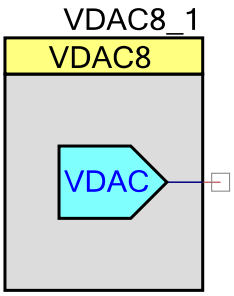
8 位电压数模转换器 (VDAC8)

1.60

特性



- 电压输出范围：1.020 V 和 4.080 V 全量程
- 软件或时钟驱动输出探针
- 数据源可以为 CPU、DMA 或 UDB

概述

VDAC8 组件是 8 位电压输出数模转换器 (DAC)。输出范围可以为从 0 到 1.020 V (4 mV/bit) 或从 0 到 4.08 V (16 mV/bit)。VDAC8 可以受硬件、软件或硬件和软件的组合限制。

输入/输出连接

本节介绍 VDAC8 的各种输入和输出连接。I/O 列表中的星号 (*) 表示该 I/O 可能在 I/O 说明中列出的情况下隐藏在符号中。

Vout – 模拟

Vout 终端是与 DAC 电压输出的连接。它可以路由到 PSoC 上的任何模拟兼容引脚。

data[7:0] – 输入 *

此 8 位宽数据信号将 VDAC8 直接连接到 DAC 总线。DAC 总线可以由基于 UDB 的组件或控制寄存器驱动，或者它也可以从 GPIO 引脚直接路由。可以通过将 **Data_Source** 参数设置为“DAC 总线”来使能此输入。如果选择了“CPU 或 DMA”选项，则总线连接将从组件符号中消失。

当硬件能够在没有 CPU 干预的情况下设置正确的值时，使用 data[7:0] 输入。当使用此选项时，探针选项还应当设置为“外部”。

对于许多应用场合，不需要此输入，但是 CPU 或 DMA 会将值直接写入数据寄存器。在固件中，使用 VDAC8_SetRange() 函数或者直接将值写入 VDAC8 数据寄存器。

探针 – 输入 *

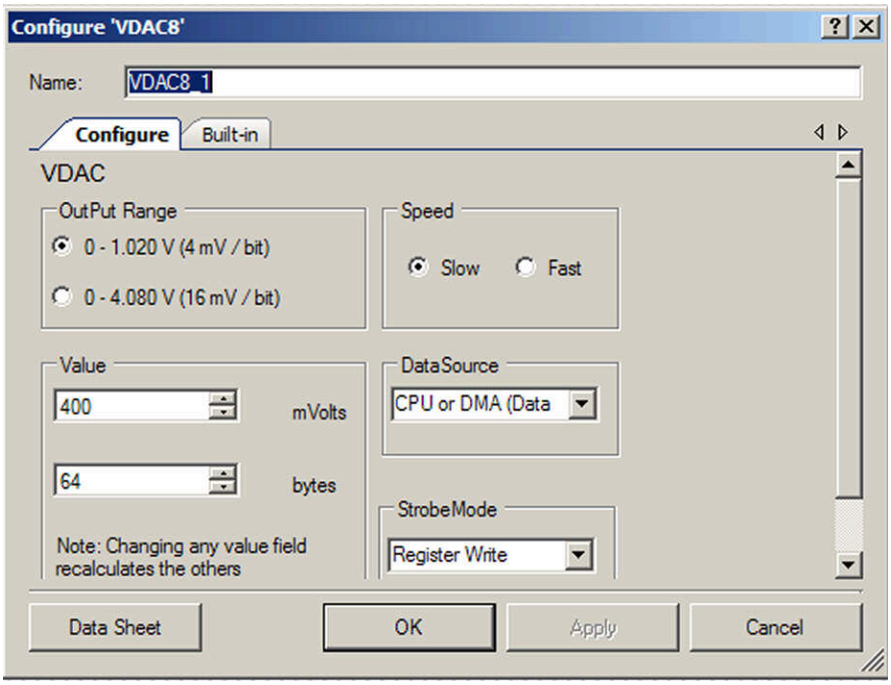
探针输入是可选信号输入，是通过 **Strobe_Mode** 参数选择的。

- 如果 **Strobe_Mode** 设置为“外部”，则探针引脚将是可见的，且必须连接到有效数据源。在此模式中，数据从 VDAC8 寄存器传输到探针信号下一个正边沿上的 DAC。
- 如果 **Strobe_Mode** 设置为“寄存器写入”，则引脚将从符号中消失，对数据寄存器的任何写入将立即传输到 DAC。

