Application Notes
ZHCADF6 – Oct. 2023

基于 DCC 和张氏标定的相机镜头畸变校正

Adam Hua *Sales and Marketing/Central FAE*

ABSTRACT

当前 Jacinto 和 Sitara 具有 ISP 的处理器中，ISP 调参的工作通常由 DCC tuning tool 帮助实施。其中相机畸变校准 LDC plugin 需要输入畸变图片和畸变表以实现畸变校正，并生成 xml 文件和 bin 文件提供给 SDK。目前生成畸变校正表依赖于摄像头模组厂提供的畸变函数或畸变表，本文提供一种基于张氏标定的畸变校正表生成方法，能够在无正确畸变函数或畸变表情形下快速生成畸变校正表。首先使用 OpenCV 集成的张氏标定方法，提供摄像头内参和畸变参数，接着利用内参与畸变参数生成 DCC 工具需要的畸变校正表。能够帮助用户节省与厂商沟通成本，更快速实现摄像头畸变校正。

Contents

Figures

1 简介

Jacinto TDA4 系列处理器与 Sitara AM62A 处理器都集成了 VPAC 模块，提供从摄像头采集的原始 RAW 图像到可用于显示或机器学习图像的初步图像处理功能，包含了白平衡、自动曝光控制、尺度变换、滤波、镜头畸变校正等功能。DCC Tuning Tool 工具提供了这些功能的调参功能，其中镜头畸变校正功能需要输入一张校正前图像以及一个畸变校正表以进行畸变校正。畸变矫正表用于表述无畸变的网格点与有畸变的网格点之间的差值，用于畸变校正时查表插值。

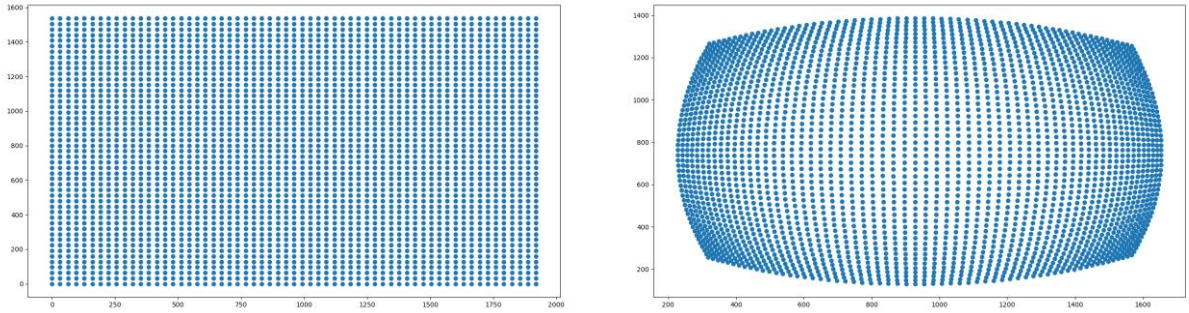


Figure 1.　理想无畸变网格点与畸变网格点

2 基于张氏标定的畸变校正表生成方法

张氏标定法能够利用几张简单的网格图片进行简单快速的摄像头标定[1]。根据张氏理论，将物体的在 3D 实际空间到像素坐标的过程分为从物空间到像空间的转换，像空间到畸变像空间的转换，最后从畸变像空间到像素空间的转换，这三步分别对应相机的外参，畸变参数和内参。

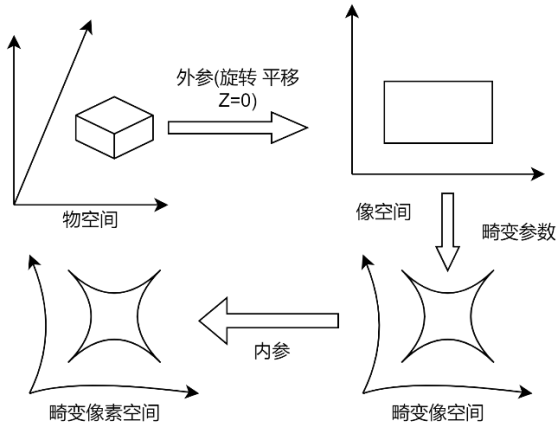


Figure 2.　空间变换

畸变校正表是从畸变像素空间到无畸变的像素空间的查找表，因此获得畸变参数和摄像机内参就可以将图像进行畸变校正。OpenCV 提供标定方法的快速实现接口，用几张网格图照片即可获取上述外参 R、畸变参数 K 和内参 A。对于物空间的任意点(X,Y,Z)，其最终像素坐标(u,v)经过以下步骤获得：

