AN-1249
应用笔记

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

使用ADV8003评估板将3D图像转换成2D图像

作者：Paul Slattery

简介

ADV8003是一款视频信号处理器(VSP)，集成晶体管-晶体管逻辑(TTL)和串行视频输入，可去隔行并可以按比例缩放输入的视频信号。它能产生并混合基于位图的OSD(屏幕菜单)视频信号，然后利用器件的一路或多路输出来输出叠加OSD的视频。可用输出由两个HDMI发送器(Tx)、6个DAC构成的编码器(支持标清和高清)以及TTL输出组成。

本应用笔记描述如何通过ADV8003传输3D视频，以及如何采用ADV8003评估板(EVAL-ADV8003EB1Z和EVAL-ADV8003EB2Z)将3D图像转换为2D图像。随后，本文将针对每个应用提供用来配置ADV8003的脚本。

本应用笔记供希望了解ADV8003芯片并探索该芯片在定制设计和平台上的可行用途的用户参考。

用户应具备一定的视频平台设计或应用经验。用户无需成为这方面的专家，但应熟悉视频链和视频信号处理方面的基本概念与相关要素。

关于ADV8003评估板

两款ADV8003评估板——EVAL-ADV8003EB1Z和EVAL-ADV8003EB2Z——可用来将3D图像转换为2D图像。图1所示为ADV8003评估板的框图。ADV7850是一款12位、170 MHz视频与图形数字化仪，集成3D梳状解码器。它是一个带有HDMI、1.4 A发送器输出的四通道、HDMI、1.4 A快速开关接收器(Rx)。它将视频信号从HDMI发送器输出至ADV8003 TMDS接收器。ADV8003处理视频(按比例缩放、去隔行、OSD叠加)，然后将视频输出至HDMI发送器。

ADV8003 3D视频处理说明

今天，3D视频内容在消费电子行业无处不在，其信号源包括蓝光播放器和3D便携式摄像机等。如果HDMI源支持3D视频格式，那么根据HDMI规格(HDMI网站提供该规格)它必须支持下列3D视频格式中的一种：

- 帧封装
- 并排(半幅)
- 顶部和底部

本文说明如何通过ADV8003传输3D视频，以及如何采用ADV8003评估板将3D图像转换为2D图像。对于拥有两个显示屏的客户而言，一个显示屏连接一个支持3D的HDMI输出，另一个显示屏连接第二个仅支持2D的HDMI输出，如图1所示，这可能是有好处的。ADV7850接收3D视频输入，然后将该视频路由至ADV8003 TMDS接收器。ADV8003 IC将3D视频切换到其中一个HDMI发送器以及主视频信号处理器(PVSP)。PVSP将3D图像裁剪并按比例缩放为2D图像，然后将此2D图像路由至其他HDMI Tx——本例中为Tx1。

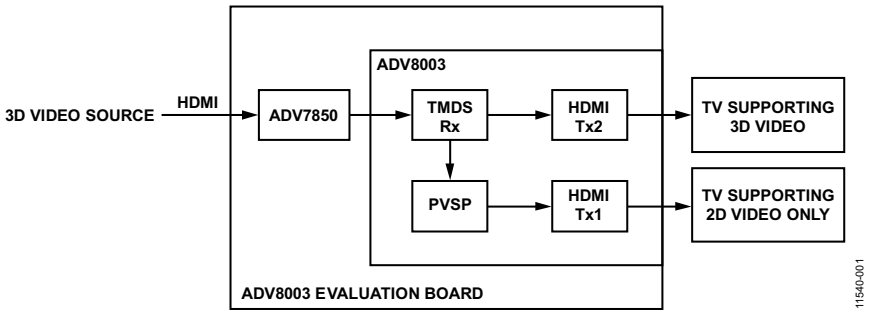

图1. 将3D视频格式裁剪为2D视频格式的示例应用

Advantiv® Advanced Television Solutions by Analog Devices | NatureVue™ Video Upconversion and Enhancement by Analog Devices

目录

修订历史

2013年6月—修订版0：初始版

3D视频处理格式

处理并排格式的(半幅)3D视频流

720P60并排3D视频流由左半幅和右半幅视频图像组成。图4显示ADV8003 Tx2输出的720P60并排视频，显示在视频分析仪上。

HDMI分析仪将3D视频分为其左右两部分内容。每幅左半部和右半部图像都是标准720P60 2D视频水平分辨率的一半，如图2所示。这样可允许3D视频格式由相同数量的像素来描绘，好比用于720P60 2D格式。

脚本1 为720P60并排(半幅)格式部署图1中的3D视频处理(脚本详情请参考脚本1 720P60并排(半幅)视频部分)。3D内容从TMDS Rx直接连接ADV8003的HDMI Tx2。

该脚本利用备用InfoFrame数据包，在HDMI Tx2中实现特定供应商的InfoFrame。发送任何3D视频标准信息时，特定供应商的InfoFrame是必不可少的HDMI数据包。特定供应商的InfoFrame包含HDMI 3D结构，定义3D视频数据的传输格式。

脚本1还在ADV8003中实现了PVSP，对左侧3D图像进行成比例缩放。随后，PVSP将该图像调整为2D 720P60视频。左侧按比例缩放图像是任意选择的，但也可使用右侧图像。2D视频输出至HDMI Tx1。

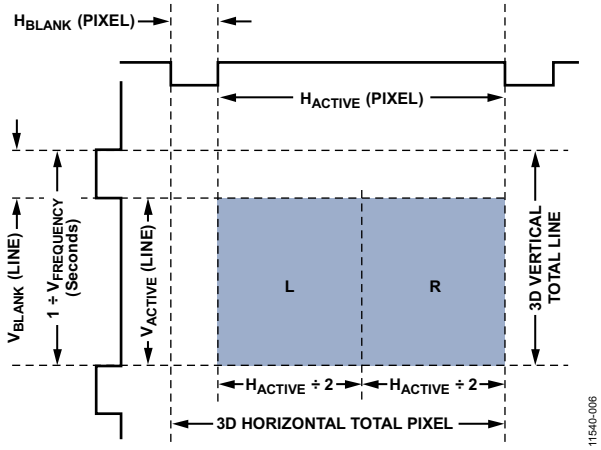

图2. 并排(半幅)结构

处理顶部和底部3D视频

图3显示左右视频图像如何形成1080P60顶部和底部视频。每幅图像均为1080P60 2D视频垂直分辨率的一半。因此，与1080P60 2D格式数量相同的像素可描绘这种3D视频格式。

脚本2 1080P60顶部和底部格式可将1080P60顶部和底部视频从TMDS Rx直接传输至HDMI Tx2和PVSP(脚本详情请参考脚本2 1080P60顶部和底部视频部分)。PVSP按比例缩放3D图像的上半部(左侧有效视频)，并将该图像按比例缩放至1080P60 2D视频。

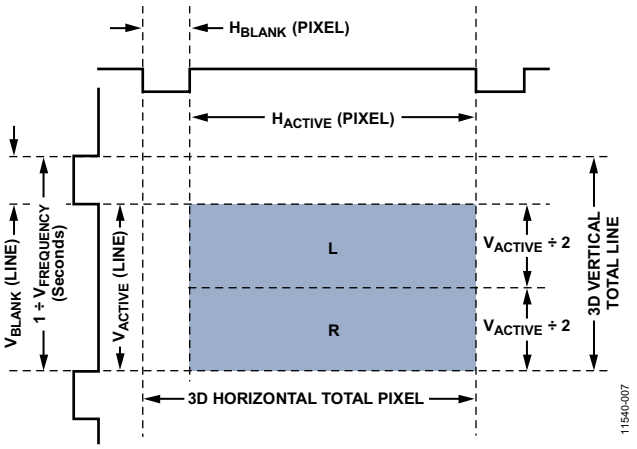

图3. 顶部和底部结构

图4. ADV8003的720P60并排(半幅)输出

处理帧封装3D视频

帧封装3D视频的每一帧都包含左侧视频图像、有效空间以及右侧视频图像。总有效行数是其等效2D视频格式的两倍。3D像素时钟频率同样等于其等效2D视频格式的两倍。

脚本3 1080P24帧封装格式可将1080P24帧封装视频从TMDS Rx直接传输至PVSP和HDMI Tx2(脚本详情请参考脚本3 1080P24帧封装视频部分)。在视频输入模块(VIM)中，PVSP裁剪3D图像的上半部(左侧有效视频)。PVSP自身无法处理1080P24帧封装视频，并且按比例缩放操作必须在输入模块内执行。从PVSP输出的1080P24 2D视频路由至HDMI Tx1。

