



LiveSuit 工具 使用指南

版本号：3.0.8

发布时间：2023-12-08

版本历史

版本	日期	责任人	版本描述
3.0.8	2023-12-08	AWA1746	增加 auto fel 功能描述。
3.0.7	2022-08-01	AWA1695	更改部分描述。
1.2	2021-12-20	AWA1695	1.增加 2.2-2.4 章节
1.1	2021-01-15	AWA1695	1.增加第 1 章。 2.更新文档模板和排版。
1.0	2020-08-04	KPA0398	创建文档。



目录

版本历史	i
目录	ii
图片目录	iii
1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
1.4 文档约定	1
1.4.1 标志说明	1
2 概述	2
2.1 安装指南	2
2.2 驱动 ko 模块安装	3
2.3 auto fel 说明	5
2.4 驱动 deb 包编译及安装	6
2.4.1 驱动 deb 包编译	6
2.4.2 驱动 deb 包安装	7
2.5 注意事项	8
2.6 Livesuit in Ubuntu20.04	8
2.6.1 驱动安装失败	8
2.6.1.1 驱动安装失败	8
2.6.1.2 驱动安装无权限或找不到	9
2.6.2 编译驱动失败	9
2.6.3 执行提示	10

图片目录

图 2-1	USB 驱动包	3
图 2-2	压缩包内容示意图	3
图 2-3	驱动安装示意图 1	3
图 2-4	驱动安装示意图 2	3
图 2-5	Makefile 修改内容示意图	4
图 2-6	驱动安装示意图 3	4
图 2-7	驱动安装示意图 4	4
图 2-8	启动工具示意图	5
图 2-9	awdev build already 示意图	6
图 2-10	dh-make 模块安装示意图	6
图 2-11	deb 包编译示意图	7
图 2-12	awdev 驱动安装示意图	7

1 前言

1.1 文档简介

本文档介绍了 LiveSuit 工具的安装指南。

1.2 目标读者

LiveSuit 工具的使用者。





1.3 适用范围

Linux 平台。

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

2 概述

Linux 版本 LiveSuit 支持的发行版本包括 Ubuntu, Fedora, Redhat, CentOS 几个常见的发行版本, 由于目前驱动未能很好支持 3.0 版本以上内核, 建议安装内核版本为 2.6.x 开头的 Linux 发行版本。

2.1 安装指南

1. 二进制可执行文件为 LiveSuit.run。如果该文件没有执行权限, 请在终端下使用命令:

`chmod +x LiveSuit.run` 为该文件添加可执行权限。

2. 打开终端, 输入 `sudo ./LiveSuit.run` 来运行安装程序。

如果提示缺少 dkms 模块, Ubuntu 用户请使用命令: `sudo apt-get install dkms` 安装, 其他发行版本用户请使用 `yum install` 或者网络下载对应自己版本的 dkms 安装。

3. 程序安装在当前用户 Home 目录的 Bin 目录下面。

4. 安装完程序, 进入程序目录输入命令 `./LiveSuit` 运行 LiveSuit。



说明

1. Ubuntu 10.04 版本可以按照如下方法添加 udev 规则实现无需 root 权限读写硬件设备:

(1) 打开终端输入 `sudo vim /etc/udev/rules.d/10-local.rules`

(2) 输入以下 udev 规则内容, 其中切记将 test 替换为当前用户所在用户组。

`SUBSYSTEM!="usb_device",ACTION!="add",GOTO="objdev_rules_end"`

`#USBasp`

`SYSFS{idVendor}=="1f3a",SYSFS{idProduct}=="efe8",GROUP="test",MODE="0666"`

`LABEL="objdev_rules_end"`

(3) 保存并重启机器或者重启 udev 服务即可以普通用户运行 LiveSuit。重启 udev 服务命令为:
`sudo service udev restart`

2. Ubuntu 12.04 的 udev 规则关键字请将 SYSFS 替换为 ATTRS。



注意

其他发行版本用户请查阅 udev 相关资料并以 Ubuntu 10.04 添加 udev 规则为例, 自行添加 udev 规则。没有添加 udev 规则的用户需要以 `sudo ./LiveSuit` 命令运行程序。

5. 刷固件过程中, 请勿关闭程序或断开设备连接, 以免造成硬件损坏。

2.2 驱动 ko 模块安装



注意

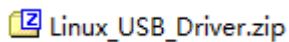
安装前的环境准备工作:

sudo dpkg -l 查看软件安装情况, 确认安装了 dkms、flex、bison, 如果缺少 xxx 库, 请执行 sudo apt-get update; sudo apt-get install xxx

安装步骤:

