Weston ivi-shell 在 Jacinto7 Soc 中的应用

Fredy Zhang *EP FAE*

摘要

Jacinto™ 7 TDA4VM 是目前 TI 最新一代的汽车处理器，面向 ADAS 和自动驾驶车辆 (AV) 应用，并且建立在 TI 在 ADAS 处理器市场十余年领先地位中积累的广泛市场知识基础之上。TDA4VM 以业界领先的功耗/性能比为传统和深度学习算法提供高性能计算，并具有很高的系统集成度，从而使支持集中式 ECU 或多种传感器的高级汽车平台实现可扩展性和更低的成本。Weston 框架专用于显示器的管理，对其输入设备（触摸屏/鼠标/键盘）的支持。从内部结构来看分为窗口管理（Shell），合成器（compositor）和输入管理几个部分。 Weston 是最小且快速的合成器，适用于许多嵌入式和移动场景。窗口管理部分 TI PSDKLA 默认已经支持了 Desktop Shell，在汽车的应用中，尤其是车载信息娱乐系统，像 Desktop Shell 专注于传统的鼠标和键盘用户界面管理难以胜任汽车应用中多窗口复杂的界面管理。IVI-shell（In-vehicle infotainment）提供了图层管理的 API 和简单的 shell 协议给客户端，客户端可以非常便捷地配置并管理窗口应用。因此,IVI-shell 在汽车里面得到了广泛的应用。

本手册旨在对 Wayland/Weston 框架快速概述，为使用 TI Jacinto™ 7 的人员提供有关使用和配置 weston ivi-shell 的一些参考，以便更好地理解 Wayland/Weston 的结构以及快速在 TI Jacinto™ 7 Soc 上使用 TI PSDKLA 快速验证 weston ivi-shell，实现桌面应用的管理。

修改记录

Version	Date	Author	Notes
1.0	April 2020	Fredy Zhang	First release

目录

图

1. Wayland/Weston 介绍

Wayland/Weston 框架专用于显示器的管理，包括内容的合成，对其输入设备（触摸屏，鼠标，键盘）事件管理和其设备的配置（背景，壁纸，分辨率，多屏显示等）。Wayland 是一个指定显示器和客户端之间的通信协议。Wayland 旨在替代 XWindow 系统，更易于开发和维护。Weston 是 Wayland 合成器的参考实现，该合成器是使用 Wayland 协议的显示服务器，Weston 是小且快的合成器，适用于许多嵌入式和移动应用场景。

Weston 从内部体系结构如图 1 所示：主要分为窗口管理（Shell），合成器（compositor），渲染器（Renderer）和输入管理（input manager）几个部分。从流程上来讲，一方面输入模块接收用户输入，然后 Shell 做出相应的窗口管理操作；另一方面将该 input event 传给 Weston 之前注册了相应输入事件的 Client 程序. Client 收到后会在其处理函数中做相应动作，如调整视图重绘。如有重绘发生，client 端新 buffer 渲染完成后，client 将其句柄 handle 传给 server，之后是 compositor 合成后 renderer 进行渲染，最后输出到显示设备。Weston 启动过程中会分别加载几个模块：shell backend 用于窗口管理；render backend 用于合成渲染；compositor backend 用于合成输出；input manager backend 用于输入管理。

