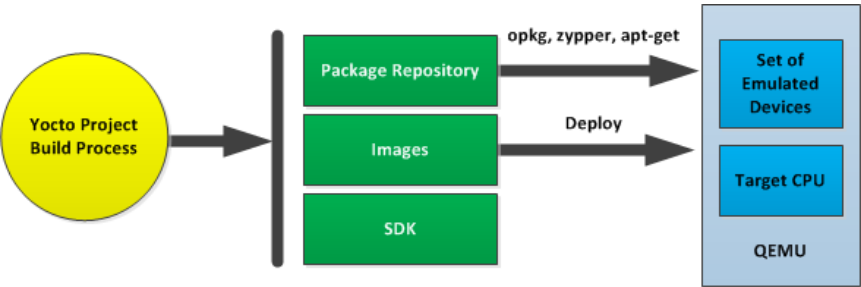
现在你已经有了一个满足要求的系统，你可以试一试Yocto项目了。这一节提供了一些你需要做的步骤：

- * 构建一个系统镜像，和在QEMU模拟器中运行。
- * 使用一个预先编译的镜像，和在QEMU模拟器中运行。

Note
这一节没有提供细节，而是最少提供，让你从命令和设计例子开始。知道更多细节，可以查看适当的手册在[Yocto Project manual set](#)中。

4.1 构建一个镜像

在开发环境中，你将必须编译镜像，一旦你改变了硬件支持，修改了系统库，或者增加改变了依赖的服务。



Building an Image

使用下列的命令来构建你的镜像。OpenEmbedded的构建过程会生成一个完整的linux系统，包括工具链，从源码。

Note about Network Proxies
默认情况下，编译过程使用一个预先定义好的顺序内而且没有设置网络代理，你可能在编译时抓取代如果你不知掉本地的代理设置，你咨询当地的网络发点也可以检查您的浏览器设置。最后，你可以找息在Yocto项目的手册FAQ上或者 [“Working Beh](#)

```
1 $ git clone http://git.yoctoproject.org/gi
2 $ cd poky
```





```
3 $ git checkout -b fido origin/fido
4 $ source oe-init-build-env
```

Tip

为了在编译过程节约磁盘空间，你可以在你项目的配置文件追加一句，配置文件位于 `poky/build/conf/local.conf`。加上这句后，一旦莫个包编译完成后，将删除编译这个包的工作目录。
`INHERIT += "rm_work"`

- 在前面的例子中，第一条命令使用Git来生成一个叫做poky的本地仓库，其是上游项目Yocto的poky仓库的一个克隆。
- 第三条命令将当前本地分支切换到Yocto发行的分支，使本地分支跟踪上游分支。以发行的分支建立你自己的分支可以确保你使用最新的代码。
- 最后一条命令运行Yocto项目的环境安装脚本-“oe-init-build-env”。运行这个脚本来设定OpenEmbedded编译需要的编译参数。运行了脚本之后，你当前的工作目录被设置成编译目录。之后，当编译组件的时候，Build目录将包含在编译过程中产生的文件。

Note

想要获得一个常驻内存BitBake更多的信息，可以查看 `oe-init-build-env-memres` 安装脚本。

花费一些时间来检查你项目配置目录下的 `local.conf` 文件，它位于 Build 目录。文件的默认内容可以保证项目的正常工作。然后，你可能对一些变量感兴趣。

默认情况下，编译的目标架构是qemux86，产生的镜像可以在QEMU模拟器上运行，它是Intel 32-bit架构。为了改变默认值，你可以修改配置文件中参数 MACHINE 的值，在你加载编译参数之前。

另外三个你可能感兴趣的参数是 `BB_NUMBER_THREADS`，`PARALLEL_MAKE` 和 `BB_NUMBER_PARSE_THREADS`。默认情况下，OpenEmbedded编译系统设置这三个参数的值是基于你主机的cpu的核心数。所以通常你不需要修改你 `local.conf` 中的参数来加快你的编译。
另一个在你编译之前需要考虑的是在你生成镜像时包管理。默认情况下，OpenEmbedded编译系统使用RPM包管理。你可以控制它通过 `PACKAGE_CLASSES` 参数。对于额外的包管理章节的信息，你可以在Yocto项目手册的 `package*.bbclass` 章节看到。

继续我们的命令来编译一个系统镜像，这个命令就是 `core-image-sato` 在例子中。对于参数-k，你可以使用 `bitbake --help` 命令来查看，也可以看Yocto项目手册的 BitBake 章节，或者 BitBake手册的 BitBake Command 章节。想获得更多的关于目标架构的信息，可以看Yocto项目手册的 images 章节。

```
1 $ bitbake -k core-image-sato
```

Note

BitBake需要python2.7.3环境。更多信息请查看 `python2.7.3` 章节。
使用QEMU模拟器运行这个镜像的最后一个命令：

```
1 $ runqemu qemux86
```