1. 请联系我司 FAE 获取 Linux 系统 USB 驱动包 Linux_USB_Driver.zip。

图 2-1 USB 驱动包



2. 解压压缩包, 得到如图 2-2 所示的文件

图 2-2 压缩包内容示意图

名称	修改日期	类型	大小
awusb.c	2020/8/4 19:35	C 源文件	17 KB
dkms.conf	2020/8/4 19:35	CONF 文件	1 KB
Makefile	2021/4/23 18:28	文件	1 KB
ReadMe.txt	2022/7/26 9:06	文本文档	1 KB

3. 执行 sudo dkms remove awusb/0.5 清除原有驱动模块, 可能会出现如图 2-3 所示情况, 无需理会。

图 2-3 驱动安装示意图 1

```
allwinnertech@allwinnertech-All-Series:/usr/src$ sudo dkms remove awusb/0.5
[sudo] allwinnertech 的密码:
Error! The module/version combo: awusb-0.5
is not located in the DKMS tree.
```

4. 执行 sudo dkms add awdev/0.5, 如果出现如图 2-4 错误提示, 请在 /usr/src 目录下创建 awdev-0.5 文件夹

图 2-4 驱动安装示意图 2

```
allwinnertech@allwinnertech-All-Series:/usr/src$ sudo dkms add awdev/0.5
Error! Could not find module source directory.
Directory: /usr/src/awdev-0.5 does not exist.
allwinnertech@allwinnertech-All-Series:/usr/src$ mkdir awdev-0.5
mkdir: 无法创建目录“awdev-0.5”: 权限不够
allwinnertech@allwinnertech-All-Series:/usr/src$ sudo mkdir awdev-0.5
allwinnertech@allwinnertech-All-Series:/usr/src$ ls
awdev-0.5          linux-hwe-5.11-headers-5.11.0-40
linux-headers-5.11.0-27-generic linux-hwe-5.11-headers-5.11.0-41
linux-headers-5.11.0-40-generic linux-source-5.4.0
linux-headers-5.11.0-41-generic linux-source-5.4.0.tar.bz2
linux-hwe-5.11-headers-5.11.0-27
```

5. 把 awusb.c、dkms.conf、Makefile 3 个文件拷贝到 /usr/src/awdev-0.5。



说明

Ubuntu 20.04 版本, 还需对 Makefile 做如图 2-5 所示修改。

图 2-5 Makefile 修改内容示意图

```
1 obj-m := awusb.o
2 KDIR := /lib/modules/$(shell uname -r)/build
3 PWD := $(shell pwd)
4
5 default:|
6     $(MAKE) -C $(KDIR) M=$(shell pwd)
7 clean:
8     $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) clean
9     rm -rf Module.markers module.order module.sysvers
10
```

6.再次执行 `sudo dkms add awdev/0.5`, 执行成功会提示如图 2-6 所示内容。

图 2-6 驱动安装示意图 3

```
allwinnertech@allwinnertech-All-Series:/usr/src$ sudo dkms add awdev/0.5
Creating symlink /var/lib/dkms/awdev/0.5/source ->
/usr/src/awdev-0.5
DKMS: add completed.
```

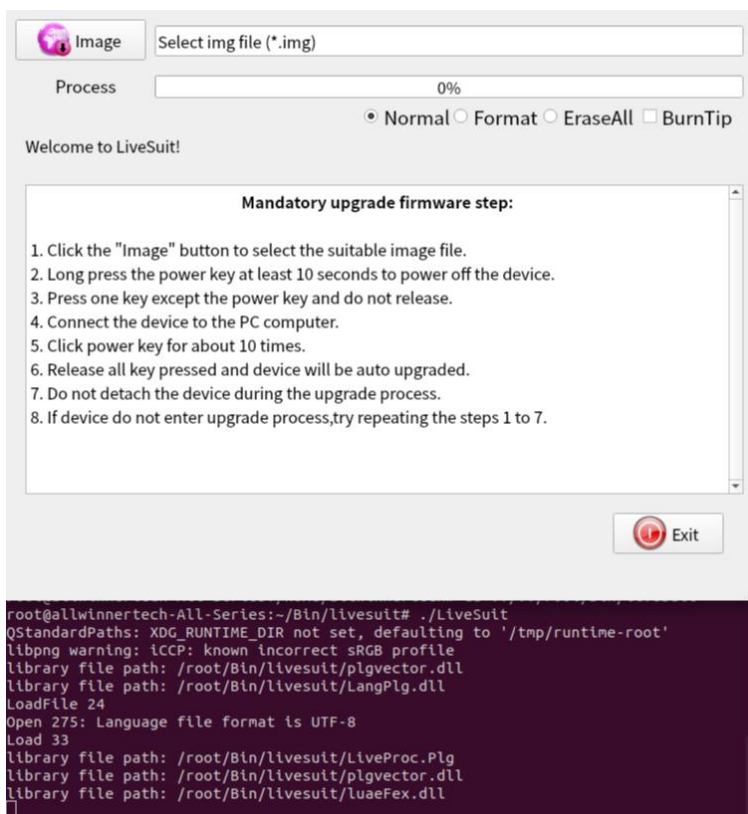
7.依次执行 `sudo dkms build awdev/0.5`; `sudo dkms install awdev/0.5`; `dkms status` 出现如图 2-7 所示结果证明安装驱动成功。