对于音频或定期采样应用场合，还可以使用对传入 DAC 的数据进行测速的同一时钟来生成中断。时钟的每个上升沿都将数据传输到 DAC，导致中断获取加载到 DAC 寄存器的下一个值。

参数和设置

将 VDAC8 组件拖动到您的设计中，双击它可打开“配置”对话框。



VDAC8 组件提供下列参数。

Data_Source

此参数选择要写入 DAC 寄存器的数据源。如果 CPU（固件）或 DMA 将数据写入 VDAC8，则选择“CPU 或 DMA”。如果数据直接从 UDB 或基于 UDB 的组件写入，则选择“DAC 总线”。

当选择 DAC 总线时，VDAC 符号上指示了输入。只有一个 DAC 总线，因此多个 VDAC 不能具有独立的硬件 (UDB) 数据源。

Initial_Value

这是执行 VDAC8_Start() 命令后 VDAC8 将提供的初始值。VDAC8_SetValue() 函数或对 DAC 寄存器的直接写入将随时覆盖默认值。合法值介于 0 和 255 之间（含）。

Strobe_Mode

此参数选择当数据写入 VDAC8 数据寄存器时数据是否立即写入 DAC。当选择“寄存器写入”选项时，选择此模式。当选择“外部”选项时，来自 UDB 的时钟或信号控制数据何时从 DAC 寄存器写入实际 DAC。

VDAC_Range

此参数允许您将两个电压范围之一设置为默认值。在运行时可以随时使用 VDAC8_SetRange() 函数更改范围。

范围	最低值	最高值	步进大小
Range_1_Volt	0.0 mV	1.020 V	4 mV
Range_4_Volt	0.0 mV	4.080 V	16 mV

输出等式：

- 1 V 范围 – Vout = (值/256) * 1.024 V
- 4 V 范围 – Vout = (值/256) * 4.096 V

注意：术语“值”是介于 0 和 255 之间的数。

VDAC_Speed

此参数提供两种设置：“低速”和“高速”。在“低速”模式下，设置时间较慢，但是使用的工作电流较小。在“高速”模式下，电压处理速度快得多，但是代价是工作电流较大。

资源

VDAC8 组件使用一个 viDAC8 模拟模块。

模拟模块	数字模块					API 存储器（字节）		引脚（每个外部 I/O）
	数据路径	宏单元	状态寄存器	控制寄存器	计数器 7	闪存	RAM	
1 viDAC8 HW	不可用	不可用	不可用	不可用	不可用	354	3	1

应用程序编程接口

应用程序编程接口 (API) 子程序允许您使用软件配置组件。下表列出了每个函数的接口，并进行了说明。以下各节将更详细地介绍每个函数。

默认情况下，PSoC Creator 将实例名称“VDAC8_1”分配给提供的设计中的第一个组件实例。您可以将该实例重命名为符合标识符语法规则的任意唯一值。实例名称会成为每个全局函数名称、变量和常量符号的前缀。为增加可读性，下表中使用了实例名称“VDAC8”。

函数	说明
void VDAC8_Start(void)	使用默认自定义程序值初始化 VDAC8。
void VDAC8_Stop(void)	禁用 VDAC8 并将其设置为最低功耗状态。
void VDAC8_SetSpeed(uint8 speed)	设置 DAC 速度。
void VDAC8_SetValue(uint8 value)	使用给定范围设置 0 到 255 之间的值。
void VDAC8_SetRange(uint8 value)	将范围设置为 1 或 4 V。
void VDAC8_Sleep(void)	停止并保存用户配置。
void VDAC8_WakeUp(void)	还原并使能用户配置。
void VDAC8_Init(void)	初始化或恢复默认 VDAC8 配置
void VDAC8_Enable(void)	使能 VDAC8。
void VDAC8_SaveConfig(void)	清空函数。提供以便将来使用。
void VDAC8_RestoreConfig(void)	清空函数。提供以便将来使用。

全局变量

变量	说明
VDAC8_initVar	指示是否已初始化 VDAC8。该变量初始化为 0 并在第一次调用 VDAC8_Start() 时设置为 1。这允许第一次调用 VDAC8_Start() 子程序后组件无需重新初始化便可重新启动。 如果需要重新初始化组件，则可以在 VDAC8_Start() 或 VDAC8_Enable() 函数之前调用 VDAC8_Init() 函数。

void VDAC8_Start(void)

