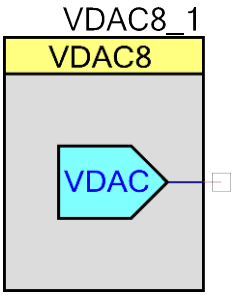
8 ビット電圧デジタル・アナログ変換器 (VDAC8)

1.70

特徴



- 電圧出力範囲: 1.020-V および 4.080-V フルスケール
- ソフトウェアまたはクロック駆動出力ストローブ
- データ ソースには、CPU、DMA、UDB があります

概要

VDAC8 コンポーネントは、8 ビット出力デジタル・アナログ変換器 (DAC) です。出力範囲は、0 から 1.020 V (4 mV/bit)、または 0 から 4.08 V (16 mV/bit) です。VDAC8 は、ハードウェア、ソフトウェア、またはハードウェアとソフトウェアの組合せによって制御されます。

入出力接続

ここでは、VDAC8 のさまざまな入出力接続について説明します。I/O リストのアスタリスク (*) は、その I/O の設定条件によって、シンボルが非表示となる場合があるいことを示します。

Vout – アナログ

Vout 端子は DAC の電圧出力への接続子です。PSoC のアナログ対応ピンにルーティングすることができます。

注 ピンにドライブされる場合は、VDAC はそのピンの VDDIO を超える値をドライブできません。希望する結果を得るには、正しい VDDIO 電源を設定してください。

data[7:0] – 入力 *

この 8 ビット幅データ信号は、VDAC8 を DAC バスに直接接続します。DAC バスは、UDB ベース コンポーネントやコントロール レジスタで駆動するか、GPIO ピンから直接ルーティングすることができます。この入力は、**Data_Source** パラメータを **DAC Bus** に設定してイネーブルにします。**CPU または DMA** オプションを選択する場合は、バス接続のシンボルが非表示になります。

ハードウェアが CPU の介入なしに適切な値を設定できる場合は、data[7:0] 入力を使用します。このオプションを使用する場合は、ストローブ オプションを **External** に設定します。

多くの用途では、この入力は必要ありませんが、CPU または DMA がデータ レジスタに直接値を書き込みます。ファームウェアでは、VDAC8_SetValue() 関数を使用するか、そうでない場合は値を直接 VDAC8 データ レジスタに書き込みます。

strobe – 入力 *

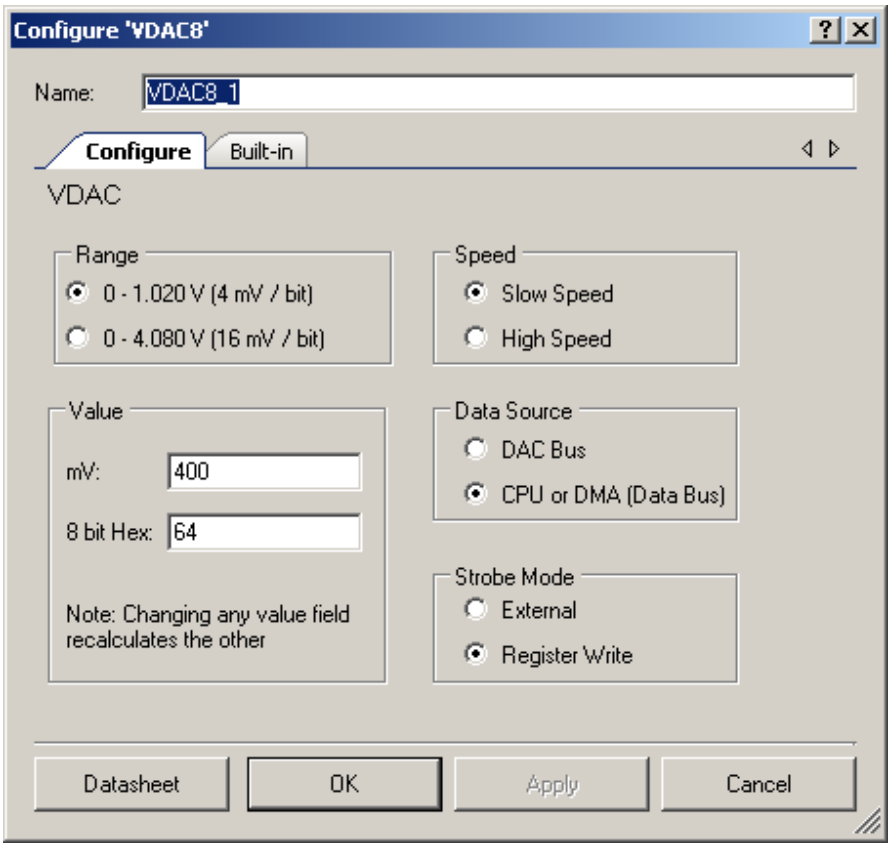
ストローブ入力はオプションの信号入力で、**StrobeMode** パラメータで選択されます。