脚本4 720P60帧封装格式可将720P60帧封装视频从TMDS Rx路由至PVSP和HDMI Tx2(脚本详情请参考脚本4 720P60帧封装视频部分)。PVSP按比例缩放3D图像的上半部(左侧有效视频)，并将该图像以720P60 2D视频格式输出。

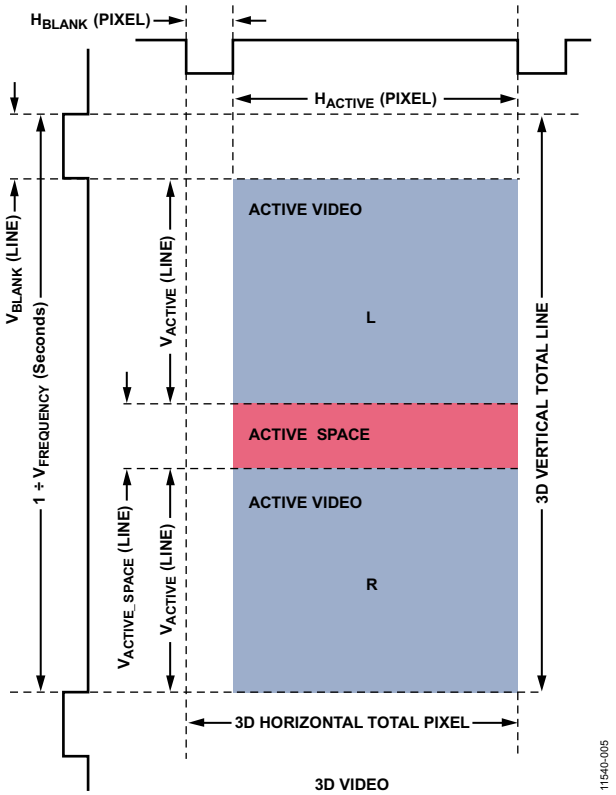

图5. 逐行视频格式的帧封装结构

ADV8003脚本

本节提供的脚本可先将其复制到.py文件，然后在DVP评估软件中直接运行。

1. 复制所需脚本至文本文件，然后保存文件为.py格式。
2. 启动DVP评估软件。
3. 加载ADV8003评估板。
4. 选择**Scripts** > **Run Py Scripts** > **Browse**，然后选择适当的.py文件。