从物空间到像空间：

$$(\mathrm{x}', \mathrm{y}', z', 1)^{\mathrm{T}} = (\mathrm{r1}, \mathrm{r2}, \mathrm{r3}, \mathrm{t})(X, Y, Z, 1)^{T}$$

由于像空间已经可以视为二维平面，忽略像空间 z 方向上的数值，接着从像空间到畸变像空间：

$$(x'', y'') = \text{distort}(x', y')$$

从畸变像空间到像素空间：

$$(u, v) = A(x'', y'')$$

因此，对于理想无畸变网格点(u', v')到畸变下的像素空间坐标(u, v)的过程如下：

获取像空间坐标：

$$(x', y') = A^{-1}(u', v')$$

获取畸变下像素空间坐标：

$$(u, v) = A\ \text{distort}(x', y')$$

3 实现流程

3.1 摄像头参数标定

该方法的具体实现依赖于 OpenCV，首先制作或者购买一个有如下网格的 checker：

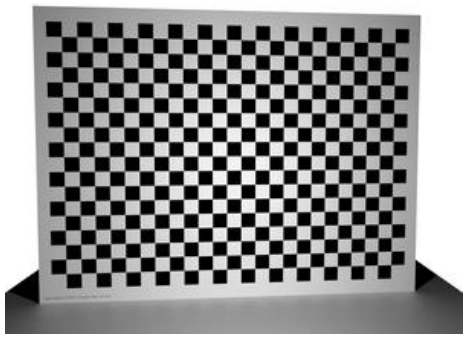
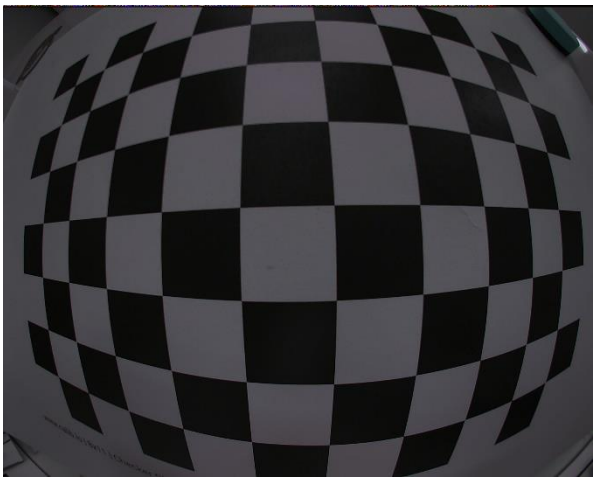
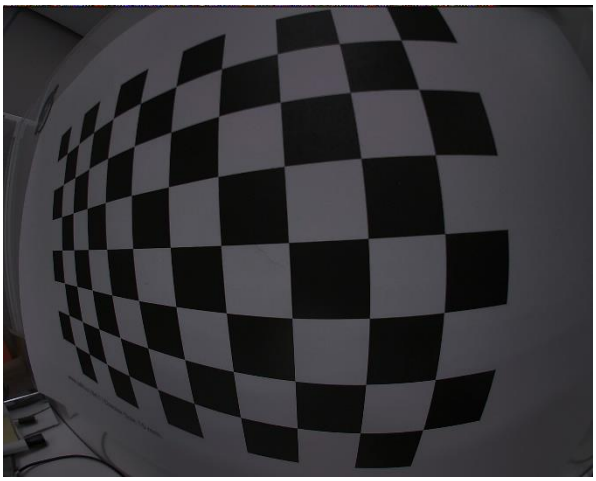


