

通过buildroot解决需要移植QT库的问题！！！！步骤详细，需要的拿走

原创

置顶

More Twinkle

于 2019-12-09 14:20:54 发布

8491

收藏

64

文章标签：


qt移植

buildroot编译生成带QT库的文件系统

arm开发板

版权

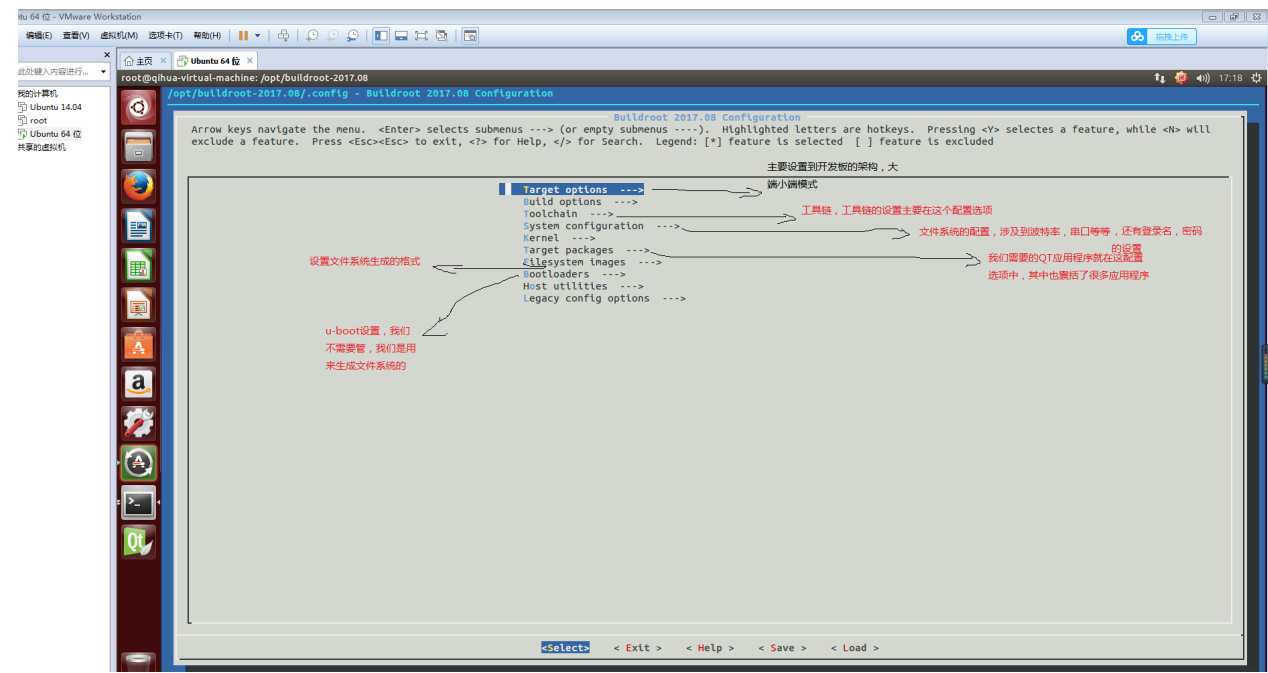
一， buildroot下载
buildroot官网
我下载的是最后的

 buildroot-2019.02.6.tar.gz	2019/10/8 8:59	WinRAR 压缩文件	6,195 KB
--	----------------	-------------	----------