关闭





Note

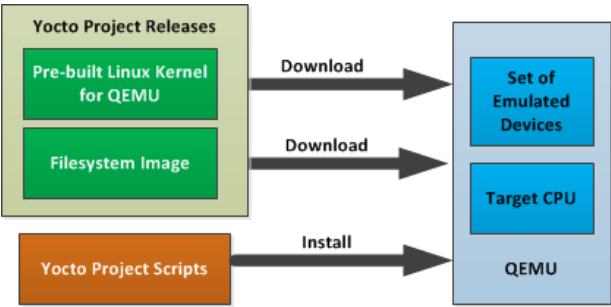
依赖于cpu核心数，内存容量，网络连接速度和其他的原因，编译过程可能花费几个小时。在第一次编译之后，其后的编译就会快很多。

如果你想学习更多关于运行QEMU，可以查看Yocto项目开发手册的 Using the Quick EMUlator(QEMU) 章节。

希望获得更多的关于使用预先编译二进制文件，继续阅读下面的内容。

4.2 使用预构建的二进制文件和QEMU

如果硬件，库和服务都是稳定的，你可以使用预构建的文件系统，内核和工具链在QEMU模拟器上来时使用它。这个对开发应用软件很有作用。



Using a Pre-Built Image

- 为了使用它，你需要做以下几件事：
- * 安装适当的独立工具压缩包。
 - * 下载QEMU能够启动的预编译的镜像。你需要确保能够得到适合你系统架构的QEMU镜像(比如arm, x86等等)。
 - * 下载适合架构的文件系统镜像。
 - * 设置硬件的模拟环境，然后启动QEMU模拟器。

4.2.1安装工具链

你能够从“ <http://downloads.yoctoproject.org/releases/yocto/yocto-1.8/toolchain/> ”下载一个压缩的安装包，其中包含了预编译的工具链，运行QEMU模拟器的脚本，帮助文件。适用于32位和64位开发系统的工具链在i686和x86_64文件目录下。Yocto提供的工具链基于 core-image-sato 镜像和包含了开发对应镜像的所需要的库。每一个开发的系统支持至少五种目标架构。

该文件安装程序脚本的名字的构成有如下的规则，主机系统开始，后紧跟一个代表着目标体系的字符串。

```
poky-glibc-host_system-image_type-arch-toolchain-release_version.sh
```

host_system 是开发系统的，比如 i686 或者 x86_64。

image_type 是你的镜像，你希望开发一个SDK给用户。

BitBake命令:

```
bitbake core-image-sato -c populate_sdk
```

arch 是目标体系架构，比如: i586, x86_64, powerpc, mips。

release_version 是发行号，比如 1.8 1.8+snapshot。

例如，下面的安装包是一个64位主机系统，为i586目标架构。

```
poky-glibc-x86_64-core-image-sato-i586-toolchain
```

工具链是独立的，默认安装在 /opt/poky 目录。或者，





录。

下面的命令展示了如何运行一个x86架构的64位主机系统用于32位x86架构的目标机的工具链压缩安装包。你必须赋予它可执行权限。

在例子中，我们假设安装包位于 `~/Downloads` 目录下。

Note

如果你对安装包所在目录没有写权限，安装包提示你这个问题和退出。请确保你此目录下具有写权限，然后才能执行它。

```
$ ~/Downloads/poky-glibc-x86_64-core-image-sato-i586-toolchain-1.8.sh
```

4.2.2 下载预编译linux内核

你可以从 <http://downloads.yoctoproject.org/releases/yocto/yocto-1.8/machines/qemu> 下载适用于QEMU模拟器运行的预编译内核。请确定你下载的内核和你想模拟的架构相匹配。一共有如下几种支持的机器架构: `qemuarm`, `qemumips`, `qemuppc`, `qemux86` 和 `qemux86-64`。

大多数内核文件有以下几种形式：

- `*zImage-qemuarch.bin`
- `vmlinux-qemuarch.bin`

这里，`arch` 是目标架构，比如 `x86`, `x86-64`, `ppc`, `mips`, `arm`。

你可以从“Yocto Project Kernel” 了解更多关于下载Yocto内核。

4.2.3 下载文件系统

你可以从 <http://downloads.yoctoproject.org/releases/yocto/yocto-1.8/machines/qemu> 下载适用相应架构的文件系统。请确定你适用的文件系统和你想模拟的架构相匹配。

文件系统镜像有两种压缩形式 `ext3` 和 `tar`。你必须使用 `ext3` 当你使用QEMU模拟器来引导一个镜像时。`tar` 形式的在主机开发系统中会被解压，在Yocto构建过程中使用。

- `core-image-profile-qemuarch.ext3`
- `core-image-profile-qemuarch.tar.bz2`

其中 `profile` 是对文件镜像的简介，比如 `lsb`, `lsb-dev`, `lsb-sdk`, `lsb-qt3`, `minimal`, `minimal-dev`, `sato`, `sato-dev`, `sato-sdk`。