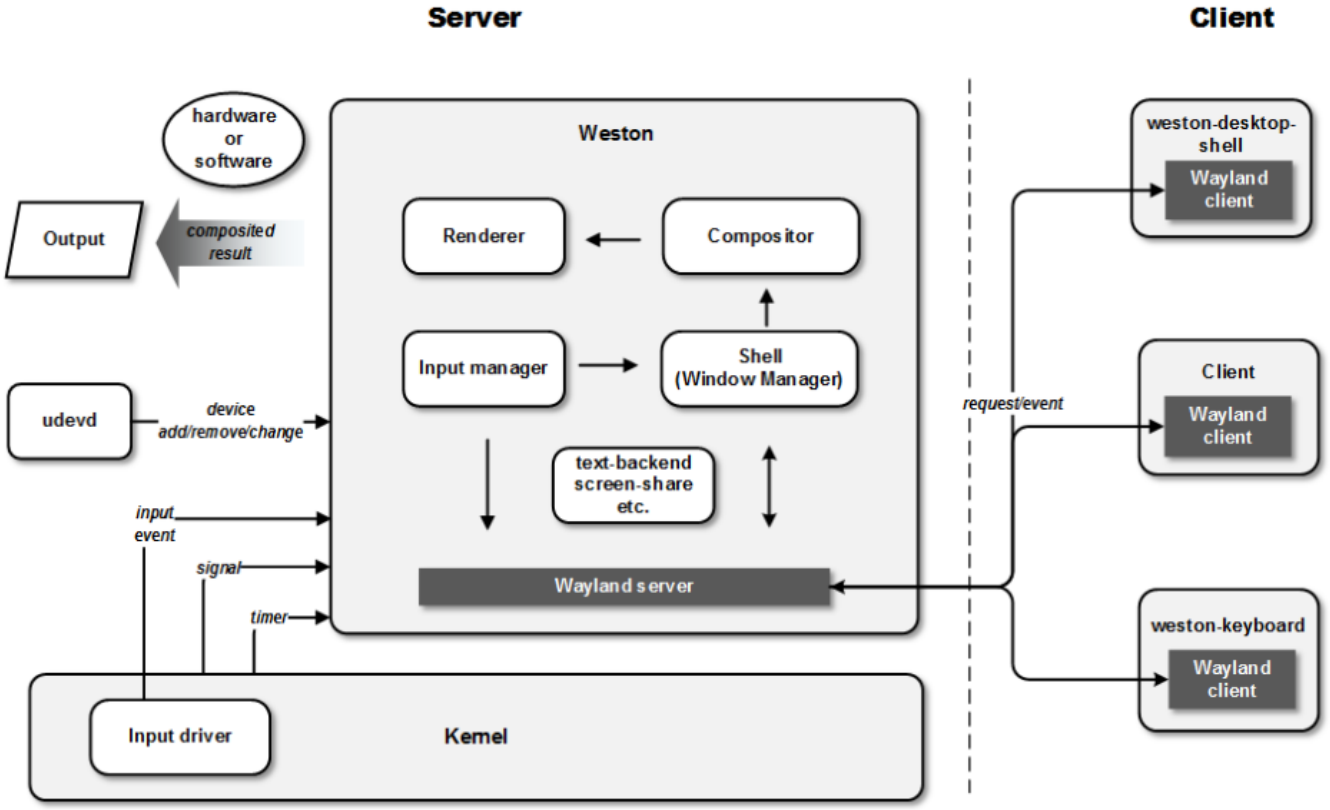

图 **1**. Weston 框架

2. 系统框架

2.1. 嵌入式系统组成

一个嵌入式 Linux 系统里面常常包含 GPU，DSS 等硬件。典型的 Linux Weston 系统框架如图 2 所示：

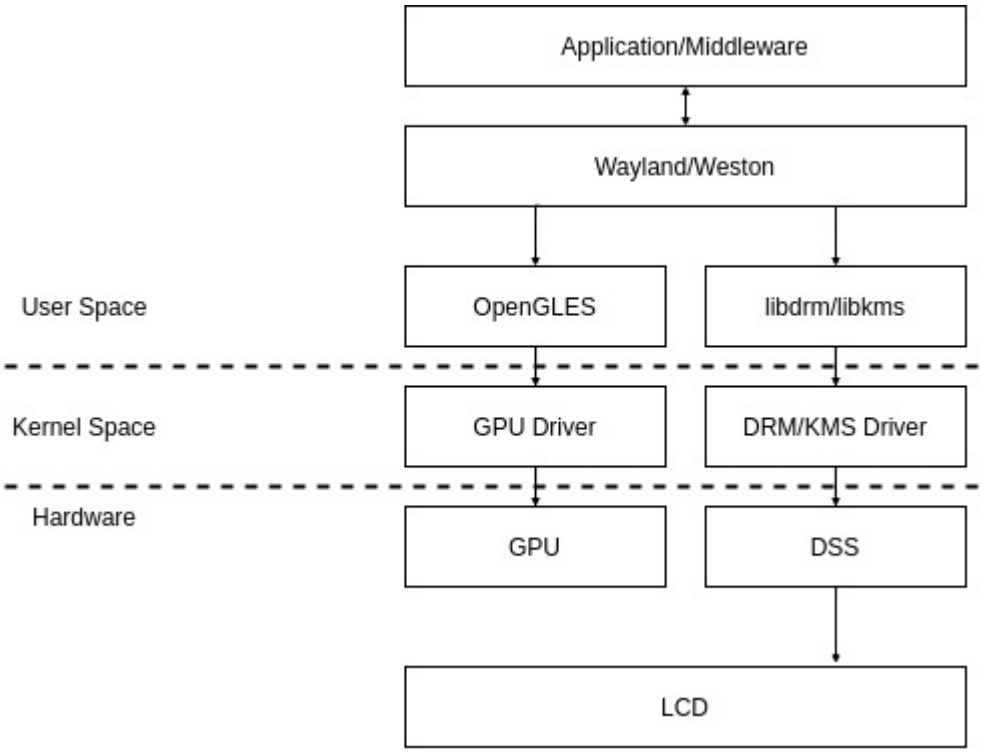

图 **2.** Linux Weston 系统框架

从应用到硬件，从 User Space 到 Hardware，包含如下模块：

- 应用程序/中间件（用户空间）：Wayland 客户端应用程序或中间件依赖于 Wayland 协议。该客户端可以是传统应用程序或客户端。
- Wayland/Weston（用户空间）：实现 Wayland 显示服务器协议 Wayland 库，以及使用 Wayland 协议来实现在 Linux 内核 KMS 上运行的显示服务器的 weston 库，该组件基于 OpenGLES 和 DRM，可对 Linux 输入设备进行管理。
- OpenGLES（用户空间）：Weston 用于基于 GPU 合成的 OpenGLES 库。
- Libdrm/libkms（用户空间）： Weston 合成器使用 libdrm 和 libkms 库配置显示路径和刷新显示内容。
- GPU 驱动程序（内核空间）： Linux 内核驱动程序，用于将 OpenGLES 命令传输到 GPU 硬件模块。
- DRM/KMS 驱动程序（内核空间）： DRM/KMS Linux 内核驱动程序，用于访问显示硬件 DSS。
- GPU（硬件）： GPU 硬件模块。
- DSS（硬件）： Display Sub System， 显示硬件子系统。
- LCD （硬件）：连接 DSS 的显示器。

2.2. 窗口管理（Shell）