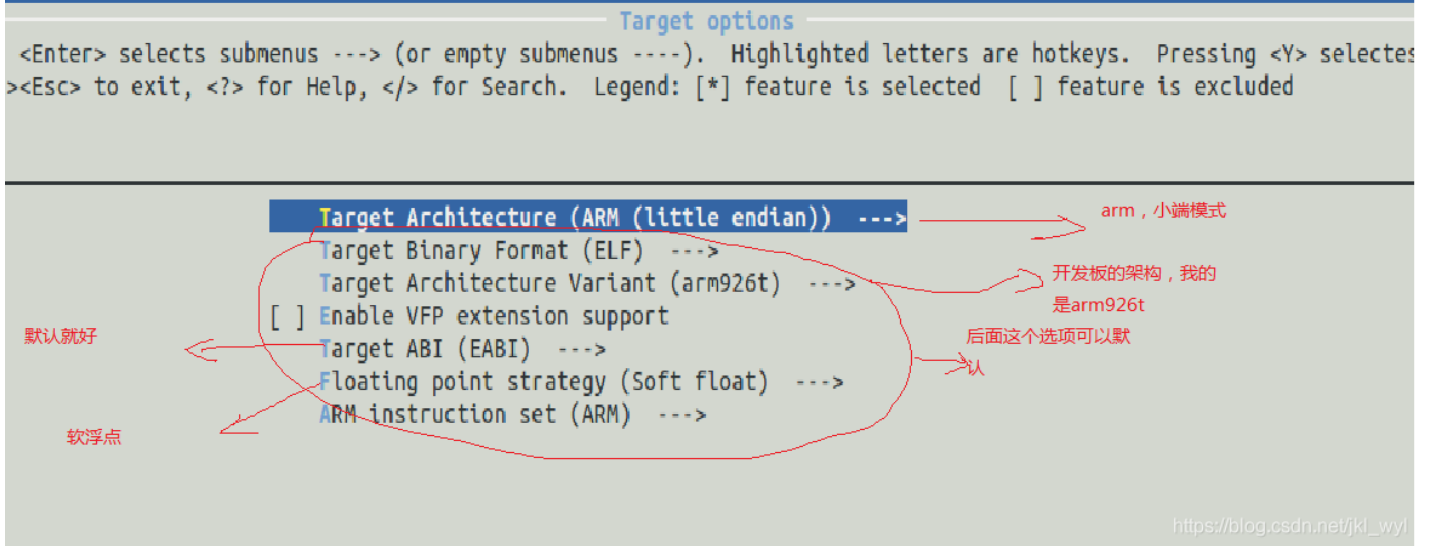
这个建议下载最新版本，版本越新，支持的应用库就越多。

二， 安装 交叉编译 器
编译器的安装网上面有很多资源，可以自行搜索。
链接：<https://pan.baidu.com/s/1jOpLPW5mOya0sKcFVrmn-Q>
提取码：dug8
我是用的是4.9版本的编译器，需要的可以下载。

三， 配置buildroot
在配置buildroot之前，先配置好交叉编译工具。
1.进入buildroot目录
cd buildroot-2017.08/
make menuconfig
会出现下面这个界面