- **StrobeMode** が **External** に設定されると、ストローブ ピンが表示されます。有効なデジタル ソースに接続する必要があります。このモードでは、VDAC8 レジスタから DAC へのデータ転送は、ストローブ信号のポジティブエッジに同期します。
- **StrobeMode** が **Register Write** に設定されると、ピンシンボルが非表示になり、、データ レジスタへ書き込まれたデータは即座に DAC に転送されます。

オーディオや定期的なサンプリング用途では、データを DAC へ転送するクロック信号で、割り込みを発生させることもできます。この場合、クロックの各立ち上がりエッジはデータを DAC へ転送し、DAC レジスタにロードされた次の値を取得するための割り込みを発生します。

コンポーネント パラメータ

VDAC8 コンポーネントをデザイン上にドラッグし、ダブルクリックして **Configure**(設定) ダイアログを開きます。



VDAC8 コンポーネントには次のパラメータがあります。

Range(範囲)

デフォルト値として 2 つの電圧範囲のいずれかを設定します。範囲は、VDAC8_SetRange() 関数を実行することで、プログラム実行中に変更することができます。

Range(範囲)	Lowest Value(最低値)	Highest Value(最高値)	ステップサイズ
Range_1_Volt	0.0 mV	1.020 V	4 mV
Range_4_Volt	0.0 mV	4.080 V	16 mV

出力式:

- 1-V 範囲: V_{OUT} = (value/256) × 1.024 V
- 4-V 範囲: V_{OUT} = (value/256) × 4.096 V

注 ここでの「value」は 0 から 255 の数字です。

値

これは、VDAC8_Start() コマンドが実行された後 VDAC8 が出力する初期値です。VDAC8_SetValue() 関数、または DAC レジスタへの直接書き込みは、いつでもデフォルト値を上書きします。有効な値は 0 から FF です。**mV** フィールドには VDAC 出力電圧 (ミリボルト) が表示され、**8 bit Hex**(8 ビット 16 進) には VDAC 入力 データ値(16 進数)が表示されます。

Speed

このパラメータには次の 2 つの設定があります。**Slow**(遅い) と **Fast**(速い)。**Slow** モードでは、電圧セトリング時間は遅いですが、動作電流消費が少なくなります。**Fast** モードでは、電圧セトリング時間が速くなりますが動作電流消費が高くなります。

Data Source(データ ソース)

DAC レジスタに書き込むデータ ソースを選択します。CPU (ファームウェア) または DMA に VDAC8 へデータを書き込ませる場合は、**CPU または DMA (データ バス)**を選択します。UDB または UDB ベース コンポーネントから直接データを書き込ませるには **DAC Bus** を選択します。

DAC Bus が選択されると、入力シンボルが VDAC に表示されます。1つしか DAC バスがないので、複数の VDAC は独立したハードウェア (UDB) データ ソースを持つことができません。

Data Source が **DAC Bus** として設定されると、カスタマイザは **Strobe Mode**(ストローブ モード) を **External** に自動的に設定し、変更できないようにオプションを無効にします。

注 PSoC 5 シリコンの場合、DAC へ新しい値を書き込むと、DAC 出力が不定値になります。所望の値を出力するには、同じ値で DAC を2回書き込みまたはストローブします。最初の書き込みが不定出力になる場合があるので、2 回の書き込みの間隔は最小にしてください。これは、CPU、DMA、ストローブによる書き込みの場合に適用されます。API IDAC8_SetValue() は提供された値を 2 回書き込んで、CPU 書き込みでのこの問題を軽減します。

Strobe Mode(ストローブ モード)

データが VDAC8 データ レジスタに書き込む場合、データがすぐに DAC に書き込まれるかを選択します。**Register Write**(レジスタ書き込み) オプションが選択されると有効になります。**External** オプションが選択した場合、UDB からのクロックまたは信号により、DAC レジスタから DAC へのデータ転送が制御されます。

リソース

VDAC8 コンポーネントは 1 つの viDAC8 アナログ ブロックを使用します。

アナログブロック	デジタルブロック					API メモリ（バイト）		ピン（外部入出力当たり）
	データパス	マクロセル	ステータス レジスタ	コントロール レジスタ	Counter7	フラッシュ	RAM	
1 viDAC8 HW	該当せず	該当せず	該当せず	該当せず	該当せず	354	3	1