当前我们有三个窗口管理 shell： Desktop-shell, Fullscreen-shell 和 IVI-shell，每个 shell 都有自己的公共的客户端协议接口。 客户端程序必须按照一种 shell 协议编写客户端，否则它将无法正常工作。

- Desktop Shell: 像现代 X 桌面环境一样，专注于传统的键盘和鼠标用户界面以及熟悉的类似桌面的窗口管理。 Desktop-shell 由 desktop-shell.so 和 weston-destop-shell 插件组成。
- Fullscreen Shell: fullscreen-shell 适用于客户端想要接管所有的输出的场景。 客户端在 weston 运行一个 compositor。 另外一个 compositor 不需要处理任何特定于平台的内容例如 DRM/KMS 或 evdev/libinput。fullscreen-shell 仅由 fullscreen-shell.so 插件组成。

- IVI shell : IVI (in-vehicle infotainment) , IVI-shell 是一种专用的 shell，它向控制模块提供了 GENIVI 层管理兼容的 API，对客户端来讲，这是一种简单的协议。 IVI-shell 首先加载 ivi-shell.so 插件，然后加载 ivi-controller.so 插件。

针对上述三种 shell，Desktop-shell 最常用，适用于传统的键盘和鼠标的用户界面管理，TI 默认的 SDK 采用 Desktop Shell。Fullscreen Shell 比较少用， 适用于客户端想要接管所有的输出的场景，它的一个实际用例是屏幕录制和共享。IVI-Shell 由于对客户端提供了简单的控制接口，从而使用者可以方便地对图层及其显示内容的上下层关系，透明度及位置关系等进行控制。为多窗口的应用开发带来了便捷性。因此，在汽车 IVI 和 ADAS 应用中，客户普遍采用了 IVI-shell。