Figure 3. Chessboard Checker

从各个角度用 sensor 采集至少 3 张 checker 的图像：





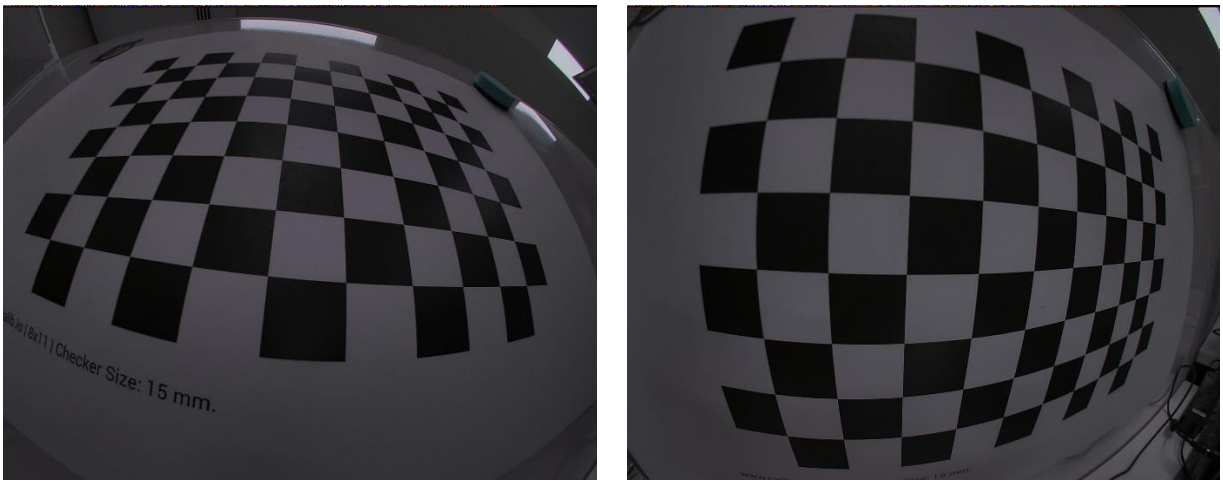


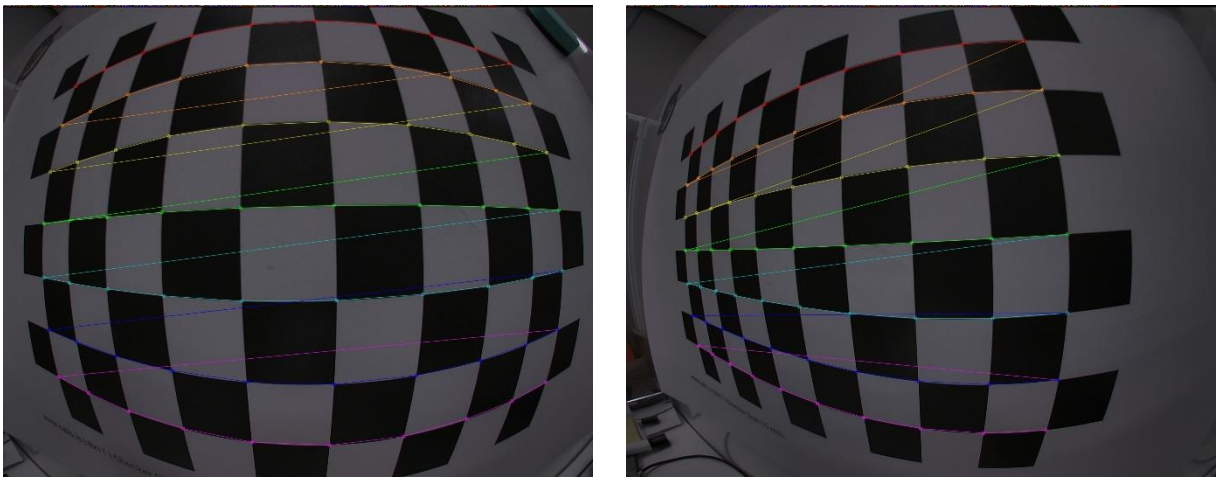
Figure 4.　不同角度的 Checker

调用 cv2.findChessboardCorners 方法找到所有的角点，需要注意的是拍摄的 Checker 图像背景应当尽量干净，尽量以白底为主，以避免角点查找失败，同时横向角点数量和纵向角点数量 corner_num_h, corner_num_v 必须与图片中实际的数量一致，否则会造成角点搜索失败。

```
1.  ret, corners = cv2.findChessboardCorners(gray, (corner_num_h,corner_num_v), flags=cv2.CALIB_CB_ADAPTIVE_THRESH )
```