更多的信息可以在“Images” 章节找到。

`arch` 是目标架构，比如 `x86`, `x86-64`, `ppc`, `mips`, `arm`。

4.2.4 设置环境和启动QEMU模拟器

在你启动QEMU模拟器之前，你需要设置仿真环境。运行如下的命令即可。

```
source /opt/poky/1.8/environment-setup-arch-poky-linux-if
```

其中，`arch` 是目标架构，比如 `x86`, `x86-64`, `ppc`, `mips`, `arm`。`if` 代表嵌入式程序的二进制接口，不是所有的脚本都包含 `if` 字符串。

下面的命令调用QEMU模拟器：

```
` $ runqemu qemuarch kernel-image filesystem-
```

`qemuarch` 是模拟目标的体系结构。

`kernel-image` 是内核镜像。

`filesystem-image` 是 `ext3` 的文件系统镜像。

继续我们的例子，下面的两天命令用来设置模拟器环境

件系统和预编译好的内核镜像都在你的home目录下面构。

关闭





```
1 $ cd $HOME
2 $ source /opt/poky/1.8/environment-setup-i586-poky-linux
3 $ runqemu qemu86 bzImage-qemu86.bin \
4 core-image-sato-qemu86.ext3
```

QEMU加载的环境变化依赖于文件系统镜像和目标架构。例如，如果你加载了ARM目标架构的环境，然后启动最小的QEMU镜像，模拟器重新启动一个命令行模式的shell终端。然而，如果你启动一个SDK镜像，那么QEMU启动一个GUI。

Note

启动PPC镜像导致QENU加载命令行模式的Shell终端。

如果想得到更多的关于使用Yocto进行镜像和应用开发，最好的方式是继续查阅Yocto的开发手册。

5.超级用户

本节将重头开始给你讲述一个最精简的说明对于Beaglebone硬件使用Yocto构建镜像。在64位的ubuntu12.04上，只需要4个核心步骤。

5.1获得Yocto项目

设置你的源码目录使用git clone操作在poky仓库，然后切换到当前的发布分支。

```
1 $ cd ~
2 $ git clone git://git.yoctoproject.org/poky
3 $ cd poky
4 $ git checkout -b fido origin/fido
```

5.2设置你本地环境

在开发过程中，你的工作依赖于其他一些软件包，去“ The Packages” 章节查阅相关内容。

5.3初始化构建环境

来到你的Yocto源码目录的根目录，初始化你的环境并制定一个存放构建内容的目录。

```
1 $ source oe-init-build-env mybuilds
```

此处，mybuilds 目录已经被生成，现在其在你当前的工作目录下。如果你没有提供，那么其默认为 build ，位于你的源码目录。

5.4配置local.conf

初始化构建环境，过程中会生成 conf/local.conf 文件，其位于你的构建目录。你需要手动修改这个文件来适应特殊的机器。

```
1 MACHINE ?= "beaglebone"
```

怎么来进行Yocto项目的构建，一个最好的处理方法就步骤可以帮助你更好的收集源码文件。比如，你可以提前取代代码直到开始你的构建。如果你想获得更多的Working with Source Files 章节。

关闭





5.5构建镜像

在这里，你需要为你的 Beaglebone 硬件选择一个需要编译的镜像。如果这是你使用Yocto第一次构建，你应该尝试最小最简单的镜像:

```
1 | $ bitbake core-image-minimal
```

现在，你可以看看书，喝咖啡，等构建完成。
默认情况下，BitBake中断如果在构建过程中遇到一个错误。如果你想即使遇到了错误，你还是想构建继续，可以使用:

```
1 | $ bitbake -k core-image-minimal
```

一旦你构建出了你的镜像，你可以按部就班的加载引导它在你的硬件上。一般情况下，你可以通过查阅 BitBake User Manual 来更多的了解BitBake的使用。


顶 1
踩 0

- 上一篇 docker和nodejs
- 下一篇 使用express创建自己第一个nodejs程序

相关文章推荐

- core-image-minimal分析1
- MySQL在微信支付下的高可用运营--莫晓东
- 使用 Yocto Project 构建自定义嵌入式 Linux 发行...
- 容器技术在58同城的实践--姚远
- yocto配置好eclipse之后编译多线程配置
- SDCC 2017之容器技术实战线上峰会
- imx6q 编译下载指南
- SDCC 2017之数据库技术实战线上峰会
- Yocto tips (16): Yocto 制作SDK分发Toolchain ...
- 腾讯云容器服务架构实现介绍--董晓杰
- 等待队列和内核线程 在触摸屏代码中的应用
- 微博热点事件背后的数据库运维心得--张冬洪
- OpenEmbedded 中文手册
- OpenEmbedded 中文手册
- Freescale Yocto Project搭建
- Yocto 项目快速入门

我在附近，30岁，找成熟男士



看我照片，同城约，距离600米,附近男士联系我

SoftNoze TEL:021-64459600



机械视觉传感器附件供应商,品质保

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

关闭

JD.COM 京东

skyworth 创维



¥ 2799.00

6/6



系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

9-2017, CSDN.NET, All Rights Reserved



关闭