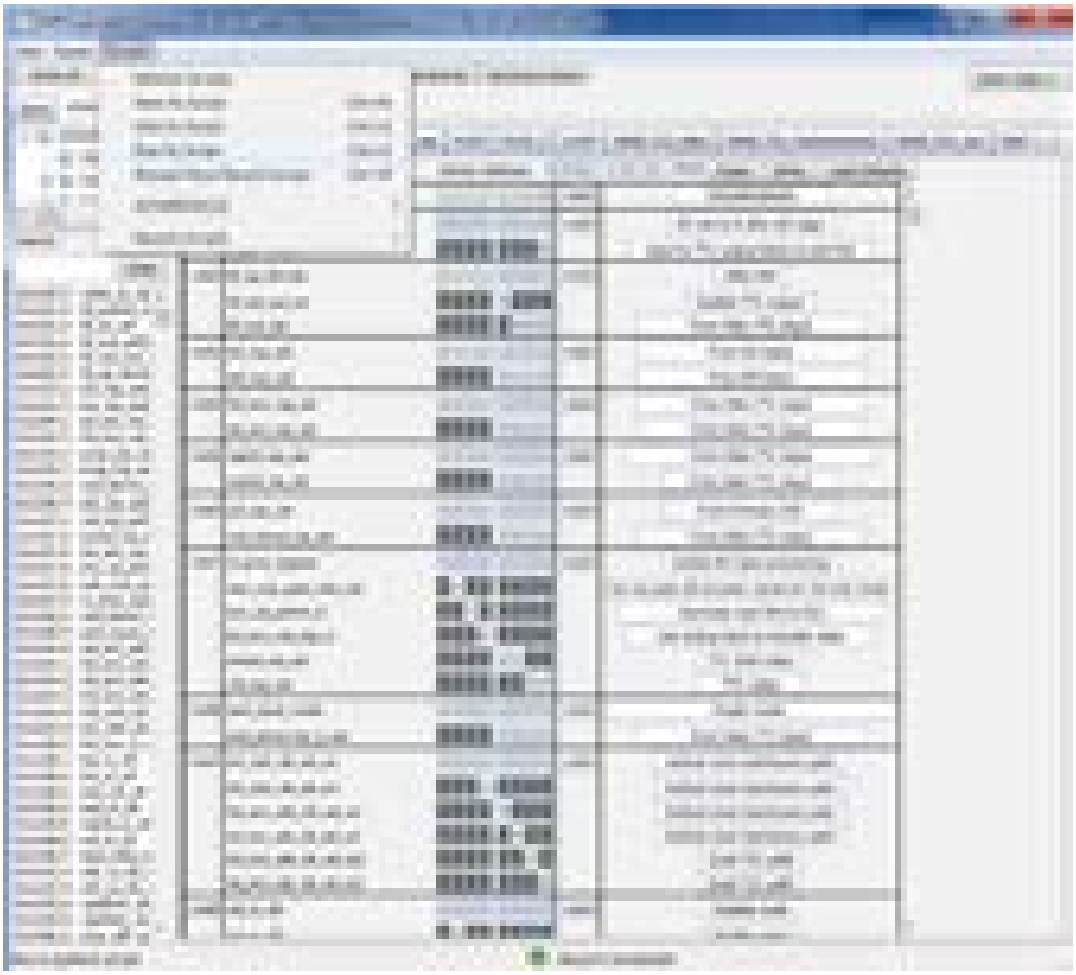

图6. 在DVP评估软件中运行脚本

脚本1 720P60并排(半幅)视频

```
import time
writeReg = topwin.devDriver.writeReg
writeRegs = topwin.devDriver.writeRegs
#*****************************Initial Settings*************************
writeRegs(0x1A, 0x1A5B, [0x22,], [2, 8])
writeRegs(0x1A, 0x1A5F, [0x0,], [2, 8])
writeRegs(0x1A, 0x1A61, [0x6,], [2, 8])
writeRegs(0x1A, 0x1AA0, [0x13,0x1,0x25,0x1D,0x81,0x81,], [2, 8])
writeRegs(0x1A, 0x1AA7, [0x10,0xB4,], [2, 8])
writeRegs(0x1A, 0x1AFE, [0x8,], [2, 8])
writeRegs(0x1A, 0xE0C0, [0xC4,], [2, 8])
```

```
writeRegs(0x1A, 0xE889, [0x3,0x46,0x7A,0x0,], [2, 8])
writeRegs(0x1A, 0xE600, [0x3,0xC5,0xA,0x0,0x3,0xD8,0x6,0x0,0x3,0xEB,0x2,0x0,0x3,0xFD,0xFE,0x0,],
[2, 8])
writeRegs(0x1A, 0xE664, [0x4,0x10,0xFA,0x0,0x4,0x23,0xF6,0x0,0x4,0x36,0xF2,0x0,], [2, 8])
writeRegs(0x1A, 0x1A45, [0x0,0xA8,0x0,0xFB,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,0x88,], [2, 8])
writeRegs(0x1A, 0xE93B, [0x40,], [2, 8])
writeRegs(0x1A, 0xE949, [0xF0,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x88,], [2, 8])
writeRegs(0x1A, 0x1A39, [0xA,], [2, 8])
writeRegs(0x1A, 0xE662, [0x81,], [2, 8])
writeRegs(0x1A, 0x1A9D, [0xFF,0x55,], [2, 8])
#*****************************HDMI Rx Settings******************
writeRegs(0x1A, 0x1A1F, [0x1,], [2, 8])
writeRegs(0x1A, 0x1A07, [0xA4,], [2, 8])
writeRegs(0x1A, 0xE2CB, [0x1,], [2, 8])
writeRegs(0x1A, 0xE23D, [0x10,0x69,0x46,], [2, 8])
writeRegs(0x1A, 0xE24E, [0xCF,0x42,], [2, 8])
writeRegs(0x1A, 0xE257, [0xA3,0x4,], [2, 8])
writeRegs(0x1A, 0xE26f, [0x4,], [2, 8])
writeRegs(0x1A, 0xE275, [0x4,], [2, 8])
writeRegs(0x1A, 0xE283, [0xF0,], [2, 8])
writeRegs(0x1A, 0xE285, [0x10,], [2, 8])
writeRegs(0x1A, 0xE2c0, [0x0,], [2, 8])
writeRegs(0x1A, 0xE21b, [0x14,], [2, 8])
writeRegs(0x1A, 0xE280, [0xA,], [2, 8])
#************************************ADV7850 Settings************************
writeRegs(0x40, 0xFF, [0x80,], [1, 8])
writeRegs(0x40, 0x1B, [0x80,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x40, 0xEC, [0xA0,], [1, 8])
writeRegs(0x40, 0xEB, [0xA8,], [1, 8])
writeRegs(0x40, 0xE7, [0x5C,], [1, 8])
writeRegs(0x40, 0xF1, [0x90,0x94,0x84,0x80,0x7C,], [1, 8])
writeRegs(0x40, 0xF8, [0x4C,0x64,0x6C,0x68,], [1, 8])
writeRegs(0x40, 0xFD, [0x44,0x48,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x68, 0x71, [0x5,], [1, 8])
writeRegs(0x40, 0x0C, [0x40,], [1, 8])
writeRegs(0x40, 0x15, [0x80,], [1, 8])
writeRegs(0x40, 0x1F, [0x10,], [1, 8])
writeRegs(0xA0, 0x01, [0x6,0xF2,], [1, 8])
writeRegs(0xA0, 0xBF, [0x1,], [1, 8])
writeRegs(0x4C, 0xB5, [0x1,], [1, 8])
writeRegs(0x4C, 0xC0, [0xC0,0x98,0x80,0xB0,], [1, 8])
writeRegs(0x64, 0x40, [0x81,], [1, 8])
writeRegs(0x68, 0xCB, [0x1,], [1, 8])
```

```
writeRegs(0x68, 0x6C, [0xA2,], [1, 8])
writeRegs(0x68, 0x3D, [0x10,0x69,0x46,], [1, 8])
writeRegs(0x68, 0x4E, [0xFE,0x8,], [1, 8])
writeRegs(0x68, 0x02, [0xF,], [1, 8])
writeRegs(0x68, 0x57, [0xA3,0x4,], [1, 8])
writeRegs(0x68, 0x6F, [0x4,], [1, 8])
writeRegs(0x68, 0x75, [0x4,], [1, 8])
writeRegs(0x68, 0x83, [0xF0,], [1, 8])
writeRegs(0x68, 0x85, [0x10,], [1, 8])
writeRegs(0x64, 0x74, [0xF,], [1, 8])
writeRegs(0xB8, 0x41, [0x10,], [1, 8])
writeRegs(0xB8, 0x01, [0x0,0x18,0x0,], [1, 8])#  N[19:0])
writeRegs(0xB8, 0x13, [0xFF,], [1, 8])
writeRegs(0xB8, 0x15, [0x20,0x61,], [1, 8])
writeRegs(0xB8, 0x40, [0x80,], [1, 8])
writeRegs(0xB8, 0x4C, [0x4,], [1, 8])
writeRegs(0xB8, 0x55, [0x40,0x8,], [1, 8])
writeRegs(0xB8, 0x96, [0x20,], [1, 8])
writeRegs(0xB8, 0xAF, [0x96,], [1, 8])
writeRegs(0xB8, 0xBA, [0x70,], [1, 8])
writeRegs(0xB8, 0xD0, [0x44,0x3C,], [1, 8])
writeRegs(0xB8, 0xD3, [0x7,], [1, 8])
writeRegs(0xB8, 0xD6, [0x2,], [1, 8])
writeRegs(0xB8, 0xDB, [0xB,], [1, 8])
writeRegs(0xB8, 0xE0, [0x90,0xFC,], [1, 8])
writeRegs(0xB8, 0xE3, [0xD0,], [1, 8])
writeRegs(0xB8, 0xE8, [0xF0,0x1C,0x85,], [1, 8])
writeRegs(0xB8, 0xEC, [0x7C,0x40,0x40,0x41,], [1, 8])
writeRegs(0xB8, 0xF3, [0x1,], [1, 8])
writeRegs(0xB8, 0xF5, [0xCC,0x8,], [1, 8])
writeRegs(0x92, 0x24, [0x43,], [1, 8])
writeRegs(0x40, 0xE3, [0x0,], [1, 8])
#************************************PVSP Script**********************************
writeRegs(0x1A, 0xE828, [0x10,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x4C,], [2, 8])
writeRegs(0x1A, 0xE884, [0x0,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE883, [0x80,], [2, 8])
writeRegs(0x1A, 0xE881, [0x4,0x4,], [2, 8])
writeRegs(0x1A, 0xE86C, [0x0,], [2, 8])