图 2-7 驱动安装示意图 4

```
allwinnertech@allwinnertech-All-Series:/usr/src$ sudo dkms build awdev/0.5
Kernel preparation unnecessary for this kernel. Skipping...
Building module:
cleaning build area...(bad exit status: 2)
make -j4 KERNELRELEASE=5.11.0-41-generic default KERNELVERSION=5.11.0-41-generic...
cleaning build area...(bad exit status: 2)
DKMS: build completed.
allwinnertech@allwinnertech-All-Series:/usr/src$ sudo dkms install awdev/0.5
awusb.ko:
Running module version sanity check.
- Original module
- Installation
- Installing to /lib/modules/5.11.0-41-generic/updates/dkms/
depmod.....
DKMS: install completed.
allwinnertech@allwinnertech-All-Series:/usr/src$ dkms status
awdev, 0.5, 5.11.0-41-generic, x86_64: installed
```

8.在 root 权限下执行 `./LiveSuit` 即可启动工具, 如图 2-8 所示。此时插上进入量产状态的设备即可开始进行烧录。

图 2-8 启动工具示意图



升级模式说明：

Normal：保留数据升级，选择此模式，则升级前，除了 UDISK 分区、私有分区和安全分区，格式化其他分区。

Format：分区擦除升级，选择此模式，则升级前，格式化普通分区，保留私有分区和安全分区。

EraseAll：全盘擦除升级，选择此模式，则升级前格式化全盘，包括 UDISK 分区、私有分区和安全分区。



注意

- EraseAll（全盘擦除升级）需要固件中的烧录插件支持该功能，否则该模式不起作用。
- Android R 及之后，UDISK 分区名称叫做 userdata 分区，即 UDISK=userdata。

BurnTip：勾选此项，工具会在开始烧录时弹窗提示烧录开始。

2.3 auto fel 说明

Auto fel 功能是用来方便升级已经升级过的机器。

当工具检测到插入的设备会首先发送 auto fel 的握手命令，如果成功，则会提示“检测到可升级设备，是否要升级该设备？”的信息，此时用户可以选择升级或取消。



注意

1. 此项功能需要烧录已经升级过的机器固件（即设备需要先烧录一次已经打开 auto fel 功能的固件），该固件打上了 uboot 的补丁且打开了 auto fel 功能。

-
2. uboot 中配置 CONFIG_CMD_SUNXI_AUTO_FEL=y
 3. dts 配置
- ```
[target]
auto_fel = 1
[platform]
sprite_keep_usb = 1
```
- 

以上详细 uboot 配置请咨询 uboot 负责人。

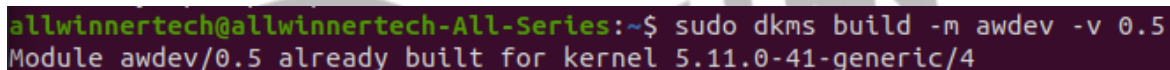
## 2.4 驱动 deb 包编译及安装

### 2.4.1 驱动 deb 包编译

编译步骤:

1. 请按照章节 2.2 中的步骤, 执行到步骤 7 中的 `sudo dkms build awdev/0.5`;
2. 完成以上步骤后, 在 `/var/lib/dkms/awdev/0.5` 目录中会成功编译出 `awusb.ko`。  
此时再次执行 `sudo dkms build -m awdev -v 0.5`, 会提示模块已编译, 如图 2-9 所示。

图 2-9 awdev build already 示意图



```
allwinnertech@allwinnertech-All-Series:~$ sudo dkms build -m awdev -v 0.5
Module awdev/0.5 already built for kernel 5.11.0-41-generic/4
```