2.3. IVI Shell

IVI-shell 环境如下图 3 所示，主要的配置文件是 weston.ini ，其中 shell 环境配置为 ivi-shell.so；modules 配置为 ivi-controller.so；ivi-input-module 配置为 ivi-input-controller.so。客户端利用 Layer-add-surfaces 和 LayerManagerControl 对图层和 surface 进行配置。

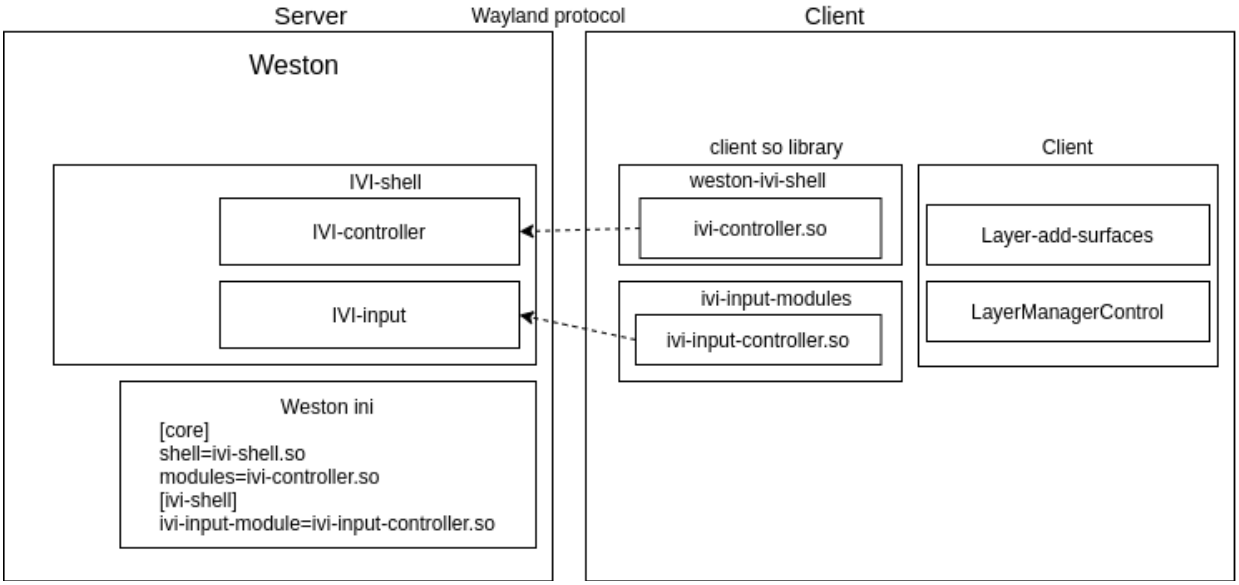

图 **3**. IVI-shell 框架

首先，介绍一些 surface。surface 代表 wayland client 的一个绘图表面。client 通过画好的 buffer attach 到 surface 上来更新窗口，因此说 surface 是 buffer 的载体。在 weston 中，shell 是窗口管理器，因此一个 surface 要想被窗口管理器管理，需要创建相应的 shell surface。如下图 4：在一个 IVI 屏幕上，我们首先要创建一个图层 layer，然后在图层上创建 surface 显示内容，在一个 layer 上可以创建多个 surface 来显示多个客户端应用。

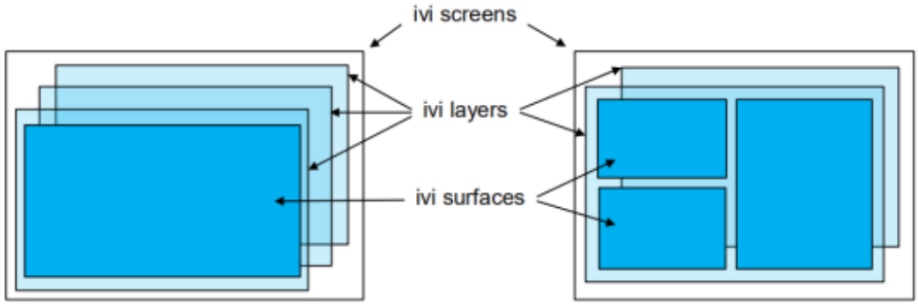

图 **4**. IVI Screen 框架

3. Jainto7 Soc 运行 weston

3.1. Weston 基本操作

Weston 的默认配置文件是 weston.ini, 默认的配置路径是/etc/weston.ini
运行下面的命令启动/停止/重启 weston：

- target# /etc/init.d/weston stop
- target# /etc/init.d/weston start
- target# /etc/init.d/weston restart