writeRegs(0x1A, 0xE935, [0xC6,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,], [2, 8])
writeRegs(0x1A, 0xE84E, [0x1,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE828, [0x11,], [2, 8])
writeRegs(0x1A, 0xE828, [0x13,], [2, 8])
writeRegs(0x1A, 0xE828, [0x17,], [2, 8])
```

```
#************************************Tx1 Delay Settings*************************
writeRegs(0x1A, 0xF324, [0x0,], [2, 8])
#************************************Tx1 Main Settings**************************
writeRegs(0x1A, 0xEC41, [0x10,], [2, 8])
writeRegs(0x1A, 0xEC01, [0x0,0x60,0x0,], [2, 8])#  Default 0x00, N[19:0])
writeRegs(0x1A, 0xEC13, [0xFF,], [2, 8])
writeRegs(0x1A, 0xEC15, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xEC40, [0x80,], [2, 8])
writeRegs(0x1A, 0xEC4C, [0x6,], [2, 8])
writeRegs(0x1A, 0xEC55, [0x40,0x8,], [2, 8])
writeRegs(0x1A, 0xEC96, [0x20,], [2, 8])
writeRegs(0x1A, 0xECAF, [0x96,], [2, 8])
writeRegs(0x1A, 0xECBA, [0x70,], [2, 8])
writeRegs(0x1A, 0xECD0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xECD3, [0x7,], [2, 8])
writeRegs(0x1A, 0xECD6, [0x2,], [2, 8])
writeRegs(0x1A, 0xECDB, [0xB,], [2, 8])
writeRegs(0x1A, 0xECE0, [0x90,0xFC,], [2, 8])
writeRegs(0x1A, 0xECE3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xECE8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xECEA, [0x85,], [2, 8])
writeRegs(0x1A, 0xECEC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xECF3, [0x1,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xECDA, [0x40,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xECE9, [0x74,], [2, 8])
#************************************Tx1 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xEC4C, [0x4,], [2, 8])
#************************************Tx2 Delay Settings*************************
writeRegs(0x1A, 0xFB24, [0x0,], [2, 8])
#************************************Tx2 Main Settings**************************
writeRegs(0x1A, 0xF441, [0x10,], [2, 8])
writeRegs(0x1A, 0xF401, [0x0,0x60,0x0,], [2, 8])
writeRegs(0x1A, 0xF413, [0xFF,], [2, 8])
writeRegs(0x1A, 0xF415, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xF440, [0x80,], [2, 8])
writeRegs(0x1A, 0xF44C, [0x6,], [2, 8])
writeRegs(0x1A, 0xF455, [0x40,0x8,], [2, 8])
writeRegs(0x1A, 0xF496, [0x20,], [2, 8])
writeRegs(0x1A, 0xF4AF, [0x96,], [2, 8])
writeRegs(0x1A, 0xF4BA, [0x70,], [2, 8])
writeRegs(0x1A, 0xF4D0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xF4D3, [0x7,], [2, 8])
writeRegs(0x1A, 0xF4D6, [0x2,], [2, 8])
writeRegs(0x1A, 0xF4DB, [0xB,], [2, 8])
writeRegs(0x1A, 0xF4E0, [0x90,0xFC,], [2, 8])
```

```
writeRegs(0x1A, 0xF4E3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xF4E8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xF4EA, [0x85,], [2, 8])
writeRegs(0x1A, 0xF4EC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xF4F3, [0x1,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xF4DA, [0x40,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xF4E9, [0x74,], [2, 8])
#*************************************Tx2 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xF44C, [0x4,], [2, 8])
#*****************Enable Tx2 Vendor Specific Infoframe for 3D SBS Pass Through**************
writeRegs(0x1A, 0xFAC0, [0x81,], [2, 8]) #Header
writeRegs(0x1A, 0xFAC1, [0x01,], [2, 8])
writeRegs(0x1A, 0xFAC2, [0x06,], [2, 8]) #   Length
writeRegs(0x1A, 0xFAC3, [0xA9,], [2, 8])
writeRegs(0x1A, 0xFAC4, [0x03,], [2, 8])
writeRegs(0x1A, 0xFAC5, [0x0C,], [2, 8])
writeRegs(0x1A, 0xFAC6, [0x00,], [2, 8])
writeRegs(0x1A, 0xFAC7, [0x40,], [2, 8]) #3D Format Indication
writeRegs(0x1A, 0xFAC8, [0x80,], [2, 8]) #Side by Side Half - 3D_Structure
writeRegs(0x1A, 0xF440, [0x81,], [2, 8]) #Enables the General Control packet and Spare Packet 1
#*******************************Enable Cropping*************************
writeRegs(0x1A, 0xE83C, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE83D, [0x01,], [2, 8]) #
writeRegs(0x1A, 0xE83E, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE83F, [0x01,], [2, 8]) #
writeRegs(0x1A, 0xE840, [0x02,], [2, 8]) #
writeRegs(0x1A, 0xE841, [0x80,], [2, 8]) #
writeRegs(0x1A, 0xE842, [0x02,], [2, 8]) #
writeRegs(0x1A, 0xE843, [0xD0,], [2, 8]) #
writeRegs(0x1A, 0xE883, [0x90,], [2, 8]) #
#Muxing
writeRegs(0x1A, 0x1A05, [0x03,], [2, 8]) #PVSP from RX input
writeRegs(0x1A, 0x1A03, [0x16,], [2, 8]) #TX1 from PVSP, TX2 from HDMI RX
writeRegs(0x1A, 0xEC9F, [0x30,], [2, 8]) #
writeRegs(0x1A, 0xF49F, [0x30,], [2, 8]) #
```