接着可以调用 drawChessboardCorners 方法将搜索到的角点绘制在图片上。

```
1.  cv2.drawChessboardCorners(img, (corner_num_h,corner_num_v), corners, ret)
```



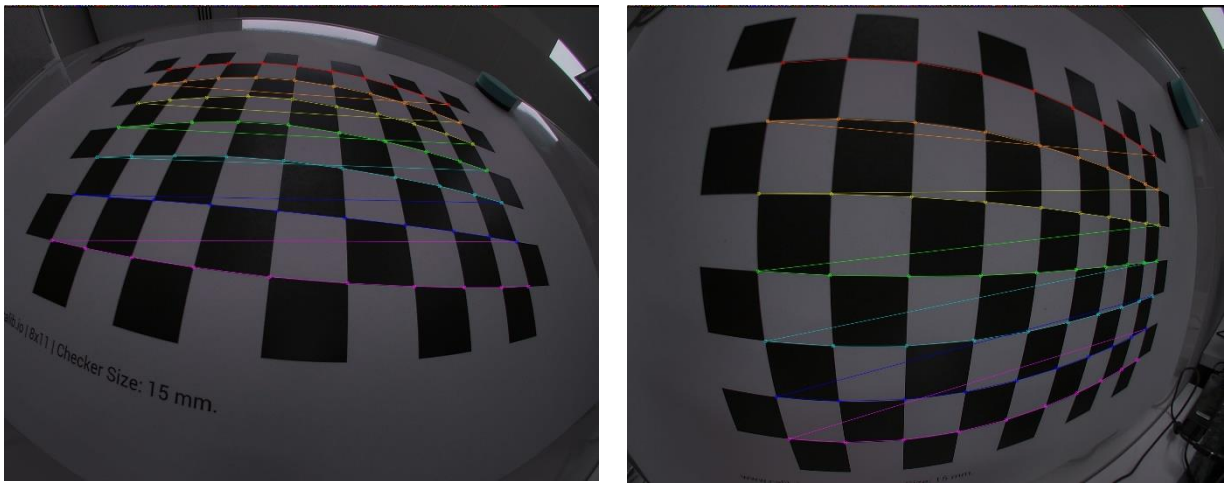


Figure 5. Checker 与检测到的角点

调用 calibrateCamera 方法标定相机参数，获取内参(mtx)，畸变参数(dist)，旋转外参(rvecs)，平移外参(tvecs)。

```
1. ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, img_size, None, None)
```

3.2 畸变标定表生成

调用 np.mgrid 初始化无畸变像素网格点，用 m 参数控制网格点的间隔。

```
1. undis_points = np.mgrid[0:width+1:(2**m), 0:height+1:(2**m)]
2. undis_points = undis_points.T.reshape(-1,2)
3. undis_points = undis_points.astype("float32")
```

调用 cv2.undistortPoints 方法，将畸变参数设置为 None 将理想无畸变像素点变换到像空间，该过程实际等效于将像素位置乘内参矩阵的逆矩阵。

```
1. undis_points_normal = cv2.undistortPoints(undis_points, mtx, None)
```

调用 np.dstack 方法将像空间 2D 点扩充为 3D 点，用 0 填充第三维，用于作为后续函数的输入。

```
1. undis_points_3D = np.dstack((undis_points_normal, np.zeros((undis_points_normal.shape[0],1))))
```

调用 cv2.projectPoints 方法将像空间的 3D 点根据内参和畸变参数投影到像素空间，得到畸变后的像素点位置。其中，设置旋转矩阵，平移矩阵为全 0 阵以跳过外参变换。

```
1. dis_points = cv2.projectPoints(undis_points_3D, np.zeros((1,3),dtype = np.float32), np.zeros((1,3), dtype = np.float32), mtx, dis)[0]
```

将畸变后网格点位置与无畸变理想网格点位置相减，得到畸变校正查找表。

```
1. mesh_x = dis_points[:,0,0] - undis_points[:,0]
2. mesh_y = dis_points[:,0,1] - undis_points[:,1]
```