通过串口控制台或者 SSH 控制台，用下面的命令运行 weston 的客户端应用：

- target# /usr/bin/weston-flower
- target# /usr/bin/weston-clickdot
- target# /usr/bin/weston-cliptest
- target# /usr/bin/weston-dnd
- target# /usr/bin/weston-editor
- target# /usr/bin/weston-eventdemo
- target# /usr/bin/weston-image /usr/share/weston/terminal.png
- target# /usr/bin/weston-resizor
- target# /usr/bin/weston-simple-egl
- target# /usr/bin/weston-simple-shm
- target# /usr/bin/weston-simple-touch
- target# /usr/bin/weston-smoke
- target# /usr/bin/weston-info
- target# /usr/bin/weston-terminal

退出 weston 应用：

- 在控制台退出 weston 应用程序 ctrl+c

3.2. Desktop shell

TI PSKDLA 默认已经配置支持了 Desktop shell，其配置/etc/weston.ini 如下：

```
[core]
require-input=false

[shell]
locking=false
animation=zoom
panel-position=top
background-image=/usr/share/demo/j7-evm-oob-wallpaper.jpg
startup-animation=fade

[screensaver]
# Uncomment path to disable screensaver
#path=@libexecdir@/weston-screensaver
```

可以查看 weston 的状态使用命令（target# systemctl status weston），log 如下，从 log 来看 weston service 显示 running。

```
    [[0;1;32m[[0m weston.service
       Loaded: loaded (/etc/init.d/weston; generated)
       Active: [[0;1;32mactive (running)[[0m since Thu 2019-10-24 03:20:55 UTC; 46s a
go
         Docs: man:systemd-sysv-generator(8)
      Process: 903 ExecStart=/etc/init.d/weston start (code=exited, status=0/SUCCESS)
        Tasks: 3 (limit: 989)
       Memory: 30.6M
       CGroup: /system.slice/weston.service
               ├─ 926 /bin/sh /usr/bin/runWeston
               ├─ 931 weston --idle-time=0
               └─1024 /usr/libexec/weston-keyboard

    Oct 24 03:20:55 j7-evm systemd[1]: Starting weston.service...
    Oct 24 03:20:55 j7-evm weston[903]: Starting Weston
    Oct 24 03:20:55 j7-evm systemd[1]: Started weston.service.
```

另外还有一种方法来检车 weston 的状态，查看/var/log/weston.log 来检查 weston 的运行状态。

3.3. IVI shell

TI PSKDLA 默认已经配置支持了 Desktop shell，也支持 weston shell 配置为 ivi-shell。从 desktop shell 更改到 ivi-shell，修改/etc/westion.ini 如下:

```
[core]
shell=ivi-shell.so
modules=ivi-controller.so

[ivi-shell]
ivi-input-module=ivi-input-controller.so
```

更改到 ivi-shell 后需要，需要重启 weston，使用如下命令：

- target# /etc/init.d/weston stop
- target# /etc/init.d/weston start

Weston 重启后，使用如下命令检查 weston 是否启动正常：

- target# systemctl status weston

> 注意
>
> 当使用 IVI-shell 启动后，默认的屏幕背景是黑色。这个是与 Desktop-shell 环境不同的。 Desktop-shell 环境下，weston 启动后你可以看到 weston 桌面。

当 weston 配置为 ivi-shell 后，使用 ivi 应用协议去管理客户端程序。Wayland-ivi-extension 提供了 ivi-controller.so 插件去配置 surfaces/layers/screens 的属性，也提供了 ivi-input-controller.so 插件去管理输入事件。

所有的 weston 客户端示例，例如“weston-simple-egl”、“weston-simple-shm ”、“weston-flower”等，都支持 IVI-shell。 TI PSDKLA 包里面包含了“wayland-ivi-extension”。 Wayland-ivi-extension 包含了应用程序“layer-add-surface”和“LayerManagerControl”。这些应用程序允许客户调用“ivi-shell”的各种功能（缩放，调整位置，图层，显示/隐藏等）并控制应用程序。

启动 weston ivi-shell 后，用下面的命令运行客户端应用：

- target# weston-simple-shm &

这个时候，屏幕上没有任何显示，因为我们还没有创建图层，用下面的命令创建一个 Layer ID 为 1000 带 2 个 surface 的图层，这个时候你就可以看到 weston-simple-shm 应用显示在屏幕上了，控制台 log 显示了 surfaceID 为 10275.