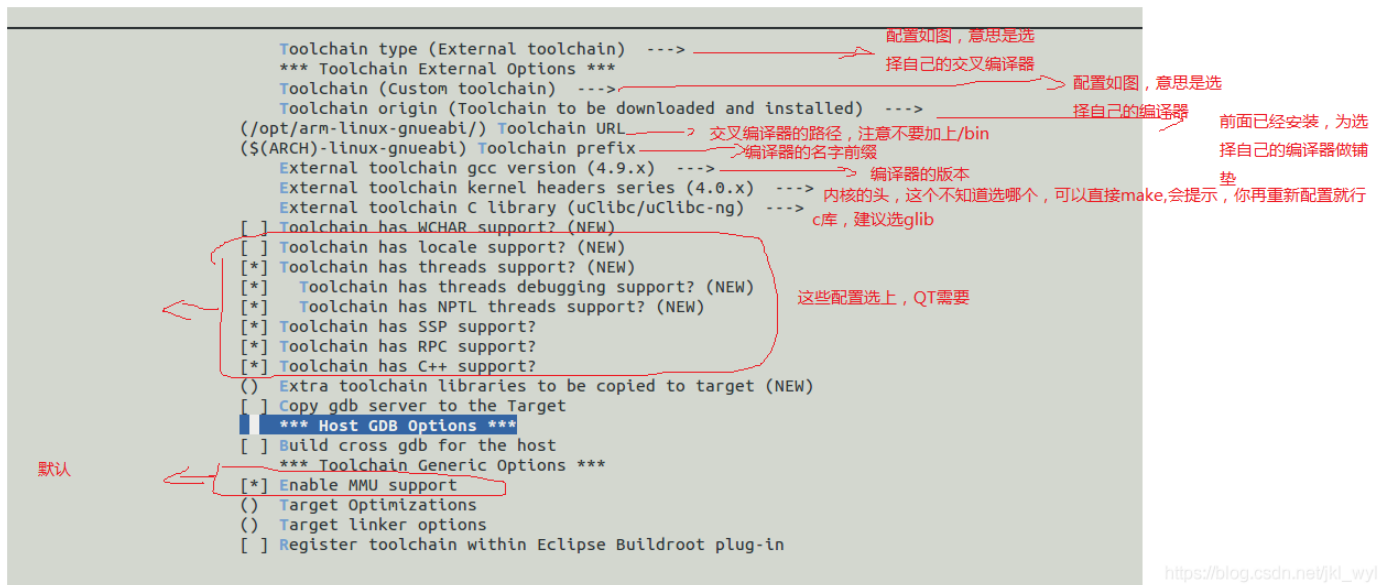


2.target option

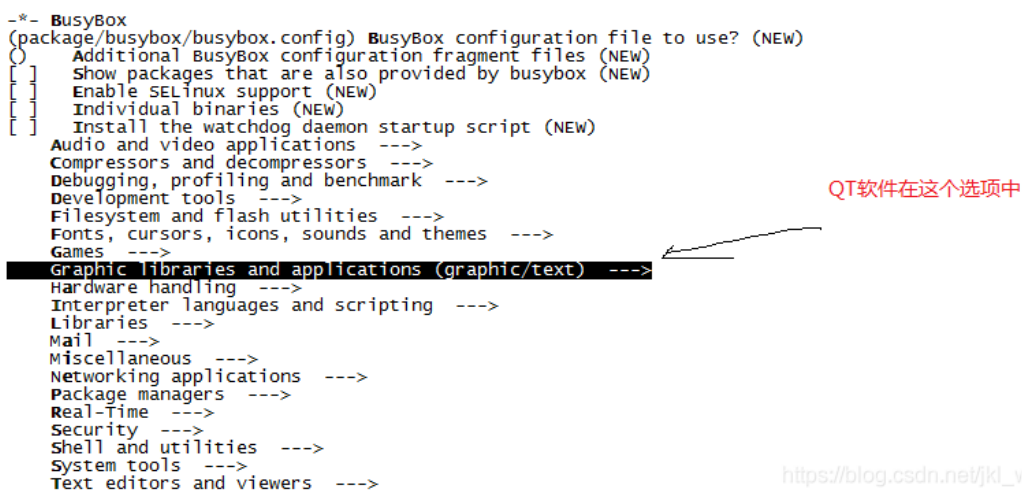


配置如图中所示:

3.toolchain



4.Target packages



```

*** Graphic applications ***
[ ] fswebcam (NEW)
[ ] ghostscript (NEW)
*** glmark2 needs a toolchain w/ C++, gcc >= 4.9 ***
[ ] gnuplot (NEW)
[ ] ihead (NEW)
[ ] libva-utils (NEW)
[ ] pngquant (NEW)
*** qt5cinex needs an open(EG)GL-capable backend ***
[ ] rrdtool (NEW)
[ ] tesseract-ocr (NEW) ----
*** Graphic libraries ***
[ ] cegui06 (NEW)
[ ] directfb (NEW)
[ ] efl (NEW)
[ ] fbdump (Framebuffer Capture Tool) (NEW)
[ ] fbgrab (NEW)
[ ] fb-test-app (NEW)
[ ] fbterm (NEW)
[ ] fbv (NEW)
[ ] freerdp (NEW)
[ ] imagemagick (NEW)
*** linux-fusion needs a Linux kernel to be built ***
[ ] mesa3d (NEW) ----
[ ] ocrad (NEW)
[ ] psplash (NEW)
[ ] SDL (NEW)
[ ] sdl2 (NEW)
*** Other GUIs ***
[ ] Qt (NEW) ----
[*] Qt5 ---->
*** QT libraries and helper libraries ***
[ ] grantlee (NEW)
[ ] qextserialport (NEW)
[ ] qjson (NEW)
[ ] quazip (NEW)
[ ] qwt (NEW)
*** tekui needs a Lua interpreter and a toolchain w/ threads, dynamic library ***
*** weston needs udev and a toolchain w/ locale, threads, dynamic library, headers >= 3.0 ***
[ ] X.org X Window System (NEW) ----
*** midori needs libgtk3 and a glibc toolchain w/ C++, gcc >= 4.9, host gcc >= 4.8 ***
[ ] qt-webkit-kiosk (NEW)
[ ] xkeyboard-config (NEW)