将查找表写入 txt 文件中，以供 DCC 工具读取。

```
1.  np.savetxt("mesh_lut.txt",mesh,delimiter=' ', newline='\n',fmt='%6d')
```

在 DCC 工具中，打开 LDC plugin，将 Output frame width, Output frame height, Input frame width, Input frame height, Mesh frame width, Mesh frame height 全部设置为图像的分辨率长宽，如 1920，1080。将 Table subsampling factor 设置为初始化无畸变像素网格点时使用的参数 m 的值，如 5。
点击 Input image 选择一张没有进行过畸变校正的 YUV（8bit）图像加载到 DCC 工具中，点击 LDC LUT file 加载生成的 txt 畸变校正表。点击该图标 进行畸变校正，工具会自动生成输入的图片的校正后结果。

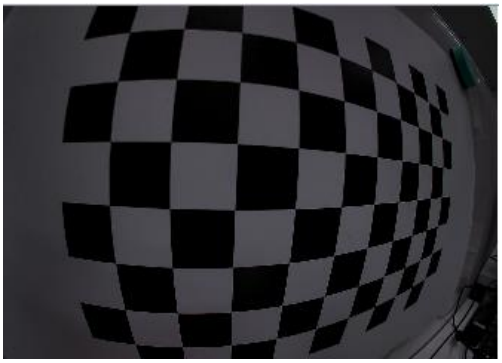


Figure 6.　输入的参考图像

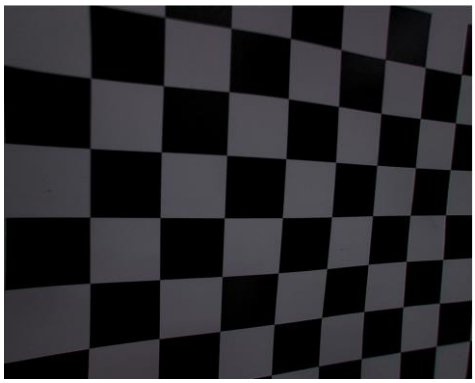


Figure 7.　畸变校正后图像

点击 生成需要的 XML 文件，用该 XML 文件替换在 SDK 中的 XML 文件，并运行 generate_dcc.sh 生成编译需要的资源文件。最后进行 SDK 和 DEMO APP 的编译。