- target# layer-add-surfaces -l 1000 -s 2 &
- 控制台 log 如下：

```
root@j7-evm:~# layer-add-surfaces -l 1000 -s 2 &
layer-add-surfaces: layer (1000) on display () created, waiting for 2 s
urfaces ...
layer-add-surfaces: layer (1000) destination region: x:0 y:0 w:1920 h:1
080
layer-add-surfaces: layer (1000) visibility TRUE
layer-add-surfaces: layer (1000) created
layer-add-surfaces: surface (10275) created
layer-add-surfaces: surface (10275) configured with:
    dst region: x:0 y:0 w:250 h:250
    src region: x:0 y:0 w:250 h:250
    visibility: TRUE
    added to layer (1000)
layer-add-surfaces: surface (10275) configured with:
    dst region: x:0 y:0 w:250 h:250
    src region: x:0 y:0 w:250 h:250
    visibility: TRUE
    added to layer (1000)
```

这个时候我们可以再运行一个客户端应用，因为我们创建两个 surface：

- target# weston-flower &

```
root@j7-evm:~# weston-flower &
[3] 1241
could not load cursor 'dnd-move'
could not load cursor 'dnd-copy'
could not load cursor 'dnd-none'
layer-add-surfaces: surface (10241) created
layer-add-surfaces: surface (10241) configured with:
    dst region: x:0 y:0 w:200 h:200
    src region: x:0 y:0 w:200 h:200
    visibility: TRUE
    added to layer (1000)
layer-add-surfaces: surface (10241) configured with:
    dst region: x:0 y:0 w:200 h:200
    src region: x:0 y:0 w:200 h:200
```

```
    visibility: TRUE
    added to layer (1000)
```

注意

使用 ivi 应用协议去管理客户端应用程序必须拥有唯一的 ID，这个 ID 在创建 layer 和 surface 时获得。分别为 LayerID 和 surfaceID。

现在用户可以通过 LayerManagerControl 去设置应用的位置、缩放、透明度、可见性等。

- target# LayerManagerControl set surface 10275 destination region 150 150 300 300
- target# LayerManagerControl set surface 10275 opacity 0.5
- target# LayerManagerControl set surface 10275 visibility 1
- target# LayerManagerControl set layer 1000 render order 10275,10241
- target# LayerManagerControl create layer 1000 600 400
- target# LayerManagerControl get layers
- target# LayerManagerControl get layer layer_id
- target# LayerManagerControl get surfaces
- target# LayerManagerControl analyze surface surface_id

最后，举个例子说明一下 IVI-shell 的使用。在一个 IVI+RVC 项目中，IVI 的 UI 和 RVC 的 UI 通常处于不同的图层中。在倒车的过程中，我们要调整 RVC 的图层在最上层。非倒车地情况下，我们我显示 IVI 的 UI。RVC 也可能还有多个视图，比如视图切换、缩放等。IVI-shell 非常方便地提供了 LayerManagerControl 的接口，用户可以方便地调整图层地缩放、位置、透明度、可见性等。

4. 参考

1. https://manpages.debian.org/experimental/weston/weston.1.en.html
2. https://wayland.freedesktop.org/
3. https://at.projects.genivi.org/wiki/display/PROJ/Wayland+IVI+Extension+Design
4. http://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/index.html
5. https://www.ti.com.cn/cn/lit/ds/symlink/tda4vm.pdf?ts=1618976597409&ref_url=https%253A%252F%252Fwww.ti.com.cn%252Fproduct%252Fcn%252FTDA4VM
6. https://lwn.net/Articles/618197/

重要声明和免责声明

TI 提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保或其他要求。这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款 (https:www.ti.com.cn/zh-cn/legal/termsofsale.html) 或 ti.com.cn 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122