说明： 这是开始执行组件操作的首选方法。VDAC9_Start() 设置 initVar 变量，调用 VDAC8_Init() 函数，调用 VDAC8_Enable() 函数，然后将 VDAC8 加电到给定的功耗水平。功耗水平 0 相当于执行停止函数。

参数： 无

返回值： 无

副作用： 如果已设置了 initVar 变量，则此函数仅调用 VDAC8_Enable() 函数。

void VDAC8_Stop(void)

说明： 将 VDAC8 断电至最低功耗状态，并禁用输出。

注意：不建议将此 API 用于 PSoC 3 ES2 和 PSoC 5 ES1 芯片。这些器件有一个缺陷，导致与某些模拟资源的连接在未断电时不可靠。当停止使用该资源的组件时，该不可靠性会在静默失败中表现出来（例如模拟组件中出现不可预见的失败结果）。建议设计中的所有模拟组件应当永远保持加电（通过调用 VDAC8_Start() API）。不要调用 VDAC8_Stop() API。

参数： 无

返回值： 无

副作用： 无

void VDAC8_SetSpeed(uint8 speed)

说明： 设置 DAC 速度。

参数： (uint8) speed：设置 DAC 速度，参见下表可获取有效参数。

选项	说明
VDAC8_LOWSPEED	低速（低功耗）
VDAC8_HIGHSPEED	高速（高功耗）

返回值： 无

副作用： 无

void VDAC8_SetRange(uint8 range)

说明： 将范围设置为 1 或 4 V。

参数： (uint8) range：设置 VDAC8 的全量程范围。有关范围，请参见下表。

选项	说明
VDAC8_RANGE_1V	设置 1.020 V 的全量程范围
VDAC8_RANGE_4V	设置 4.080 V 的全量程范围

返回值： 无

副作用： 无

void VDAC8_SetValue(uint8 value)

说明： 将值设置为 VDAC8 上的输出。有效值介于 0 和 255 之间。

参数： (uint8) value：介于 0 和 255 之间的值。值 0 是最低值（零），值 255 是全量程值。全量程值取决于可使用 SetRange API 选择的范围。

返回值： 无

副作用： 在 PSoC 3 ES2 和 PSoC 5 ES1 芯片上，应当在使能 VDAC 电源后调用 VDAC8_SetValue() 函数。

void VDAC8_Sleep(void)

说明： 这是准备组件进入睡眠的首选 API。VDAC8_Sleep() API 保存当前组件状态。然后它调用 VDAC8_Stop() 函数并调用 VDAC8_SaveConfig() 以保存硬件配置。
在调用 CyPmSleep() 或 CyPmHibernate() 函数之前调用 VDAC8_Sleep() 函数。有关电源管理函数的更多信息，请参考 PSoC Creator *System Reference Guide*（《系统参考指南》）。

参数： 无

返回值： 无

副作用： 无

void VDAC8_Wakeup(void)

说明： 这是将组件恢复为调用 VDAC8_Sleep() 时的状态的首选 API。VDAC8_Wakeup() 函数调用 VDAC8_RestoreConfig() 函数以恢复配置。如果在调用 VDAC8_Sleep() 函数之前组件处于使能状态，则 VDAC8_Wakeup() 函数还将重新使能组件。

参数： 无

返回值： 无

副作用： 在不首先调用 VDAC8_Sleep() 或 VDAC8_SaveConfig() 函数的情况下调用 VDAC8_Wakeup() 函数会产生意外行为。

void VDAC8_Init(void)

说明： 根据定制器“配置”对话框设置来初始化或恢复组件。不需要调用 VDAC8_Init()，因为 VDAC8_Start() API 会调用此函数，这是开始组件操作的首选方法。

参数： 无

返回值： 无

副作用： 所有寄存器将设置为其初始值。这将重新初始化 组件。如果您要设置除寄存器中当前值之外的新值，则调用 VDAC8_Init() 函数需要调用 VDAC8_SetValue()。

void VDAC8_Enable(void)

说明： 激活硬件并开始执行组件操作。不需要调用 VDAC8_Enable()，因为 VDAC8_Start() API 会调用此函数，这是开始组件操作的首选方法。

参数： 无

返回值： 无

副作用： 无

void VDAC8_SaveConfig(void)