脚本2 1080P60顶部和底部视频

```
import time
writeReg = topwin.devDriver.writeReg
writeRegs = topwin.devDriver.writeRegs
#****************************Initial Settings************************
writeRegs(0x1A, 0x1A5B, [0x22,], [2, 8])
writeRegs(0x1A, 0x1A5F, [0x0,], [2, 8])
writeRegs(0x1A, 0x1A61, [0x6,], [2, 8])
writeRegs(0x1A, 0x1AA0, [0x13,0x1,0x25,0x1D,0x81,0x81,], [2, 8])
writeRegs(0x1A, 0x1AA7, [0x10,0xB4,], [2, 8])
```

```
writeRegs(0x1A, 0x1AFE, [0x8,], [2, 8])
writeRegs(0x1A, 0xE0C0, [0xC4,], [2, 8])
writeRegs(0x1A, 0xE889, [0x3,0x46,0x7A,0x0,], [2, 8])
writeRegs(0x1A, 0xE600, [0x3,0xC5,0xA,0x0,0x3,0xD8,0x6,0x0,0x3,0xEB,0x2,0x0,0x3,0xFD,0xFE,0x0,],
[2, 8])
writeRegs(0x1A, 0xE664, [0x4,0x10,0xFA,0x0,0x4,0x23,0xF6,0x0,0x4,0x36,0xF2,0x0,], [2, 8])
writeRegs(0x1A, 0x1A45, [0x0,0xA8,0x0,0xFB,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,0x88,], [2, 8])
writeRegs(0x1A, 0xE93B, [0x40,], [2, 8])
writeRegs(0x1A, 0xE949, [0xF0,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x88,], [2, 8])
writeRegs(0x1A, 0x1A39, [0xA,], [2, 8])
writeRegs(0x1A, 0xE662, [0x81,], [2, 8])
writeRegs(0x1A, 0x1A9D, [0xFF,0x55,], [2, 8])
#*****************************HDMI Rx Settings******************
writeRegs(0x1A, 0x1A1F, [0x1,], [2, 8])
writeRegs(0x1A, 0x1A07, [0xA4,], [2, 8])
writeRegs(0x1A, 0xE2CB, [0x1,], [2, 8])
writeRegs(0x1A, 0xE23D, [0x10,0x69,0x46,], [2, 8])
writeRegs(0x1A, 0xE24E, [0xCF,0x42,], [2, 8])
writeRegs(0x1A, 0xE257, [0xA3,0x4,], [2, 8])
writeRegs(0x1A, 0xE26f, [0x4,], [2, 8])
writeRegs(0x1A, 0xE275, [0x4,], [2, 8])
writeRegs(0x1A, 0xE283, [0xF0,], [2, 8])
writeRegs(0x1A, 0xE285, [0x10,], [2, 8])
writeRegs(0x1A, 0xE2c0, [0x0,], [2, 8])
writeRegs(0x1A, 0xE21b, [0x14,], [2, 8])
writeRegs(0x1A, 0xE280, [0xA,], [2, 8])
#************************************ADV7850 Settings************************
writeRegs(0x40, 0xFF, [0x80,], [1, 8])
writeRegs(0x40, 0x1B, [0x80,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x40, 0xEC, [0xA0,], [1, 8])
writeRegs(0x40, 0xEB, [0xA8,], [1, 8])
writeRegs(0x40, 0xE7, [0x5C,], [1, 8])
writeRegs(0x40, 0xF1, [0x90,0x94,0x84,0x80,0x7C,], [1, 8])
writeRegs(0x40, 0xF8, [0x4C,0x64,0x6C,0x68,], [1, 8])
writeRegs(0x40, 0xFD, [0x44,0x48,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x68, 0x71, [0x5,], [1, 8])
writeRegs(0x40, 0x0C, [0x40,], [1, 8])
writeRegs(0x40, 0x15, [0x80,], [1, 8])
writeRegs(0x40, 0x1F, [0x10,], [1, 8])
writeRegs(0xA0, 0x01, [0x6,0xF2,], [1, 8])
writeRegs(0xA0, 0xBF, [0x1,], [1, 8])
writeRegs(0x4C, 0xB5, [0x1,], [1, 8])
writeRegs(0x4C, 0xC0, [0xC0,0x98,0x80,0xB0,], [1, 8])
```

```
writeRegs(0x64, 0x40, [0x81,], [1, 8])
writeRegs(0x68, 0xCB, [0x1,], [1, 8])
writeRegs(0x68, 0x6C, [0xA2,], [1, 8])
writeRegs(0x68, 0x3D, [0x10,0x69,0x46,], [1, 8])
writeRegs(0x68, 0x4E, [0xFE,0x8,], [1, 8])
writeRegs(0x68, 0x02, [0xF,], [1, 8])
writeRegs(0x68, 0x57, [0xA3,0x4,], [1, 8])
writeRegs(0x68, 0x6F, [0x4,], [1, 8])
writeRegs(0x68, 0x75, [0x4,], [1, 8])
writeRegs(0x68, 0x83, [0xF0,], [1, 8])
writeRegs(0x68, 0x85, [0x10,], [1, 8])
writeRegs(0x64, 0x74, [0xF,], [1, 8])
writeRegs(0xB8, 0x41, [0x10,], [1, 8])
writeRegs(0xB8, 0x01, [0x0,0x18,0x0,], [1, 8])#  N[19:0])
writeRegs(0xB8, 0x13, [0xFF,], [1, 8])
writeRegs(0xB8, 0x15, [0x20,0x61,], [1, 8])
writeRegs(0xB8, 0x40, [0x80,], [1, 8])
writeRegs(0xB8, 0x4C, [0x4,], [1, 8])
writeRegs(0xB8, 0x55, [0x40,0x8,], [1, 8])
writeRegs(0xB8, 0x96, [0x20,], [1, 8])
writeRegs(0xB8, 0xAF, [0x96,], [1, 8])
writeRegs(0xB8, 0xBA, [0x70,], [1, 8])
writeRegs(0xB8, 0xD0, [0x44,0x3C,], [1, 8])
writeRegs(0xB8, 0xD3, [0x7,], [1, 8])
writeRegs(0xB8, 0xD6, [0x2,], [1, 8])
writeRegs(0xB8, 0xDB, [0xB,], [1, 8])
writeRegs(0xB8, 0xE0, [0x90,0xFC,], [1, 8])
writeRegs(0xB8, 0xE3, [0xD0,], [1, 8])
writeRegs(0xB8, 0xE8, [0xF0,0x1C,0x85,], [1, 8])
writeRegs(0xB8, 0xEC, [0x7C,0x40,0x40,0x41,], [1, 8])
writeRegs(0xB8, 0xF3, [0x1,], [1, 8])
writeRegs(0xB8, 0xF5, [0xCC,0x8,], [1, 8])
writeRegs(0x92, 0x24, [0x43,], [1, 8])
writeRegs(0x40, 0xE3, [0x0,], [1, 8])
#******************PVSP Script for Scaling 1080P60 3D to 1080P60 2D*******************
writeRegs(0x1A, 0xE828, [0x10,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x4C,], [2, 8])
writeRegs(0x1A, 0xE884, [0x0,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE883, [0x80,], [2, 8])
writeRegs(0x1A, 0xE881, [0x10,0x10,], [2, 8])
writeRegs(0x1A, 0xE86C, [0x0,], [2, 8])
writeRegs(0x1A, 0xE935, [0xC6,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,], [2, 8])
writeRegs(0x1A, 0xE84E, [0x1,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE828, [0x11,], [2, 8])
```

```
writeRegs(0x1A, 0xE828, [0x13,], [2, 8])
writeRegs(0x1A, 0xE828, [0x17,], [2, 8])
#******************Tx1 Delay Settings for Greater than 130 MHz*****************
writeRegs(0x1A, 0xF324, [0x80,], [2, 8])
#************************************Tx1 Main Settings**************************
writeRegs(0x1A, 0xEC41, [0x10,], [2, 8])
writeRegs(0x1A, 0xEC01, [0x0,0x60,0x0,], [2, 8])#  Default 0x00, N[19:0])
writeRegs(0x1A, 0xEC13, [0xFF,], [2, 8])
writeRegs(0x1A, 0xEC15, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xEC40, [0x80,], [2, 8])
writeRegs(0x1A, 0xEC4C, [0x6,], [2, 8])
writeRegs(0x1A, 0xEC55, [0x40,0x8,], [2, 8])
writeRegs(0x1A, 0xEC96, [0x20,], [2, 8])
writeRegs(0x1A, 0xECAF, [0x96,], [2, 8])
writeRegs(0x1A, 0xECBA, [0x70,], [2, 8])
writeRegs(0x1A, 0xECD0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xECD3, [0x7,], [2, 8])
writeRegs(0x1A, 0xECD6, [0x2,], [2, 8])
writeRegs(0x1A, 0xECDB, [0xB,], [2, 8])
writeRegs(0x1A, 0xECE0, [0x90,0xFC,], [2, 8])
writeRegs(0x1A, 0xECE3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xECE8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xECEA, [0x85,], [2, 8])
writeRegs(0x1A, 0xECEC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xECF3, [0x1,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xECDA, [0x40,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xECE9, [0x74,], [2, 8])
#************************************Tx1 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xEC4C, [0x4,], [2, 8])
#******************Tx2 Delay Settings for Greater than 130 MHz*****************
writeRegs(0x1A, 0xFB24, [0x80,], [2, 8])
#************************************Tx2 Main Settings**************************
writeRegs(0x1A, 0xF441, [0x10,], [2, 8])
writeRegs(0x1A, 0xF401, [0x0,0x60,0x0,], [2, 8])#  Default 0x00, N[19:0])
writeRegs(0x1A, 0xF413, [0xFF,], [2, 8])
writeRegs(0x1A, 0xF415, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xF440, [0x80,], [2, 8])
writeRegs(0x1A, 0xF44C, [0x6,], [2, 8])
writeRegs(0x1A, 0xF455, [0x40,0x8,], [2, 8])
writeRegs(0x1A, 0xF496, [0x20,], [2, 8])
writeRegs(0x1A, 0xF4AF, [0x96,], [2, 8])
writeRegs(0x1A, 0xF4BA, [0x70,], [2, 8])
writeRegs(0x1A, 0xF4D0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xF4D3, [0x7,], [2, 8])
writeRegs(0x1A, 0xF4D6, [0x2,], [2, 8])
```

```
writeRegs(0x1A, 0xF4DB, [0xB,], [2, 8])
writeRegs(0x1A, 0xF4E0, [0x90,0xFC,], [2, 8])
writeRegs(0x1A, 0xF4E3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xF4E8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xF4EA, [0x85,], [2, 8])
writeRegs(0x1A, 0xF4EC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xF4F3, [0x1,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xF4DA, [0x40,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xF4E9, [0x74,], [2, 8])
#*************************************Tx2 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xF44C, [0x4,], [2, 8])

#***************Enable Tx2 Vendor Specific Infoframe for 3D Top-Bottom Pass Through*************
writeRegs(0x1A, 0xFAC0, [0x81,], [2, 8]) #Header
writeRegs(0x1A, 0xFAC1, [0x01,], [2, 8])
writeRegs(0x1A, 0xFAC2, [0x05,], [2, 8]) #   Length
writeRegs(0x1A, 0xFAC3, [0xCa,], [2, 8])
writeRegs(0x1A, 0xFAC4, [0x03,], [2, 8])
writeRegs(0x1A, 0xFAC5, [0x0C,], [2, 8])
writeRegs(0x1A, 0xFAC6, [0x00,], [2, 8])
writeRegs(0x1A, 0xFAC7, [0x40,], [2, 8]) #
writeRegs(0x1A, 0xFAC8, [0x60,], [2, 8]) #
writeRegs(0x1A, 0xF40, [0x81,], [2, 8]) #
#*********************************Enable 2D Cropping*************************
writeRegs(0x1A, 0xE83C, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE83D, [0x01,], [2, 8]) #
writeRegs(0x1A, 0xE83E, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE83F, [0x01,], [2, 8]) #
writeRegs(0x1A, 0xE840, [0x07,], [2, 8]) #
writeRegs(0x1A, 0xE841, [0x80,], [2, 8]) #
writeRegs(0x1A, 0xE842, [0x02,], [2, 8]) #
writeRegs(0x1A, 0xE843, [0x1C,], [2, 8]) #
writeRegs(0x1A, 0xE883, [0x90,], [2, 8]) #
#Muxing
writeRegs(0x1A, 0x1A05, [0x03,], [2, 8]) #PVSP from RX input
writeRegs(0x1A, 0x1A03, [0x16,], [2, 8]) #TX1 from PVSP, TX2 from HDMI RX
writeRegs(0x1A, 0xEC9F, [0x30,], [2, 8]) #
writeRegs(0x1A, 0xF49F, [0x30,], [2, 8]) #
```