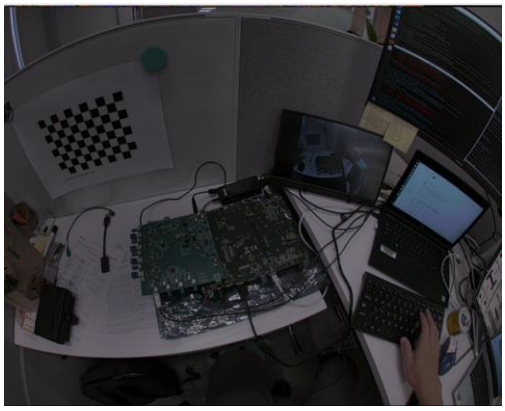


Figure 8.　畸变校正前

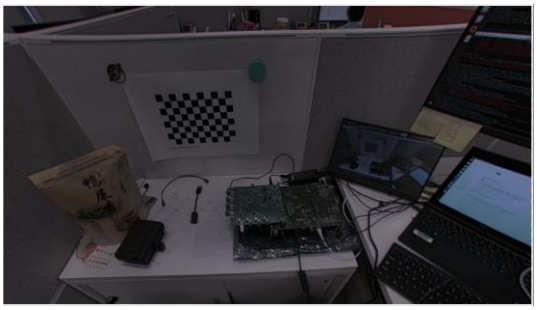


Figure 9.　畸变校正后

4　参考程序

参考程序采用 python 语言。
摄像头标定程序：

```
import numpy as np
import cv2
import glob
import matplotlib.pyplot as plt

## initial settings
img_width = 1920
img_height = 1536
corner_num_h = 10 # corner number horizontal
corner_num_v = 7 # corner number vertical

# prepare object points, like (0,0,0), (1,0,0), (2,0,0) ....,(6,5,0)
objp = np.zeros((corner_num_h*corner_num_v,3), np.float32)
objp[:,:2] = np.mgrid[0:corner_num_h, 0:corner_num_v].T.reshape(-1,2)

# Arrays to store object points and image points from all the images.
objpoints = [] # 3d points in real world space
imgpoints = [] # 2d points in image plane.

# Make a list of calibration images
```

```
images = glob.glob('000*.jpg')

# Step through the list and search for chessboard corners
for idx, fname in enumerate(images):
    img = cv2.imread(fname)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    cv2.namedWindow('img',cv2.WINDOW_NORMAL)
    cv2.imshow('img', gray)
    cv2.waitKey(0)
    # Find the chessboard corners
    ret, corners = cv2.findChessboardCorners(gray, (corner_num_h,corner_num_v), flags=cv2.CALIB_CB_ADAPTIVE_THRESH )

    # If found, add object points, image points
    if ret == True:
        objpoints.append(objp)
        imgpoints.append(corners)

        # Draw and display the corners
        cv2.drawChessboardCorners(img, (corner_num_h,corner_num_v), corners, ret)
        write_name = 'corners_found'+str(idx)+'.jpg'
        cv2.imwrite(write_name, img)
        cv2.namedWindow('img',cv2.WINDOW_NORMAL)
        cv2.imshow('img', img)

        cv2.waitKey(0)

cv2.destroyAllWindows()

import pickle

# Test undistortion on an image
img_size = (img.shape[1], img.shape[0])

# Do camera calibration given object points and image points
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, img_size,None,None)

# Save the camera calibration result for later use (we won't worry about rvecs / tvecs)
np.savetxt("Matrix.txt", mtx, newline="\n")
np.savetxt("distortion.txt", dist, newline="\n")
np.save("Matrix", mtx)
np.save("distortion", dist)
```

利用摄像头参数生成畸变校正表

```
import numpy as np
import cv2

## initial settings
width = 1920
height = 1536
m = 5

# initialize undistorted points
```

```
undis_points = np.mgrid[0:width+1:(2**m), 0:height+1:(2**m)]
undis_points = undis_points.T.reshape(-1,2)
undis_points = undis_points.astype("float32")

# load camera matrix and distortion coefficients
mtx = np.load('Matrix.npy')
dis = np.load('distortion.npy')

# get the corresponding distorted points
undis_points_normal = cv2.undistortPoints(undis_points, mtx, None)
undis_points_3D = np.dstack((undis_points_normal, np.zeros((undis_points_normal.shape[0],1))))
dis_points = cv2.projectPoints(undis_points_3D, np.zeros((1,3),dtype = np.float32), np.zeros((1,3), dtype = np.float32), mtx, dis)[0]

import matplotlib.pyplot as plt

xpoints = dis_points[:,0,0]
ypoints = dis_points[:,0,1]

plt.figure(0)
plt.scatter(xpoints, ypoints)
plt.show()

plt.figure(1)
plt.scatter(undis_points[:,0], undis_points[:,1])
plt.show()

# get the mesh table
mesh_x = dis_points[:,0,0] - undis_points[:,0]
mesh_y = dis_points[:,0,1] - undis_points[:,1]
mesh = np.dstack((mesh_x * 8, mesh_y * 8))[0].astype('int32')

np.savetxt("mesh_lut.txt",mesh,delimiter=' ', newline='\n',fmt='%6d')
```

5 总结

本文提供一种基于张氏标定方法的 DCC 工具中畸变校正表的生成方式，该方式不需要镜头畸变函数或畸变表，只需要使用未标定的摄像头对棋盘格标定板拍摄多个角度的照片即可完成 LDC 的参数设置。该方法适用于校正一些用于初步测试的摄像头模组以实现在没有畸变函数时，或者畸变函数不准确时的镜头畸变校正。

6 参考文献：

张友正标定法：https://ieeexplore.ieee.org/document/791289
DCC Tuning Guide TDAx: https://www.ti.com/lit/an/spracu7a/spracu7a.pdf
ISP Tuning Guide AM6xA: https://www.ti.com/lit/an/sprad86/sprad86.pdf

重要声明和免责声明

TI“按原样”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功能安全、信息安全、监管或其他要求。

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 的销售条款或 ti.com 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。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265