说明： 此函数保存组件配置。它将保存非保留寄存器。此函数还将保存当前“配置”对话框中定义的或通过相应 API 修改的组件参数值。此函数由 VDAC8_Sleep() 函数调用。

参数： 无

返回值： 无

副作用： 清空函数。实现以供将来使用。调用此函数对组件没有作用。

void VDAC8_RestoreConfig(void)

说明： 此函数恢复组件配置。此将恢复非保留寄存器。此函数还将组件参数值恢复为调用 VDAC8_Sleep() 函数之前的值。

参数： 无

返回值： 无

副作用： 清空函数。实现以供将来使用。调用此函数对组件没有作用。

DMA 向导

VDAC8 组件不需要实现 DMA 请求信号。典型的用法是信号生成，应当在外部控制传输到 VDAC8 组件的数据速率。可以使用 DMA 向导按如下所示配置 DMA 操作：

DMA 向导中 DMA 源/目标的名称	方向	DMA 请求信号	DMA 请求类型	说明
VDAC8_Data_PTR	目标	不可用	不可用	存储介于 0 到 255 的 DAC 值

固件源代码示例

PSoC Creator 在“查找示例项目”对话框中提供了大量包括原理图和代码示例的示例项目。要获取组件特定的示例，请打开组件目录中的对话框或原理图中的组件实例。要获取通用的示例，请打开开始页或**文件**菜单中的对话框。根据需要，使用对话框中的**滤波器选项**可缩小可选项目的列表。

有关更多信息，请参考 PSoC Creator 帮助中的“查找示例项目”主题。

功能说明

当用作 VDAC8 时，viDAC8 模拟模块配置为电压 DAC 且可用作电压源。

当用作 VDAC 时，输出是 8 位数模转换电压，以支持需要参考电压的场合。在这里，参考源是从取自称为 VREF(DAC) 的模拟参考模块的电压参考。通过设置 DACx_CR0 [4] 寄存器，DAC 可以配置为在电压模式下工作。在此模式下，有两个通过 DACx_CR0[3:2] 寄存器选择的输出范围：

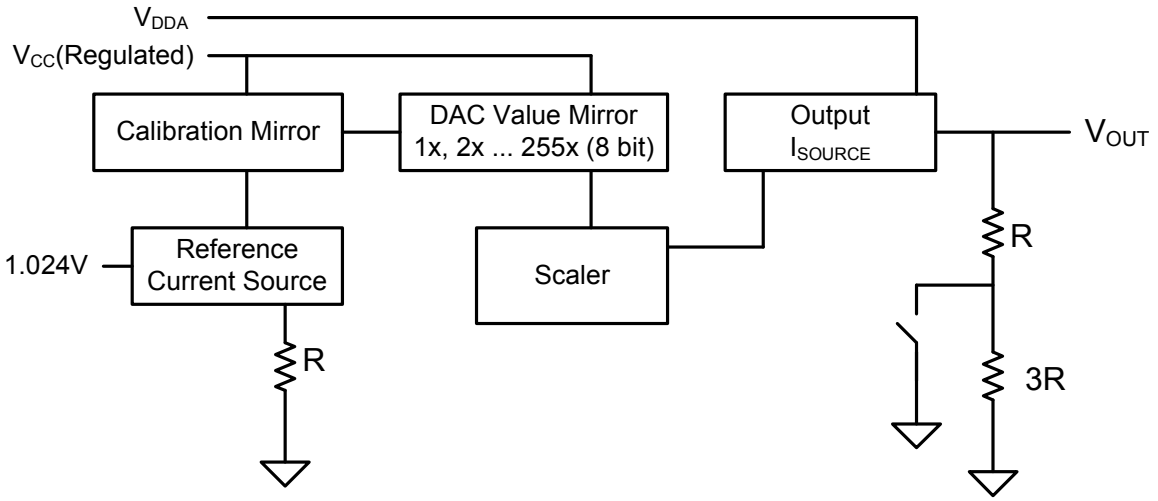
- 0 V 到 1.024 V
- 0 V 到 4.096 V

两个输出范围都具有 255 个相等步进。通过电阻驱动当前 DAC 的输出，并获取电压输出，可以实现 VDAC。由于不使用缓冲区，任何取自 DAC 的直流电流都影响输出电平。因此，在此模式下，任何连接到输出的负载都应当是电容性负载。

VDAC 最高能够将转换为 1 Msps。此外，DAC 在 4 V 模式下比 1 V 模式下慢，这是因为 Vssa 的电阻式阻抗比它大 4 倍。在 4 V 模式下，VDAC 最高能够将转换为 250 ksps。