```

选中QT5,进入其中配置qt相关内容

注意 “一般情况下, 照着我的配置进行配置, 不会有问题!!!!”

```

-- Qt5
Qt5 version (LTS (5.6)) ---->
*** qt53d module needs an OpenGL-capable backend ***
qt5base
() Custom configuration options (NEW)
() Config file (NEW)
[ ] Compile and install examples (with code) (NEW)
[ ] Concurrent module (NEW)
[*] MySQL Plugin
[ ] PostgreSQL Plugin (NEW)
[ ] SQLite 3 support (No sqlite support) ---->
[*] gui module
[*] widgets module
*** OpenGL support needs an OpenGL-capable backend ***
linuxfb support
*** directfb backend available if directfb is enabled ***
*** X.org XCB backend available if X.org is enabled ***
*** eglfs backend available if OpenGL and EGL are enabled ***
() Default graphical platform (NEW)
[*] fontconfig support
[*] GIF support
[*] JPEG support
[*] PNG support
[*] DBus module (NEW)
Enable ICU support (NEW)
Enable tslib support
*** qt5canvas3d needs an OpenGL-capable backend ***
*** qt5connectivity needs neard and/or bluez(5)_utils ***
[ ] qt5declarative (NEW)
[ ] qt5enginio (NEW)
*** qt5graphicaleffects needs an OpenGL-capable backend ***
[ ] qt5imageformats (NEW)
[ ] qt5location (NEW)
[ ] qt5multimedia (NEW)
*** qt5quickcontrols needs an OpenGL-capable backend ***
*** qt5quickcontrols2 needs an OpenGL-capable backend ***
[ ] qt5script (NEW)
[ ] qt5sensors (NEW)
[ ] qt5serialbus (NEW)
[ ] qt5serialport (NEW)
[ ] qt5svg (NEW)
[ ] qt5tools (NEW)
*** qt5virtualkeyboard needs at least qt-5.7 and an OpenGL backend ***
[ ] qt5webchannel (NEW)
[ ] qt5webkit (NEW)
[ ] qt5websockets (NEW)
[ ] qt5xmlpatterns (NEW)

```

选定QT版本, 我这里选定的是5.6

MySQL的插件, 需要用到

的就选上, 也就是数据库

gui界面

widgets 界面需要

支持显示屏

字体, 建议全选中

支持的图片格式

5. Filesystem images

```

[ ] axfs root filesystem (NEW)
[ ] cloop root filesystem for the target device (NEW)
[ ] cpio the root filesystem (for use as an initial RAM filesystem) (NEW)
[ ] cramfs root filesystem (NEW)
[*] ext2/3/4 root filesystem
(0) ext2/3/4 variant (ext4) ---->
(60M) filesystem label (NEW)
(0) exact size (NEW)
(0) exact number of inodes (leave at 0 for auto calculation) (NEW)
(5) reserved blocks percentage (NEW)
(-0 ^64bit) additional mke2fs options (NEW)
Compression method (no compression) ---->
*** initramfs needs a Linux kernel to be built ***
[ ] jffs2 root filesystem (NEW)
[ ] romfs root filesystem (NEW)
[ ] squashfs root filesystem (NEW)
[*] tar the root filesystem (NEW)
[*] compression method (gzip) ---->
() other random options to pass to tar (NEW)
[ ] ubifs root filesystem (NEW)
[ ] yaffs2 root filesystem (NEW)

```

配置生成文件系统的格式, 一般选择ext4, 看个人需要

指定压缩方式

三, 编译文件系统

make

编译完成后，在output目录中有一下目录：

```
root@ubuntu:~/work/f1c100/buildroot-2017.08_QT4.8.6_4.8/output# ls
build host images staging target
```

编译完成后，在buildroot-2017.08/output/image中会生成rootfs.tar以及你所设置需要生成的文件系统格式。在buildroot-2017.08/output/target中就是你生成的文件系统。