脚本3 1080P24帧封装视频

```
import time
writeReg = topwin.devDriver.writeReg
writeRegs = topwin.devDriver.writeRegs
#***************************Initial Settings************************
writeRegs(0x1A, 0x1A5B, [0x22,], [2, 8])
writeRegs(0x1A, 0x1A5F, [0x0,], [2, 8])
```

```
writeRegs(0x1A, 0x1A61, [0x6,], [2, 8])
writeRegs(0x1A, 0x1AA0, [0x13,0x1,0x25,0x1D,0x81,0x81,], [2, 8])
writeRegs(0x1A, 0x1AA7, [0x10,0xB4,], [2, 8])
writeRegs(0x1A, 0x1AFE, [0x8,], [2, 8])
writeRegs(0x1A, 0xE0C0, [0xC4,], [2, 8])
writeRegs(0x1A, 0xE889, [0x3,0x46,0x7A,0x0,], [2, 8])
writeRegs(0x1A, 0xE600, [0x3,0xC5,0xA,0x0,0x3,0xD8,0x6,0x0,0x3,0xEB,0x2,0x0,0x3,0xFD,0xFE,0x0,],
[2, 8])
writeRegs(0x1A, 0xE664, [0x4,0x10,0xFA,0x0,0x4,0x23,0xF6,0x0,0x4,0x36,0xF2,0x0,], [2, 8])
writeRegs(0x1A, 0x1A45, [0x0,0xA8,0x0,0xFB,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,0x88,], [2, 8])
writeRegs(0x1A, 0xE93B, [0x40,], [2, 8])
writeRegs(0x1A, 0xE949, [0xF0,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x88,], [2, 8])
writeRegs(0x1A, 0x1A39, [0xA,], [2, 8])
writeRegs(0x1A, 0xE662, [0x81,], [2, 8])
writeRegs(0x1A, 0x1A9D, [0xFF,0x55,], [2, 8])
#*****************************HDMI Rx Settings******************
writeRegs(0x1A, 0x1A1F, [0x1,], [2, 8])
writeRegs(0x1A, 0x1A07, [0xA4,], [2, 8])
writeRegs(0x1A, 0xE2CB, [0x1,], [2, 8])
writeRegs(0x1A, 0xE23D, [0x10,0x69,0x46,], [2, 8])
writeRegs(0x1A, 0xE24E, [0xCF,0x42,], [2, 8])
writeRegs(0x1A, 0xE257, [0xA3,0x4,], [2, 8])
writeRegs(0x1A, 0xE26f, [0x4,], [2, 8])
writeRegs(0x1A, 0xE275, [0x4,], [2, 8])
writeRegs(0x1A, 0xE283, [0xF0,], [2, 8])
writeRegs(0x1A, 0xE285, [0x10,], [2, 8])
writeRegs(0x1A, 0xE2c0, [0x0,], [2, 8])
writeRegs(0x1A, 0xE21b, [0x14,], [2, 8])
writeRegs(0x1A, 0xE280, [0xA,], [2, 8])
#************************************ADV7850 Settings************************
writeRegs(0x40, 0xFF, [0x80,], [1, 8])
writeRegs(0x40, 0x1B, [0x80,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x40, 0xEC, [0xA0,], [1, 8])
writeRegs(0x40, 0xEB, [0xA8,], [1, 8])
writeRegs(0x40, 0xE7, [0x5C,], [1, 8])
writeRegs(0x40, 0xF1, [0x90,0x94,0x84,0x80,0x7C,], [1, 8])
writeRegs(0x40, 0xF8, [0x4C,0x64,0x6C,0x68,], [1, 8])
writeRegs(0x40, 0xFD, [0x44,0x48,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x68, 0x71, [0x5,], [1, 8])
writeRegs(0x40, 0x0C, [0x40,], [1, 8])
writeRegs(0x40, 0x15, [0x80,], [1, 8])
writeRegs(0x40, 0x1F, [0x10,], [1, 8])
writeRegs(0xA0, 0x01, [0x6,0xF2,], [1, 8])
```

```
writeRegs(0xA0, 0xBF, [0x1,], [1, 8])
writeRegs(0x4C, 0xB5, [0x1,], [1, 8])
writeRegs(0x4C, 0xC0, [0xC0,0x98,0x80,0xB0,], [1, 8])
writeRegs(0x64, 0x40, [0x81,], [1, 8])
writeRegs(0x68, 0xCB, [0x1,], [1, 8])
writeRegs(0x68, 0x6C, [0xA2,], [1, 8])
writeRegs(0x68, 0x3D, [0x10,0x69,0x46,], [1, 8])
writeRegs(0x68, 0x4E, [0xFE,0x8,], [1, 8])
writeRegs(0x68, 0x02, [0xF,], [1, 8])
writeRegs(0x68, 0x57, [0xA3,0x4,], [1, 8])
writeRegs(0x68, 0x6F, [0x4,], [1, 8])
writeRegs(0x68, 0x75, [0x4,], [1, 8])
writeRegs(0x68, 0x83, [0xF0,], [1, 8])
writeRegs(0x68, 0x85, [0x10,], [1, 8])
writeRegs(0x64, 0x74, [0xF,], [1, 8])
writeRegs(0xB8, 0x41, [0x10,], [1, 8])
writeRegs(0xB8, 0x01, [0x0,0x18,0x0,], [1, 8])#  N[19:0])
writeRegs(0xB8, 0x13, [0xFF,], [1, 8])
writeRegs(0xB8, 0x15, [0x20,0x61,], [1, 8])
writeRegs(0xB8, 0x40, [0x80,], [1, 8])
writeRegs(0xB8, 0x4C, [0x4,], [1, 8])
writeRegs(0xB8, 0x55, [0x40,0x8,], [1, 8])
writeRegs(0xB8, 0x96, [0x20,], [1, 8])
writeRegs(0xB8, 0xAF, [0x96,], [1, 8])
writeRegs(0xB8, 0xBA, [0x70,], [1, 8])
writeRegs(0xB8, 0xD0, [0x44,0x3C,], [1, 8])
writeRegs(0xB8, 0xD3, [0x7,], [1, 8])
writeRegs(0xB8, 0xD6, [0x2,], [1, 8])
writeRegs(0xB8, 0xDB, [0xB,], [1, 8])
writeRegs(0xB8, 0xE0, [0x90,0xFC,], [1, 8])
writeRegs(0xB8, 0xE3, [0xD0,], [1, 8])
writeRegs(0xB8, 0xE8, [0xF0,0x1C,0x85,], [1, 8])
writeRegs(0xB8, 0xEC, [0x7C,0x40,0x40,0x41,], [1, 8])
writeRegs(0xB8, 0xF3, [0x1,], [1, 8])
writeRegs(0xB8, 0xF5, [0xCC,0x8,], [1, 8])
writeRegs(0x92, 0x24, [0x43,], [1, 8])
writeRegs(0x40, 0xE3, [0x0,], [1, 8])
#********************PVSP Script for Scaling 1080P24 3D to 1080P24 2D***********************
writeRegs(0x1A, 0xE828, [0x10,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x4C,], [2, 8])
writeRegs(0x1A, 0xE884, [0x0,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE883, [0x80,], [2, 8])
writeRegs(0x1A, 0xE881, [0x20,0x20,], [2, 8])
writeRegs(0x1A, 0xE86C, [0x0,], [2, 8])
writeRegs(0x1A, 0xE935, [0xC6,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,], [2, 8])
```

```
writeRegs(0x1A, 0xE84E, [0x1,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE828, [0x11,], [2, 8])
writeRegs(0x1A, 0xE828, [0x13,], [2, 8])
writeRegs(0x1A, 0xE828, [0x17,], [2, 8])
#************************Tx1 Delay Settings for Less than 130 MHz***********************
writeRegs = topwin.devDriver.writeRegs
writeRegs(0x1A, 0xF324, [0x0,], [2, 8])
#************************************Tx1 Main Settings*************************
writeRegs(0x1A, 0xEC41, [0x10,], [2, 8])
writeRegs(0x1A, 0xEC01, [0x0,0x60,0x0,], [2, 8])#  Default 0x00, N[19:0])
writeRegs(0x1A, 0xEC13, [0xFF,], [2, 8])
writeRegs(0x1A, 0xEC15, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xEC40, [0x80,], [2, 8])
writeRegs(0x1A, 0xEC4C, [0x6,], [2, 8])
writeRegs(0x1A, 0xEC55, [0x40,0x8,], [2, 8])
writeRegs(0x1A, 0xEC96, [0x20,], [2, 8])
writeRegs(0x1A, 0xECAF, [0x96,], [2, 8])
writeRegs(0x1A, 0xECBA, [0x70,], [2, 8])
writeRegs(0x1A, 0xECD0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xECD3, [0x7,], [2, 8])
writeRegs(0x1A, 0xECD6, [0x2,], [2, 8])
writeRegs(0x1A, 0xECDB, [0xB,], [2, 8])
writeRegs(0x1A, 0xECE0, [0x90,0xFC,], [2, 8])
writeRegs(0x1A, 0xECE3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xECE8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xECEA, [0x85,], [2, 8])
writeRegs(0x1A, 0xECEC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xECF3, [0x1,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xECDA, [0x40,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xECE9, [0x74,], [2, 8])
#************************************Tx1 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xEC4C, [0x4,], [2, 8])

#***************************Tx2 Delay Settings for Less than 130 MHz*********************
writeRegs = topwin.devDriver.writeRegs
writeRegs(0x1A, 0xFB24, [0x0,], [2, 8])
#************************************Tx1 Main Settings*************************
writeRegs(0x1A, 0xF441, [0x10,], [2, 8])
writeRegs(0x1A, 0xF401, [0x0,0x60,0x0,], [2, 8])#  Default 0x00, N[19:0])
writeRegs(0x1A, 0xF413, [0xFF,], [2, 8])
writeRegs(0x1A, 0xF415, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xF440, [0x80,], [2, 8])
writeRegs(0x1A, 0xF44C, [0x6,], [2, 8])
writeRegs(0x1A, 0xF455, [0x40,0x8,], [2, 8])
```

```
writeRegs(0x1A, 0xF496, [0x20,], [2, 8])
writeRegs(0x1A, 0xF4AF, [0x96,], [2, 8])
writeRegs(0x1A, 0xF4BA, [0x70,], [2, 8])
writeRegs(0x1A, 0xF4D0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xF4D3, [0x7,], [2, 8])
writeRegs(0x1A, 0xF4D6, [0x2,], [2, 8])
writeRegs(0x1A, 0xF4DB, [0xB,], [2, 8])
writeRegs(0x1A, 0xF4E0, [0x90,0xFC,], [2, 8])
writeRegs(0x1A, 0xF4E3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xF4E8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xF4EA, [0x85,], [2, 8])
writeRegs(0x1A, 0xF4EC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xF4F3, [0x1,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xF4DA, [0x40,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xF4E9, [0x74,], [2, 8])
#************************************Tx1 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xF44C, [0x4,], [2, 8])
#************Enable Tx2 Vendor Specific Infoframe for 3D Frame Pack Pass Through*************
writeRegs(0x1A, 0xFAC0, [0x81,], [2, 8]) #
writeRegs(0x1A, 0xFAC1, [0x01,], [2, 8])
writeRegs(0x1A, 0xFAC2, [0x05,], [2, 8]) #
writeRegs(0x1A, 0xFAC3, [0x2A,], [2, 8]) #
writeRegs(0x1A, 0xFAC4, [0x03,], [2, 8]) #
writeRegs(0x1A, 0xFAC5, [0x0C,], [2, 8])
writeRegs(0x1A, 0xFAC6, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xFAC7, [0x40,], [2, 8]) #
writeRegs(0x1A, 0xFAC8, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xF440, [0x81,], [2, 8]) #
#*************************Enable 3D to 2D Cropping for Frame Pack Mode*******************
writeRegs(0x1A, 0xE832, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE833, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE834, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE835, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE836, [0x07,], [2, 8]) #
writeRegs(0x1A, 0xE837, [0x80,], [2, 8]) #
writeRegs(0x1A, 0xE838, [0x04,], [2, 8]) #
writeRegs(0x1A, 0xE839, [0x38,], [2, 8]) #
writeRegs(0x1A, 0xE883, [0xC0,], [2, 8]) #
#Muxing
writeRegs(0x1A, 0x1A05, [0x03,], [2, 8]) #PVSP from RX input
writeRegs(0x1A, 0x1A03, [0x16,], [2, 8]) #TX1 from PVSP, TX2 from HDMI RX
writeRegs(0x1A, 0xEC9F, [0x30,], [2, 8]) #
writeRegs(0x1A, 0xF49F, [0x30,], [2, 8]) #
```