框图和配置

下面显示了 VDAC8 组件的框图。



寄存器

提供的函数支持大多数应用场合下所需的大多数通用运行时函数。以下寄存器参考信息为高级用户提供了简要的说明。无需使用 API，可以使用 VDAC8_Data 寄存器将数据直接写入 DAC。这对于 CPU 或 DMA 非常有用。

表 1 VDAC8_CR0

位	7	6	5	4	3	2	1	0
值	RSVD			模式	Range[1:0]		hs	RSVD

- 模式：将 DAC 设置为电压或电流模式。
- range[1:0]：DAC 范围设置。
- hs：用于设置数据速度。

表 2 VDAC8_CR1

位	7	6	5	4	3	2	1	0
值	RSVD		mx_data	reset_udb_en	mx_idir	idirbit	Mx_ioff	ioffbit

- mx_data：选择数据源。
- reset_udb_en：DAC 复位使能。
- mx_idir：DAC 电流方向控制的复用器选择。
- idirbit：DAC 电流方向的寄存器源。
- mx_off：DAC 电流关断控制的复用器选择。
- ioffbit：DAC 电流关断的寄存器源

表 3 VDAC8_DATA

位	7	6	5	4	3	2	1	0
值	Data[7:0]							

- Data[7:0]：DAC 数据寄存器。

直流和交流电气特性

下列值基于特性数据。除非另外说明，否则规格仅对适用于 -40°C ≤ T_A ≤ 85°C 和 T_J ≤ 100°C。除非下表中另外指定，否则所有典型值都针对 T_A = 25°C、Vdda = 5.0V、参考模拟地 (Vssa) 的输出、快速模式。

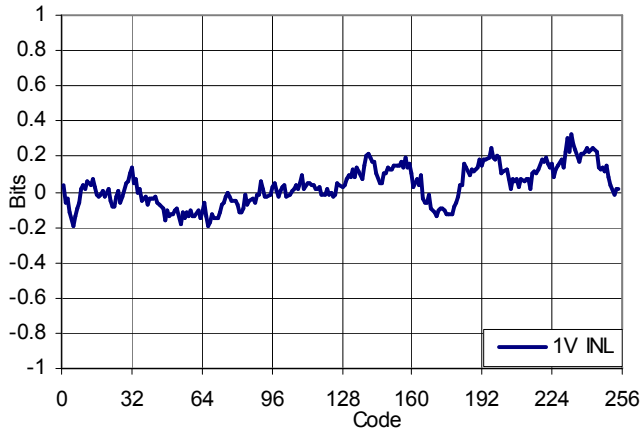
VDAC8 直流电气特性

参数	说明	条件	最小值	典型值	最大值	单位
	分辨率		–	8	–	位
INL1	积分非线性	1 V 量程	–	±2.1	±2.5	LSB
DNL1	差分非线性	1 V 量程	–	±0.3	±1	LSB
Rout	输出阻抗	1 V 量程	–	4	–	k
		4 V 量程	–	16	–	k
V_{OUT}	输出电压范围，代码 = 255	1 V 量程	–	1	–	V
		4 V 量程，Vdda = 5 V	–	4	–	V
	单调性		–	–	是	–
V_{OS}	零量程错误		–	0	±0.9	LSB

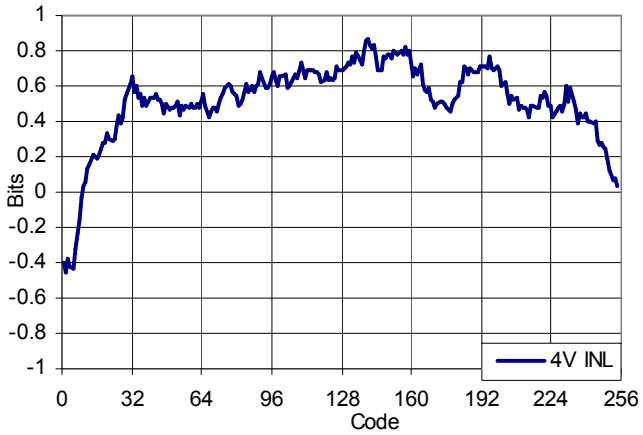
参数	说明	条件	最小值	典型值	最大值	单位
FSGainErr	全量程增益误差	1 V 量程	–	±1.6	±2.5	%
		4 V 量程	–	±1.5	±2.5	%
TCGainErr	温度系数，增益误差	1 V 量程	–	–	0.02	%FSR / °C
		4 V 量程	–	–	0.02	%FSR / °C
I_{DD}	工作电流	慢速模式	–	–	100	µA
		快速模式	–	–	500	µA

