

Tina

交叉编译器介绍 v1.2

文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2016/8/19		初始版本
V1.1	2016/10/28		第二次 review
V1.2	2018/4/17		更新工具链信息

Yllwinnertec

目 录

1. 概述.....	4
1.1. 运行环境.....	4
1.2. 交叉编译器版本号.....	4
2. 目录结构.....	5
3. 使用例子.....	9
4. 编译交叉工具链.....	10
5. Declaration.....	12

Allwinner

1. 概述

文档主要介绍 Tina 使用的交叉编译器。

1.1. 运行环境

Tina 默认使用的交叉编译器需要运行在 Linux 64bit 的系统上（建议 ubuntu 14.04 64bit）。

如果由于 PC 的系统、CPU 架构等原因，编译器等可执行文件无法正常运行，则可以参考参考章节 4，重新编译工具链。

1.2. 交叉编译器版本号

Tina 交叉编译器的版本号为：

Linaro GCC 6.4-2017.11

musl 1.1.16

glibc 2.23

binutils 2.27

gdb 8.0

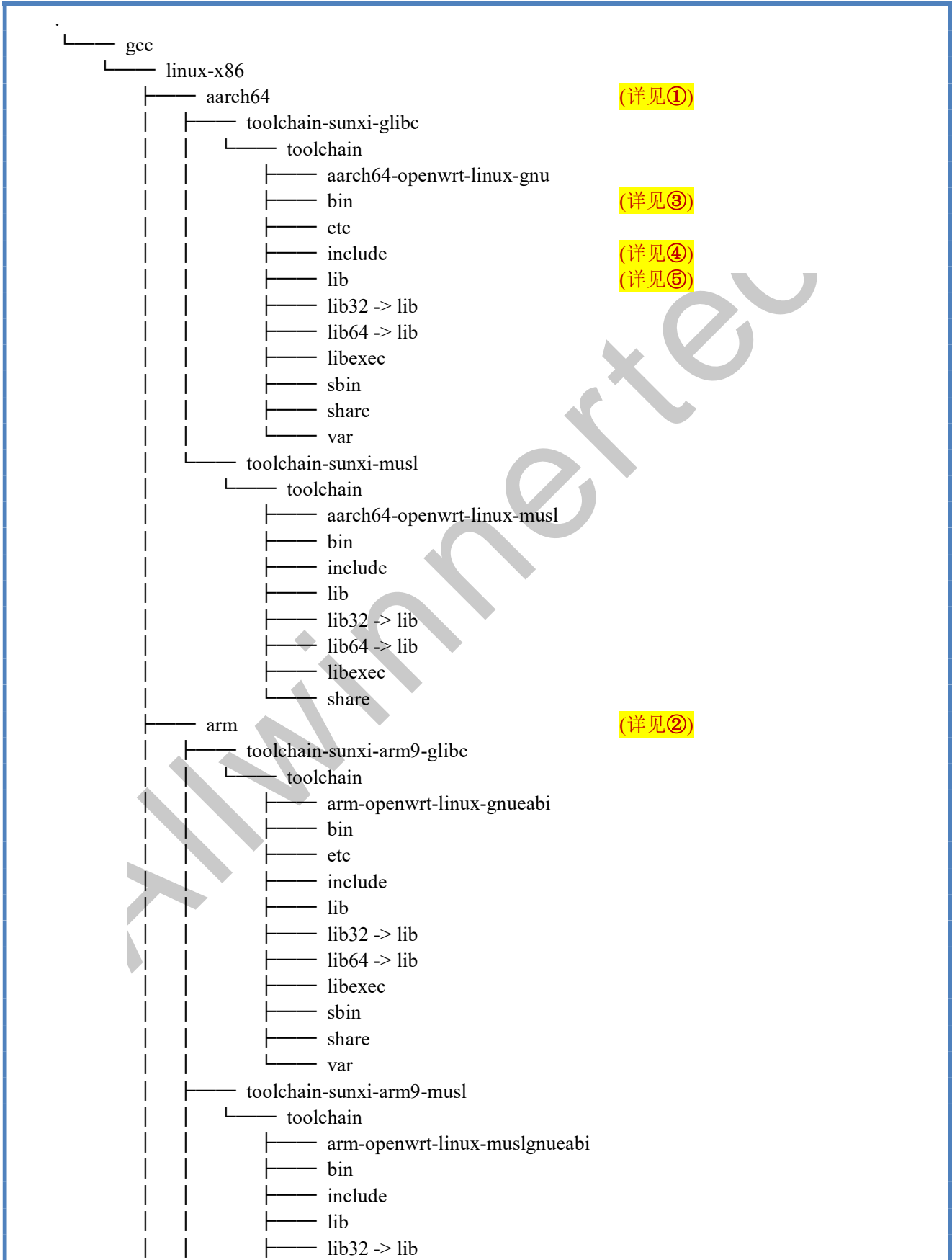
Allwinner

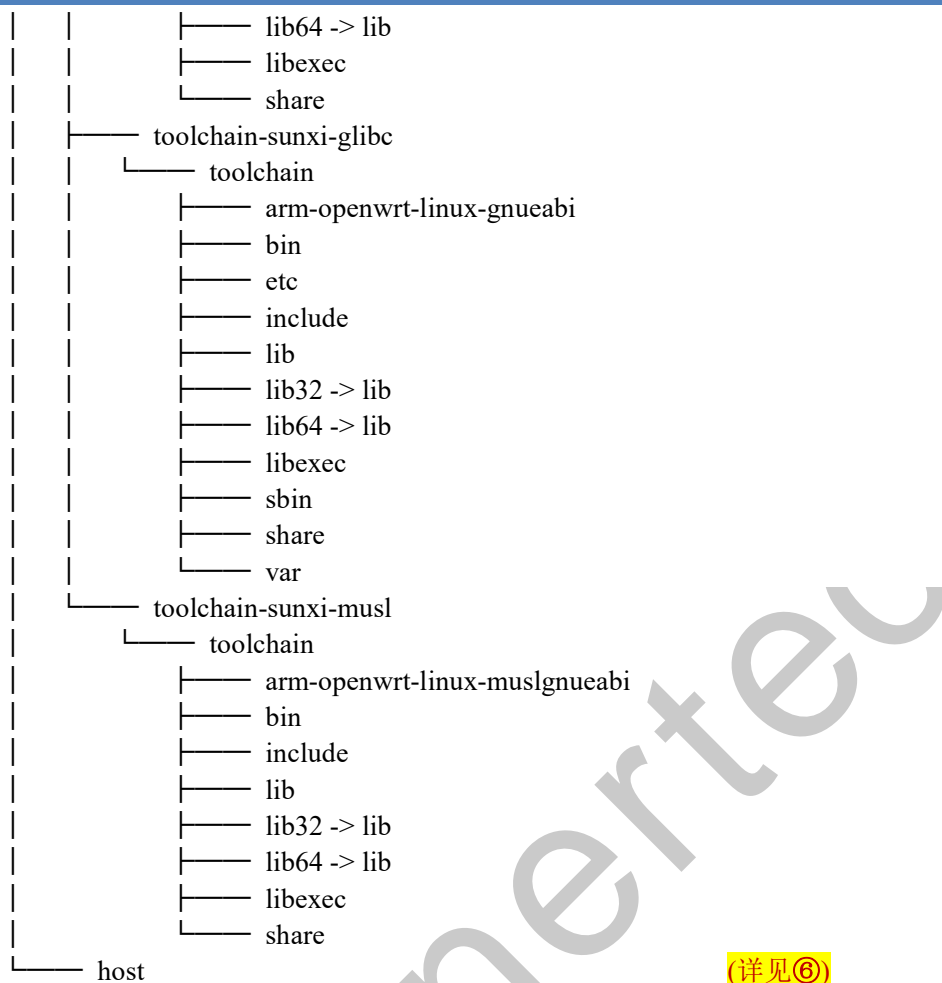
2. 目录结构

Tina 交叉编译器是预先编译好的，目录为：

tina/prebuilt/gcc

具体目录结构如下：





详注:

- ① gcc/linux-x86/aarch64 是小机为 64 位的编译器目录;
- ② gcc/linux-x86/arm 是小机为 32 位的编译器目录 (R40 是 32 位机器);
- ③ bin 目录: 存放交叉编译工具, 包括 gcc, g++, ar, nm, objdump 等。
 gcc: C 交叉编译器;
 g++: C++交叉编译器;
 ar: 创建静态链接库;
 ld: 连接器, 将目标文件链接为可执行程序;
 objdump: 查看目标文件或者执行文件的构成;
 nm: 列出目标文件的符号;
- ④ include 目录: 存放编译应用程序时程序包含的头文件, 如 stdio.h 等
- ⑤ lib 目录: 存放编译器相关的库, 如 libc 库
- ⑥ host 目录: 存放 host 端编译器相关目录 (暂时未使用, 目前 host 端使用的是本地 gcc 工具等)