脚本4 720P60帧封装视频

```
import time
writeReg = topwin.devDriver.writeReg
writeRegs = topwin.devDriver.writeRegs
#*****************************Initial Settings*************************
writeRegs(0x1A, 0x1A5B, [0x22,], [2, 8])
writeRegs(0x1A, 0x1A5F, [0x0,], [2, 8])
writeRegs(0x1A, 0x1A61, [0x6,], [2, 8])
writeRegs(0x1A, 0x1AA0, [0x13,0x1,0x25,0x1D,0x81,0x81,], [2, 8])
writeRegs(0x1A, 0x1AA7, [0x10,0xB4,], [2, 8])
writeRegs(0x1A, 0x1AFE, [0x8,], [2, 8])
writeRegs(0x1A, 0xE0C0, [0xC4,], [2, 8])
writeRegs(0x1A, 0xE889, [0x3,0x46,0x7A,0x0,], [2, 8])
writeRegs(0x1A, 0xE600, [0x3,0xC5,0xA,0x0,0x3,0xD8,0x6,0x0,0x3,0xEB,0x2,0x0,0x3,0xFD,0xFE,0x0,],
[2, 8])
writeRegs(0x1A, 0xE664, [0x4,0x10,0xFA,0x0,0x4,0x23,0xF6,0x0,0x4,0x36,0xF2,0x0,], [2, 8])
writeRegs(0x1A, 0x1A45, [0x0,0xA8,0x0,0xFB,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,0x88,], [2, 8])
writeRegs(0x1A, 0xE93B, [0x40,], [2, 8])
writeRegs(0x1A, 0xE949, [0xF0,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x88,], [2, 8])
writeRegs(0x1A, 0x1A39, [0xA,], [2, 8])
writeRegs(0x1A, 0xE662, [0x81,], [2, 8])
writeRegs(0x1A, 0x1A9D, [0xFF,0x55,], [2, 8])
#*****************************HDMI Rx Settings*****************
writeRegs(0x1A, 0x1A1F, [0x1,], [2, 8])
writeRegs(0x1A, 0x1A07, [0xA4,], [2, 8])
writeRegs(0x1A, 0xE2CB, [0x1,], [2, 8])
writeRegs(0x1A, 0xE23D, [0x10,0x69,0x46,], [2, 8])
writeRegs(0x1A, 0xE24E, [0xCF,0x42,], [2, 8])
writeRegs(0x1A, 0xE257, [0xA3,0x4,], [2, 8])
writeRegs(0x1A, 0xE26f, [0x4,], [2, 8])
writeRegs(0x1A, 0xE275, [0x4,], [2, 8])
writeRegs(0x1A, 0xE283, [0xF0,], [2, 8])
writeRegs(0x1A, 0xE285, [0x10,], [2, 8])
writeRegs(0x1A, 0xE2c0, [0x0,], [2, 8])
writeRegs(0x1A, 0xE21b, [0x14,], [2, 8])
writeRegs(0x1A, 0xE280, [0xA,], [2, 8])
#************************************ADV7850 Settings************************
writeRegs(0x40, 0xFF, [0x80,], [1, 8])
writeRegs(0x40, 0x1B, [0x80,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x40, 0xEC, [0xA0,], [1, 8])
writeRegs(0x40, 0xEB, [0xA8,], [1, 8])
writeRegs(0x40, 0xE7, [0x5C,], [1, 8])
writeRegs(0x40, 0xF1, [0x90,0x94,0x84,0x80,0x7C,], [1, 8])
writeRegs(0x40, 0xF8, [0x4C,0x64,0x6C,0x68,], [1, 8])
```

```
writeRegs(0x40, 0xFD, [0x44,0x48,], [1, 8])
writeRegs(0x40, 0xE3, [0x92,], [1, 8])
writeRegs(0x68, 0x71, [0x5,], [1, 8])
writeRegs(0x40, 0x0C, [0x40,], [1, 8])
writeRegs(0x40, 0x15, [0x80,], [1, 8])
writeRegs(0x40, 0x1F, [0x10,], [1, 8])
writeRegs(0xA0, 0x01, [0x6,0xF2,], [1, 8])
writeRegs(0xA0, 0xBF, [0x1,], [1, 8])
writeRegs(0x4C, 0xB5, [0x1,], [1, 8])
writeRegs(0x4C, 0xC0, [0xC0,0x98,0x80,0xB0,], [1, 8])
writeRegs(0x64, 0x40, [0x81,], [1, 8])
writeRegs(0x68, 0xCB, [0x1,], [1, 8])
writeRegs(0x68, 0x6C, [0xA2,], [1, 8])
writeRegs(0x68, 0x3D, [0x10,0x69,0x46,], [1, 8])
writeRegs(0x68, 0x4E, [0xFE,0x8,], [1, 8])
writeRegs(0x68, 0x02, [0xF,], [1, 8])
writeRegs(0x68, 0x57, [0xA3,0x4,], [1, 8])
writeRegs(0x68, 0x6F, [0x4,], [1, 8])
writeRegs(0x68, 0x75, [0x4,], [1, 8])
writeRegs(0x68, 0x83, [0xF0,], [1, 8])
writeRegs(0x68, 0x85, [0x10,], [1, 8])
writeRegs(0x64, 0x74, [0xF,], [1, 8])
writeRegs(0xB8, 0x41, [0x10,], [1, 8])
writeRegs(0xB8, 0x01, [0x0,0x18,0x0,], [1, 8])#  N[19:0])
writeRegs(0xB8, 0x13, [0xFF,], [1, 8])
writeRegs(0xB8, 0x15, [0x20,0x61,], [1, 8])
writeRegs(0xB8, 0x40, [0x80,], [1, 8])
writeRegs(0xB8, 0x4C, [0x4,], [1, 8])
writeRegs(0xB8, 0x55, [0x40,0x8,], [1, 8])
writeRegs(0xB8, 0x96, [0x20,], [1, 8])
writeRegs(0xB8, 0xAF, [0x96,], [1, 8])
writeRegs(0xB8, 0xBA, [0x70,], [1, 8])
writeRegs(0xB8, 0xD0, [0x44,0x3C,], [1, 8])
writeRegs(0xB8, 0xD3, [0x7,], [1, 8])
writeRegs(0xB8, 0xD6, [0x2,], [1, 8])
writeRegs(0xB8, 0xDB, [0xB,], [1, 8])
writeRegs(0xB8, 0xE0, [0x90,0xFC,], [1, 8])
writeRegs(0xB8, 0xE3, [0xD0,], [1, 8])
writeRegs(0xB8, 0xE8, [0xF0,0x1C,0x85,], [1, 8])
writeRegs(0xB8, 0xEC, [0x7C,0x40,0x40,0x41,], [1, 8])
writeRegs(0xB8, 0xF3, [0x1,], [1, 8])
writeRegs(0xB8, 0xF5, [0xCC,0x8,], [1, 8])
writeRegs(0x92, 0x24, [0x43,], [1, 8])
writeRegs(0x40, 0xE3, [0x0,], [1, 8])
#*****************************PVSP Script for Scaling 720P60 3D to 720P60 2D*****************
writeRegs(0x1A, 0xE828, [0x10,], [2, 8])
writeRegs(0x1A, 0x1A44, [0x4C,], [2, 8])
```

```
writeRegs(0x1A, 0xE884, [0x0,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE883, [0x80,], [2, 8])
writeRegs(0x1A, 0xE881, [0x4,0x4,], [2, 8])
writeRegs(0x1A, 0xE86C, [0x0,], [2, 8])
writeRegs(0x1A, 0xE935, [0xC6,], [2, 8])
writeRegs(0x1A, 0x1A4E, [0x88,], [2, 8])
writeRegs(0x1A, 0xE84E, [0x1,], [2, 8])
writeRegs(0x1A, 0xE890, [0x0,], [2, 8])