图形

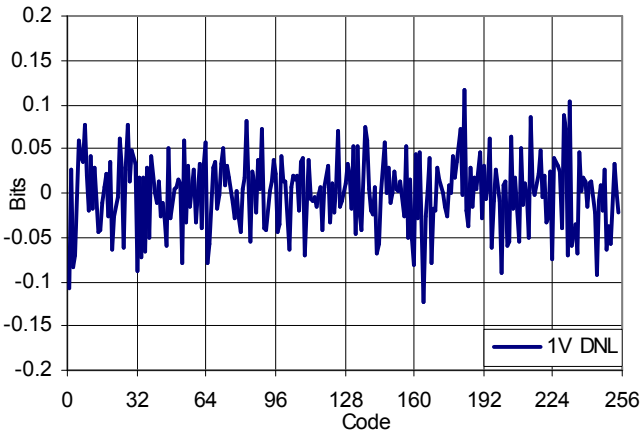
INL 与 DAC 代码，1.0V 范围



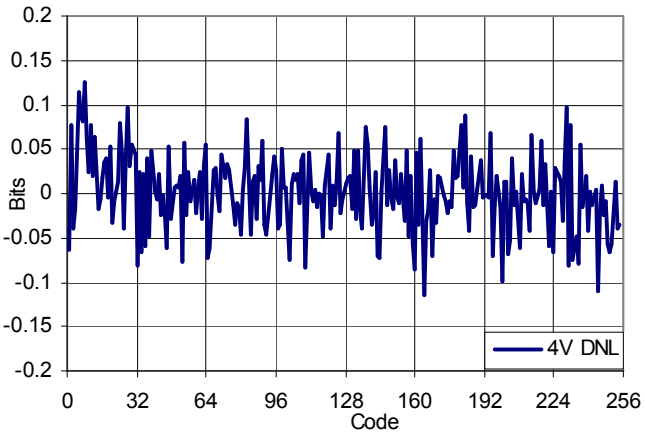
INL 与 DAC 代码，4.0V 范围



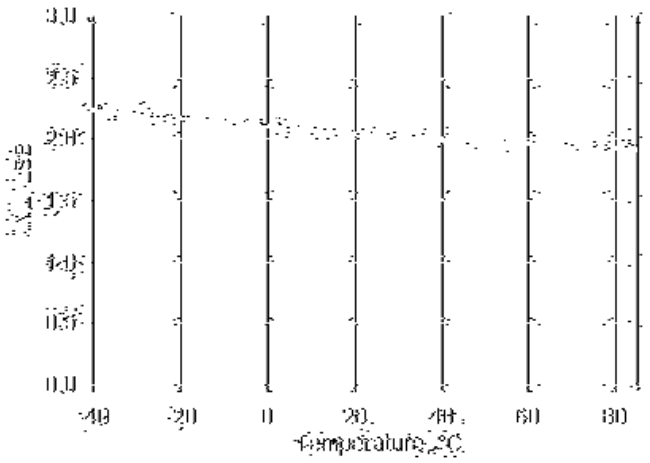
典型 DNL 与 DAC 代码，1.0V 范围



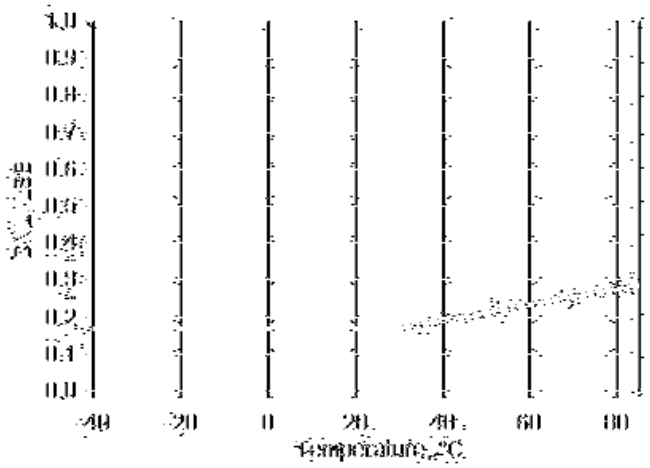
DNL 与 DAC 代码，4.0V 范围



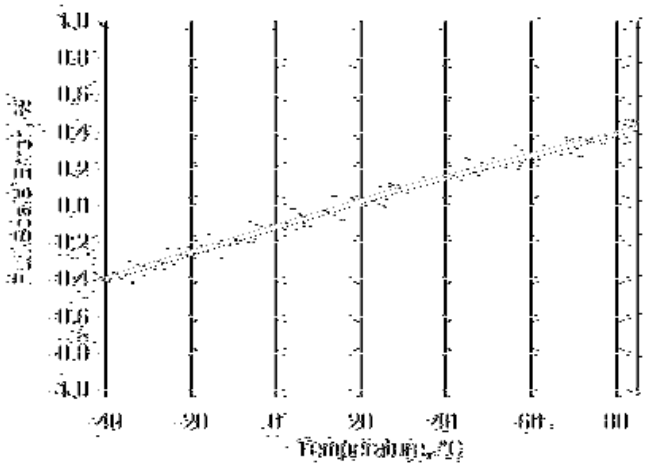
VDAC INL 与温度，1V 模式



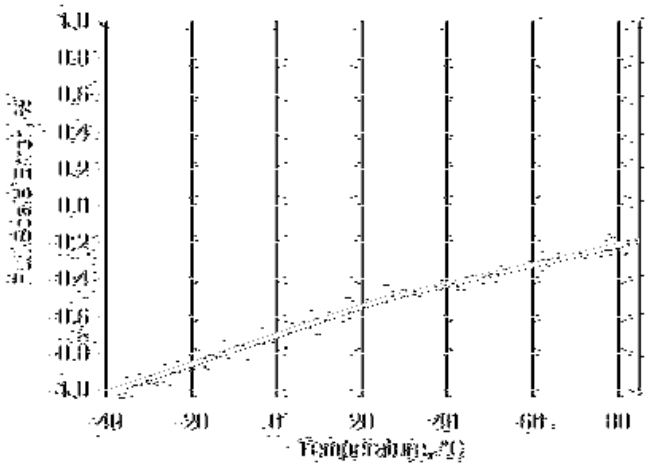
VDAC DNL 与温度，1V 模式



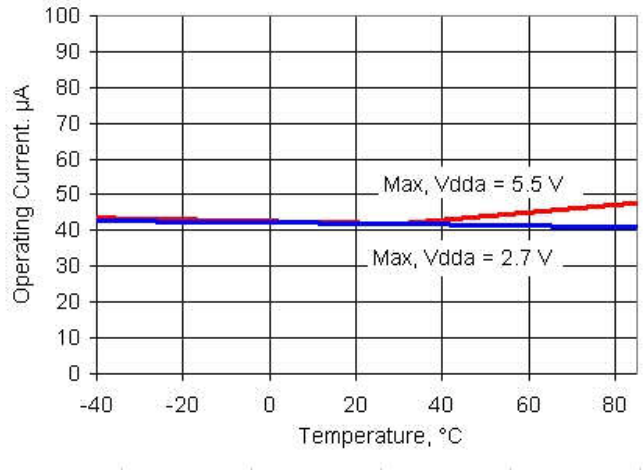
VDAC 全量程错误与温度，1V 模式



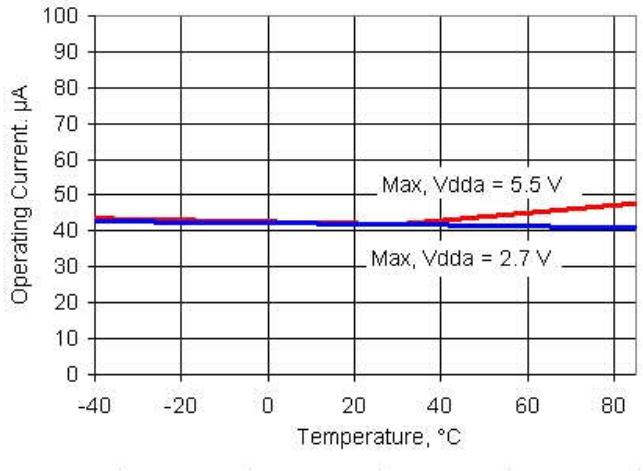
VDAC 全量程误差与温度，4 V 模式



VDAC 工作电流与温度，1V 模式，慢速模式



VDAC 工作电流与温度，1V 模式，快速模式



VDAC8 交流电气特性

参数	说明	条件	最小值	典型值	最大值	单位
F_{DAC}	更新速率	1 V 量程	–	–	1000	ksps
		4 V 量程	–	–	250	ksps
TsettleP	设置时间为 0.1%，步进 25% 到 75%	1 V 量程，Cload = 15 pF	–	0.45	1	µs
		4 V 量程，Cload = 15 pF	–	0.8	3.2	µs
TsettleN	设置时间为 0.1%，步进 75% 到 25%	1 V 量程，Cload = 15 pF	–	0.45	1	µs
		4 V 量程，Cload = 15 pF	–	0.7	3	µs
SRP	斜率，步进 10% 到 90%	1 V 量程，Cload = 15 pF	–	0.3	0.5	µs
		4 V 量程，Cload = 15 pF	–	0.5	1.3	µs