3. 执行 `sudo dpkg -l` 查看 `dh-make` 模块是否安装, 如图 2-10 所示。

图 2-10 dh-make 模块安装示意图



```
ii dh-make 2.201903
```

如果模块安装目录中没有 `dh-make` 模块, 请执行 `sudo apt-get install dh-make`, 完成 `dh-make` 模块安装。

4. 执行 `sudo dkms mkdeb -m awdev -v 0.5`, 成功编译后 `deb` 包存放于 `/var/lib/dkms/awdev/0.5/deb` 目录中, 如图 2-11 所示。

图 2-11 deb 包编译示意图

```
allwinnertech@allwinnertech-All-Series:~$ sudo dkms mkdeb -m awdev -v 0.5
Using /etc/dkms/template-dkms-mkdeb
copying template...
modifying debian/changelog...
modifying debian/compat...
modifying debian/control...
modifying debian/copyright...
modifying debian/dirs...
modifying debian/postinst...
modifying debian/prerm...
modifying debian/README.Debian...
modifying debian/rules...
copying legacy postinstall template...
Copying source tree...
Building binary package...dpkg-buildpackage: 警告: 使用超级用户命令
dpkg-source --before-build .
fakeroot debian/rules clean
dh_clean: warning: Compatibility levels before 9 are deprecated (level 7 in use)
debian/rules build
fakeroot debian/rules binary
dh_installdirs: warning: Compatibility levels before 9 are deprecated (level 7 in use)
dh_strip: warning: Compatibility levels before 9 are deprecated (level 7 in use)
dh_compress: warning: Compatibility levels before 9 are deprecated (level 7 in use)
dh_installdeb: warning: Compatibility levels before 9 are deprecated (level 7 in use)
dh_shlibdeps: warning: Compatibility levels before 9 are deprecated (level 7 in use)
dpkg-genbuildinfo --build=binary
dpkg-genchanges --build=binary >../awdev-dkms_0.5_amd64.changes
dpkg-genchanges: info: binary-only upload (no source code included)
dpkg-source --after-build .

DKMS: mkdeb completed.
Moving built files to /var/lib/dkms/awdev/0.5/deb...
Cleaning up temporary files...
```

以上, 成功编译出驱动 deb 包, 请安装 2.3.2 章节指示进行安装即可。

安装成功后, 执行 `sudo dpkg -l`, 在模块目录中可以看到 awdev 驱动模块, 如图 2-12 所示。

图 2-12 awdev 驱动安装示意图

|    |            |     |
|----|------------|-----|
| ii | awdev-dkms | 0.5 |
|----|------------|-----|

## 2.4.2 驱动 deb 包安装

1. 安装目录保留有驱动的备份文件包:

- (1) Ubuntu 平台安装包: awdev-dkms\_0.1\_all.deb
- (2) RedHat, Fedora, CentOS 平台的安装包: awdev-0.1-1dkms.noarch.rpm

2. 如果更新过内核版本, 驱动没有自动加载, 请手动安装一次驱动, 驱动安装需要 root 权限。

3. 打开终端, Ubuntu 平台使用命令 `sudo dpkg -i awdev-dkms_0.1_all.deb` 安装。

RedHat, Fedora, CentOS 平台使用命令 `sudo rpm -ivh awdev-0.1-1dkms.noarch.rpm` 安装。

## 2.5 注意事项

1. 安装 awdev-dkms 提示缺少 dkms 时, 可尝试执行命令: `sudo apt-get install dkms`。
2. 编译 awusb.ko 提示失败时, 如果是 signal\_pending 问题, 是因为高版本内核的 signal 头文件路径有变, 可直接在 awusb.c 添加 `#include <linux/sched/signal.h>` 即可。此项为 awdev-0.5 版本驱动支持 4.11.0 及以上内核版本的主要更新内容, 已经在预编译定义中实现自动识别内核版本号, 无需人为修改源码。
3. 通过 insmod 命令加载 awusb.ko 提示失败时, 首先 `sudo make clean` 然后 `sudo make` 生成使用当前工具链编译出来的 ko, 再加载。如果仍然无法加载, 则是由于高版本操作系统的内核使能了 EFI\_SECURE\_BOOT\_SIG\_ENFORCE 配置, 即阻止加载第三方模块, 当使用桌面版 Ubuntu 时, 可选择安装 awdev-dkms\_0.1\_all.deb, 会出现弹窗提示, 确认即可; 通用的解决方法如下:
  - (1) 打开终端输入 `sudo apt install mokutil`;
  - (2) 继续输入 `sudo mokutil -disable-validation`;
  - (3) 终端会让你设置 8-16 位的密码, 请记住设置的密码;
  - (4) 重启电脑, 会出现蓝屏, 显示选项如下:

```
-Continue boot
-Change Secure Boot state
-Enroll key from disk
-Enroll hash from disk
请选择 Change Secure Boot state
```