writeRegs(0x1A, 0xE828, [0x11,], [2, 8])
writeRegs(0x1A, 0xE828, [0x13,], [2, 8])
writeRegs(0x1A, 0xE828, [0x17,], [2, 8])
#**************************Tx1 Delay Settings for Less than 130 MHz**********************
writeRegs = topwin.devDriver.writeRegs
writeRegs(0x1A, 0xF324, [0x0,], [2, 8])
#************************************Tx1 Main Settings*************************
writeRegs(0x1A, 0xEC41, [0x10,], [2, 8])
writeRegs(0x1A, 0xEC01, [0x0,0x60,0x0,], [2, 8])#  Default 0x00, N[19:0])
writeRegs(0x1A, 0xEC13, [0xFF,], [2, 8])
writeRegs(0x1A, 0xEC15, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xEC40, [0x80,], [2, 8])
writeRegs(0x1A, 0xEC4C, [0x6,], [2, 8])
writeRegs(0x1A, 0xEC55, [0x40,0x8,], [2, 8])
writeRegs(0x1A, 0xEC96, [0x20,], [2, 8])
writeRegs(0x1A, 0xECAF, [0x96,], [2, 8])
writeRegs(0x1A, 0xECBA, [0x70,], [2, 8])
writeRegs(0x1A, 0xECD0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xECD3, [0x7,], [2, 8])
writeRegs(0x1A, 0xECD6, [0x2,], [2, 8])
writeRegs(0x1A, 0xECDB, [0xB,], [2, 8])
writeRegs(0x1A, 0xECE0, [0x90,0xFC,], [2, 8])
writeRegs(0x1A, 0xECE3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xECE8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xECEA, [0x85,], [2, 8])
writeRegs(0x1A, 0xECEC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xECF3, [0x1,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xECDA, [0x40,], [2, 8])
writeRegs(0x1A, 0xECF5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xECE9, [0x74,], [2, 8])
#************************************Tx1 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xEC4C, [0x4,], [2, 8])

#**************************Tx2 Delay Settings for Less than 130 MHz*********************
writeRegs = topwin.devDriver.writeRegs
writeRegs(0x1A, 0xFB24, [0x0,], [2, 8])
#************************************Tx2 Main Settings*************************
```

```
writeRegs(0x1A, 0xF441, [0x10,], [2, 8])
writeRegs(0x1A, 0xF401, [0x0,0x60,0x0,], [2, 8])#  Default 0x00, N[19:0])
writeRegs(0x1A, 0xF413, [0xFF,], [2, 8])
writeRegs(0x1A, 0xF415, [0xE0,0x61,], [2, 8])
writeRegs(0x1A, 0xF440, [0x80,], [2, 8])
writeRegs(0x1A, 0xF44C, [0x6,], [2, 8])
writeRegs(0x1A, 0xF455, [0x40,0x8,], [2, 8])
writeRegs(0x1A, 0xF496, [0x20,], [2, 8])
writeRegs(0x1A, 0xF4AF, [0x96,], [2, 8])
writeRegs(0x1A, 0xF4BA, [0x70,], [2, 8])
writeRegs(0x1A, 0xF4D0, [0x44,0x3C,], [2, 8])
writeRegs(0x1A, 0xF4D3, [0x7,], [2, 8])
writeRegs(0x1A, 0xF4D6, [0x2,], [2, 8])
writeRegs(0x1A, 0xF4DB, [0xB,], [2, 8])
writeRegs(0x1A, 0xF4E0, [0x90,0xFC,], [2, 8])
writeRegs(0x1A, 0xF4E3, [0xD0,], [2, 8])
writeRegs(0x1A, 0xF4E8, [0xF0,], [2, 8])
writeRegs(0x1A, 0xF4EA, [0x85,], [2, 8])
writeRegs(0x1A, 0xF4EC, [0x74,0x40,0x40,0x41,], [2, 8])
writeRegs(0x1A, 0xF4F3, [0x1,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xCC,0x8,0xF0,], [2, 8])
writeRegs(0x1A, 0xF4DA, [0x40,], [2, 8])
writeRegs(0x1A, 0xF4F5, [0xD4,], [2, 8])
writeRegs(0x1A, 0xF4E9, [0x74,], [2, 8])
#************************************Tx2 Eight Bits Per Pixel**********
writeRegs(0x1A, 0xF44C, [0x4,], [2, 8])
#************************************Setting Up VIM Cropping**********
writeRegs(0x1A, 0xE832, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE833, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE834, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE835, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE836, [0x05,], [2, 8]) #
writeRegs(0x1A, 0xE837, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xE838, [0x02,], [2, 8]) #
writeRegs(0x1A, 0xE839, [0xD0,], [2, 8]) #
writeRegs(0x1A, 0xE883, [0xC0,], [2, 8]) #
#************************************Setting Up the Vendor Specific InfoFrame**********
writeRegs(0x1A, 0xFAC0, [0x81,], [2, 8]) #Header
writeRegs(0x1A, 0xFAC1, [0x01,], [2, 8])
writeRegs(0x1A, 0xFAC2, [0x05,], [2, 8]) #
writeRegs(0x1A, 0xFAC3, [0x2A,], [2, 8]) #
writeRegs(0x1A, 0xFAC4, [0x03,], [2, 8]) #
writeRegs(0x1A, 0xFAC5, [0x0C,], [2, 8])
writeRegs(0x1A, 0xFAC6, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xFAC7, [0x40,], [2, 8]) #
writeRegs(0x1A, 0xFAC8, [0x00,], [2, 8]) #
writeRegs(0x1A, 0xF440, [0x81,], [2, 8]) #
```

```
#Muxing
writeRegs(0x1A, 0x1A05, [0x03,], [2, 8]) #PVSP from RX input
writeRegs(0x1A, 0x1A03, [0x16,], [2, 8]) #TX1 from PVSP, TX2 from HDMI RX
writeRegs(0x1A, 0xEC9F, [0x30,], [2, 8]) #
writeRegs(0x1A, 0xF49F, [0x30,], [2, 8]) #
```

注释

ESD警告

ESD(静电放电)敏感器件。带电器件和电路板可能会在没有察觉的情况下放电。尽管本产品具有专利或专有保护电路，但在遇到高能量ESD时，器件可能会损坏。因此，应当采取适当的ESD防范措施，以避免器件性能下降或功能丧失。

AN11540sc-0-6/13(0)

ANALOG DEVICES

www.analog.com