SRN	斜率，步进 90% 到 10%	1 V 量程，Cload = 15 pF	–	0.3	0.5	µs
		4 V 量程，Cload = 15 pF	–	0.3	1.3	µs

术语

积分非线性误差 (INL)

INL（积分非线性误差）是对 LSB 中与超出 DAC 工作范围的最佳拟合直线的最大偏差的测量。

微分非线性误差 (DNL)

DNL（微分非线性误差）是任何两个相邻代码之间的测量出的变化与理想 1 LSB 变化之间的差异。设计上保证此 VDAC 呈单调变化。输出为“温度计编码”，通过打开用以前使能的输出源汇总的单独输出源，进行每个后续步骤。

单调性

对于每个增大的数字代码输入值，如果输出增大或保持不变，则 DAC 定义为单调变化。VDAC8 组件在电压和温度的整个工作范围之上单调变化。

零量程误差

零量程误差是代码 0x00 处测量值与代码 0x00 处最佳拟合直线的值之间的差。

全量程增益误差

全量程增益误差是最大值代码处测量值与额定值之间的差。在代码 = 255 (0x00) 处，最大值为 1.020 V 或 4.080 V。

全量程增益温度系数 (TC)

全量程增益温度系数是全量程值（最大值代码 0xFF）的变化与温度变化。较低值处的增益更改与代码值成正比。

电源抑制比 (PSRR)

电源抑制比测量 VDAC 输出与电源的隔离度。

建立时间

建立时间是输出建立到特定数字输入更改的特定级别而所需的时间量。

斜率

斜率是 VDAC 输出的最大变化速率。斜率测量值为整个量程值的 10% 到 90%

短时脉冲幅度

短时脉冲幅度是当输入代码在中间量程更改单一计数时（0x7f 到 0x80）注入到输出的脉冲峰值幅度。该脉冲超出了数据更改前后的静态值之差。

电压噪声

电压噪声是 VDAC 输出阻抗的噪声和电流输出噪声与 VDAC 输出阻抗的乘积之和。此噪声作为代码值的函数变化。

组件更改

本节介绍组件与以前版本相比的主要更改。

版本	更改说明	更改/影响原因
1.60	添加了 GUI 配置编辑器	以前的配置窗口没有提供足够的易于使用的信息。
	向数据手册中添加了特性数据	
	对数据表进行了少量编辑和更新	
1.50	添加了睡眠/唤醒和初始化/使能 API。	为了支持低功耗模式，以及提供通用接口以分别控制大多数组件的初始化和使能。
	向组件中添加了 DMA 功能文件。	此文件允许 PSoC Creator 中的 DMA 向导工具支持 VDAC8。

© 赛普拉斯半导体公司，2011。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品的内嵌电路之外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任，也不根据专利或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯产品不保证能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

PSoC® 是赛普拉斯半导体公司的注册商标，PSoC Creator™ 和 Programmable System-on-Chip™ 是赛普拉斯半导体公司的商标。此处引用的所有其他商标或注册商标归其各自所有者所有。

所有源代码（软件和/或固件）均归赛普拉斯半导体公司（赛普拉斯）所有，并受全球专利法规（美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和/或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定的用途之外，未经赛普拉斯的明确书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对此材料提供任何类型的明示或暗示保证，包括（但不仅限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不做出通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于可能发生运转异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统中，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用的赛普拉斯软件许可协议限制。