下面列出各个平台下所使用工具链的路径:

平台	C 库	路径
R6	musl	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-arm9-musl
R6	glibc	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-arm9-glibc
R11	musl	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-musl
R11	glibc	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-glibc
R16	musl	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-musl
R16	glibc	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-glibc
R18	musl	tina/prebuilt/gcc/linux-x86/aarch64/toolchain-sunxi-musl
R18	glibc	tina/prebuilt/gcc/linux-x86/aarch64/toolchain-sunxi-glibc
R30	musl	tina/prebuilt/gcc/linux-x86/aarch64/toolchain-sunxi-musl

R30	glibc	tina/prebuilt/gcc/linux-x86/aarch64/toolchain-sunxi-glibc
R40	musl	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-musl
R40	glibc	tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-glibc

Tina 平台支持 musl, glibc 两种 C 库，一般来说默认配置下均使用 musl 作为 C 库，如果想更换 C 库，可执行 make menuconfig 进行配置

配置路径：

```
Advanced configuration options (for developers)
└─> Select external toolchain C library (Use musl) --->
```

操作图示：

```
.config - Tina Configuration
> Advanced configuration options (for developers)
  Advanced configuration options (for developers)
  Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlig
  hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to e
  </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

  --- Advanced configuration options (for developers)
  [ ] Show broken platforms / packages
  ( ) Local mirror for source packages
  [*] Automatic rebuild of packages
  ( ) Build suffix to append to the target BUILD_DIR variable
  ( ) Override the default TARGET_ROOTFS_DIR variable
  [ ] Use ccache
  [ ] Enable log files during build process (NEW)
  [ ] Enable package source tree override (NEW)
  (-fno-caller-saves -fno-plt) Additional compiler options
  [ ] Target Options (NEW) ----
  [ ] Update Toolchain
  [ ] Select external toolchain C library (Use musl) --->
```