- (5) 系统会让你输入之前设置密码的某一位;
  - (6) 接下来进入 Disable Secure Boot 选择界面, 选择 yes 后回到最开始界面, 在选择 reboot, 重新进入系统。
4. 提示缺少 libpng12.so.0, 是由于 Ubuntu 14 以上就不再支持 libpng12, 然而 LiveSuit 依赖 libpng12, 所以需要使用特定方法安装低版本的 libpng:
    - (1) 打开终端输入 `sudo vim /etc/apt/source.list`;
    - (2) 根据 Ubuntu 官网的提示, 软件源中添加 deb (<http://cz.archive.ubuntu.com/ubuntu>);
    - (3) 保存并退出 vim, 终端输入 `sudo apt-get update`;
    - (4) 继续输入 `sudo apt-get install libpng12-0`。

## 2.6 Livesuit in Ubuntu20.04

### 2.6.1 驱动安装失败

#### 2.6.1.1 驱动安装失败

LiveSuitV307\_For\_Linux64.zip 工具安装失败, 提示驱动安装不成功, 建议使用 sunxi.org 代码。

仓库地址: <https://github.com/linux-sunxi/sunxi-livesuite.git>

最近 commit: 1a0b52a1aeef60de3144c148063e061bbc542684

### 2.6.1.2 驱动安装无权限或找不到

按照 readme, 拷贝编译生成的 awusb.ko 到/lib/modules/`uname -r`/kernel, 需要注意, ubuntu20.04 可能因为使能了 secureboot, 导致驱动安装有问题, 提示没有权限或者找不到驱动。

修复方式:

- (1) openssl req -new -x509 -newkey rsa:2048 -keyout MOK.priv -outform DER -out MOK.der -nodes -days 36500 -subj "/CN=awusb/"
- (2) sudo /usr/src/linux-headers-\$(uname -r)/scripts/sign-file sha256 ./MOK.priv ./MOK.der \$(modinfo -n awusb)
- (3) sudo mokutil --import MOK.der, 中间会提示输入密码, 下次 reboot 时会用到.
- (4) 重启, 启动时会弹出 perform MOK management, 选择 Enroll MOK > Continue, 会提示收入密码, 这个密码是第三步设置的密码.

参考链接地址:

<https://unix.stackexchange.com/questions/327240/virtualbox-not-working-modules-not-working>

### 2.6.2 编译驱动失败

按照 sunxi-livesuite/readme 文件, 编译 awusb 驱动时可能会遇到的问题。

1. 编译提示 \*\*\* Error during writing of the configuration.

修复方式:

```
diff --git a/awusb/Makefile b/awusb/Makefile
index e0e4d71..09713c2 100644
--- a/awusb/Makefile
+++ b/awusb/Makefile
@@ -3,7 +3,7 @@ KDIR := /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)

default:
- $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules
+ $(MAKE) -C $(KDIR) M=$(shell pwd)

clean:
$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) clean
rm -rf Module.markers module.order module.sysvers
```

2. 提示 sunxi-livesuite/awusb/awusb.c:376:8: error: implicit declaration of function 'signal\_pending' [-Werror=implicit-function-declaration]

修复方式:

```
diff --git a/awusb/awusb.c b/awusb/awusb.c
index 16b3e2c..2a8a02b 100644
--- a/awusb/awusb.c
+++ b/awusb/awusb.c
@@ -26,7 +26,6 @@

#include <linux/module.h>
#include <linux/kernel.h>
-#include <linux/signal.h>
#include <linux/sched.h>
#include <linux/hardirq.h> //<linux/smp_lock.h>
#include <linux/errno.h>
@@ -38,6 +37,11 @@
#include <linux/usb.h>
#include <linux/wait.h>

+#include <linux/version.h>
+#if LINUX_VERSION_CODE >= KERNEL_VERSION(4, 11, 0)
+#include <linux/sched/signal.h>
+#endif
+
/* by Cesc */
```

### 2.6.3 执行提示

执行 Livesuit, 可能会提示

tarting x86-64/LiveSuit.

x86-64/LiveSuit: error while loading shared libraries: libpng12.so.0: cannot open shared object file: No such file or directory

修复方式: 参考 <https://blog.csdn.net/TaChean/article/details/104873253>

- (1) sudo add-apt-repository ppa:linuxuprising/libpng12
- (2) sudo apt update
- (3) sudo apt install libpng12-0



## 著作权声明

版权所有©2023 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、、**全志科技**、 Allwinner Technology（不完全列举）均为珠海全志科技股份有限公司

的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。