想要知道是否生成了QT库，进入buildroot-2017.08/output/target/usr/lib目录中，会有QT库，我生成的下图：

```
root@ubuntu:~/work/f1c100/buildroot-2017.08_QT4.8.6_4.8/output# cd target/usr/lib
root@ubuntu:~/work/f1c100/buildroot-2017.08_QT4.8.6_4.8/output/target/usr/lib# ls
fonts          libform.so.6.0  libjpeg.so.9.2.0  libmysqlclient_r.so.16.0.0  libncurses.so.6.0  libpng16.so.16.31.0  libqtgui.so.4.8  libqtscript.so.4.8  libqtxml.so.4.8  libstdc++.so.6.0.19  libz.so.1.2.11
libatomic.so.1  libhistory.so   libmenu.so        libmysqlclient.so.16       libpanel.so        libpng.so           libqtgui.so.4.8.7  libqtscript.so.4.8.7  libqtxml.so.4.8.7  libstdc++.so.6.0.19  libz.so.1.2.11
libatomic.so.1.0.0  libhistory.so.7  libmenu.so.6      libmysqlclient.so.16       libpanel.so.6      libpng.so           libqtgui.so.4.8.7  libqtscript.so.4.8.7  libqtxml.so.4.8.7  libstdc++.so.6.0.19  libz.so.1.2.11
libcurses.so      libhistory.so.7.0  libmenu.so.6.0    libmysqlclient.so.16.0.0   libpanel.so.6.0    libpng.so           libqtgui.so.4.8.7  libqtscript.so.4.8.7  libqtxml.so.4.8.7  libstdc++.so.6.0.19  libz.so.1.2.11
libform.so        libjpeg.so       libmysqlclient_r.so  libncurses.so             libpanel.so.6.0    libpng16.so.16     libqtcore.so.4.8.7  libqtnetwork.so.4.8.7  libqtsql.so.4.8.7  libstdc++.so.6.0.19  libz.so.1.2.11
libform.so.6      libjpeg.so.9     libmysqlclient_r.so.16  libncurses.so.6          libpanel.so.6.0    libpng16.so.16     libqtcore.so.4.8.7  libqtnetwork.so.4.8.7  libqtsql.so.4.8.7  libstdc++.so.6.0.19  libz.so.1.2.11
root@ubuntu:~/work/f1c100/buildroot-2017.08_QT4.8.6_4.8/output/target/usr/lib#
```

到这一步，带QT库的文件系统编译完成了，就缺最后的挂载文件系统。

四，挂载文件系统

我是使用的是通过SD卡挂载文件系统。每个开发板的挂载可能会有不同，我的挂载如下：

方法：通过在ubuntu下的fdisk命令对SD卡进行分区。

例如：

4.1 fdisk /dev/sdb

n 1 默认 +32M

n 2 默认 默认

完成两个分区。

w 退出分区操作

4.2 mkfs.vfat /dev/sdb1 //格式化第一个分区为fat16格式

mkfs.ext4 /dev/sdb2 //格式化第一个分区为ext4格式

注意：在进行分区和格式化操作的时候，不要挂载分区，如果分区自动挂载了，手动卸载，例如：umount /dev/sdb1,否则会出错。

4.3 存入数据

mount /dev/sdb1 /media/usb0 //挂载分区1

mount /dev/sdb12 /media/usb1 //挂载分区2

然后将设备树文件和内核镜像文件复制到第一个分区。

将 文件系统的压缩包解压到第二个分区。

4.4，将TF卡插入开发板，进入u-boot界面

设置启动参数：

setenv bootargs 'console=ttyS0,115200 panic=5 rootwait root=/dev/mmcblk0p2 rw rootfstype=ext4'

重启开发板，带QT的文件系统挂载完成，可以自己写QT程序进行测试，也可以使用自带的测试代码进行测试。

五，测试

在buildroot-2017.08_QT4.8.6_4.8/output/build/qt-4.8.7/examples/widgets/analogclock目录下有已经编译好的可执行文件analogclock。

直接运行：

./analogclock。

会出新报错，找不到eglfs的plugin,这是因为编译成立的文件系统默认的plugin是eglfs,这时候需要我们手动设置环境变量

export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0 //指定我们的显示屏

再次./analogclock，开发板的显示屏上会出现一个图案。到这一步，整个过程就完成了，希望能够帮到有需要的人