图 2-1 menuconfig

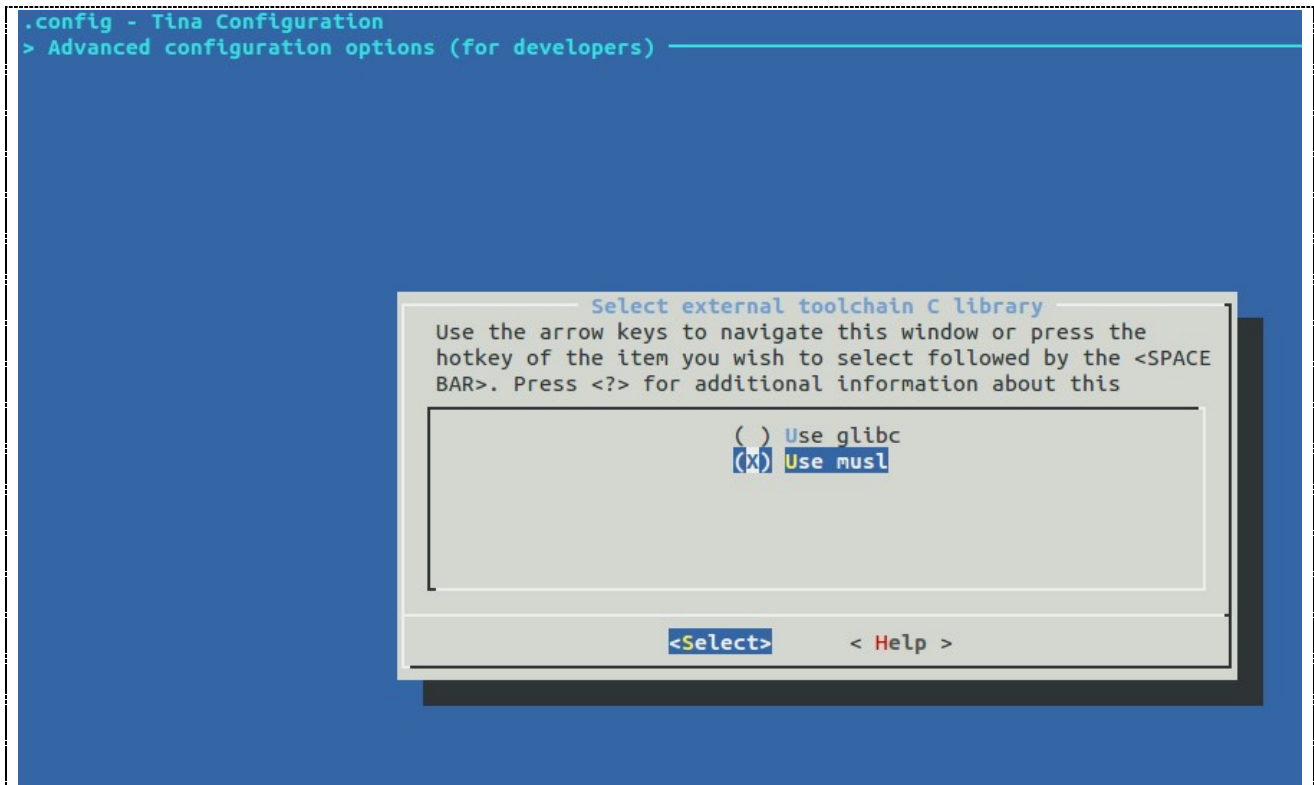


图 2-2 选择 C 库

3. 使用例子

如果想要在 Tina 源码上添加自己的应用，可以参考《Tina SDK Quick Start Guide》中的“创建和移植软件包”章节，这种方法不需要对编译器的路径、编译选项等进行设置，同时也方便开发者对软件包进行管理，建议使用该方法。

如果需要在 Tina 源码之外单独编译应用，则需要设置工具链相关路径，下面是编译一个 helloworld 的例子：

目录结构：

```
.
├── helloworld
├── main.c
└── Makefile
```

main.c:

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

Makefile:

```
CROSS_COMPILE_PATH=tina/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-musl/toolchain/bin
CROSS_COMPILE=$(CROSS_COMPILE_PATH)/arm-openwrt-linux-
CC=$(CROSS_COMPILE)gcc

all:
    $(CC) -o helloworld main.c
```

CROSS_COMPILE_PATH 需要写自己的编译器路径。

在当前目录下执行 make 即可进行编译，编译成功会生成可执行程序 helloworld，将它拷贝到小机端后即可运行。

4. 编译交叉工具链

Tina 中包含有 GCC, C 库等源码, 配置后可以重新编译(耗时较长), 并且在编译其它模块时均会使用新的编译器及 C 库。

在命令行中进入 Tina 根目录, 执行命令进入配置主界面:

```
source build/envsetup.sh (详见①)
lunch (详见②)
make kernel_menuconfig (详见③)
```

详注:

- ① 加载环境变量及 tina 提供的命令
- ② 输入编号, 选择方案, 例如 R40 对应方案为: azalea_m2ultra-tina
- ③ 进入内核配置主界面(对一个 shell 而言, 前两个命令只需要执行一次)

配置路径:

```
Advanced configuration options (for developers)
└─> Update Toolchain
```

操作图示:

```
Target System (azalea-m2ultra) --->
Target Profile (tina) --->
Target Images --->
Global build settings --->
[*] Advanced configuration options (for developers)
[ ] Build the OpenWrt Image Builder
[ ] Build the OpenWrt SDK
[ ] Package the OpenWrt-based Toolchain
[ ] Image configuration --->
Package features --->
Base system --->
Allwinner --->
```

图 4-1 make menuconfig 主界面

```
--- Advanced configuration options (for developers)
[ ] Show broken platforms / packages (NEW)
() Local mirror for source packages (NEW)
[*] Automatic rebuild of packages (NEW)
() Build suffix to append to the target BUILD_SUFFIX
() Override the default TARGET_ROOTFS_DIR variable
[ ] Use ccache (NEW)
[ ] Enable log files during build process (NEW)
[ ] Enable package source tree override (NEW)
(-fno-caller-saves -fno-plt) Additional compiler options
[ ] Target Options (NEW) ----
[*] Update Toolchain
```

图 4-2 Advanced configuration options 界面

在 Tina 根目录上执行编译 Tina 命令

```
$ make
```

注意:

1. 编译器编译过程中有依赖关系, **不建议使用并行编译**, 例如 `make -j8`;
2. 交叉编译器成功编译后, 会继续进行 **Tina 固件的编译**

编译完成后, 编译器相关目录为:

```
tina/out/azalea-m2ultra/staging_dir/toolchain
```

Yllwinner