アプリケーションプログラミングインタフェース

アプリケーションプログラミングインターフェース (API) ルーチンにより、ソフトウェアを使用してコンポーネントを設定できます。次の表は、各関数へのインターフェースとその説明を示しています。続くセクションでは、各関数について詳しく説明します。

デフォルトでは、PSoC Creator は、特定の設計のコンポーネントの最初のインスタンスにインスタンス名「VDAC8_1」を割り当てます。インスタンス名は、識別子の構文ルールに従った固有の値に変更できます。インスタンス名は、すべてのグローバル関数名、変数名、定数名のプリフィックスになります。便宜上、次の表ではインスタンス名として「DMA」を使っています。

関数	説明
VDAC8_Start()	VDAC8 をデフォルトのカスタマイザ値に初期化します。
VDAC8_Stop()	VDAC8 を無効にして、最低電源消費量に設定します。
VDAC8_SetSpeed()	DACスピードを設定します。
VDAC8_SetValue()	0 から 255 の範囲の値を特定の値に設定します。
VDAC8_SetRange()	範囲を 1 または 4 ボルトに設定します。
VDAC8_Sleep()	動作を停止し、ユーザ構成を保存します。
VDAC8_WakeUp()	ユーザ構成を復元し、イネーブルにします。
VDAC8_Init()	初期化するか、デフォルトの VDAC8 設定を復元します。
VDAC8_Enable()	VDAC8 有効にします。
VDAC8_SaveConfig()	保持されない DAC データ レジスタ値を保存します。
VDAC8_RestoreConfig()	保持されない DAC データ レジスタ値を復元します。

グローバル変数

変数	説明
VDAC8_initVar	VDAC8 が初期化されているかを示します。変数は、0 に初期化され、最初に VDAC8_Start() が呼び出されると 1 に設定されます。これにより、コンポーネントは VDAC8_Start() ルーチンへの最初の呼び出し後、再初期化せずに再起動できます。 コンポーネントの初期化が必要な場合は、VDAC8_Start() または VDAC8_Enable() 関数の前に VDAC8_Init() 関数を呼び出します。

void VDAC8_Start(void)

説明: これは、コンポーネントの動作を開始する際に推奨される方法です。VDAC8_Start() は initVar 変数を設定し、VDAC8_Init() 関数、VDAC8_Enable() 関数の順に呼び出して、VDAC8 を特定の消費電力レベルに上げます。消費電力レベル 0 は、VDAC_Stop() 関数実行と同じです。

パラメータ: なし

返り値: なし

副作用: initVar 変数がすでに設定されている場合は、この関数は VDAC8_Enable() 関数を呼び出すだけです。

void VDAC8_Stop(void)

説明: VDAC8 を最低消費電力に下げて、出力を無効にします。

注 PSoC 3 ES2 および PSoC 5 シリコンでこの API を使用することは推奨されません。これらのデバイスは、電源が切れている場合、アナログリソースへの接続が不安定になる欠陥があります。不安定な状態とは、そのリソースを使用しているコンポーネントが停止した際に、サイレントな欠陥 (例:アナログ コンポーネントの予期しない不良な結果) という形で現れます。VDAC8 コンポーネントは必ず消費電力を上げます (VDAC8_Start() API 呼び出し)。VDAC8_Stop() API を呼び出さないでください。

パラメータ: なし

返り値: なし

副作用: なし

void VDAC8_SetSpeed(uint8 speed)

説明: DAC シードを設定します。

パラメータ: uint8 speed: DACスピードを設定します。有効なパラメータについては、以下の表を参照してください。

オプション	説明
VDAC8_LOWSPEED	低速度 (低消費電力)
VDAC8_HIGHSPEED	高速 (高消費電力)

返り値: なし

副作用: なし

void VDAC8_SetRange(uint8 range)

説明: 範囲を 1 または 4 ボルトに設定します。

パラメータ: uint8 range: VDAC8 にフル スケール範囲を設定します。範囲については以下の表を参照してください。

オプション	説明
VDAC8_RANGE_1V	1.020 V のフル スケール範囲を設定
VDAC8_RANGE_4V	4.080 V のフル スケール範囲を設定

返り値: なし

副作用: なし

void VDAC8_SetValue(uint8 value)

説明: VDAC8 の出力に値を設定します。有効な値は 0 ～ 255 までです。

パラメータ: uint8 value: 0 ～ 255 の値。0 値は最低 (ゼロ) で、255 値はフルスケール値です。フルスケール値は、VDAC8_SetRange() API で選択される範囲によって異なります。

返り値: なし

副作用: PSoC 3 ES2、PSoC 3 Production、PSoC 5 シリコンでは、VDAC への電力を有効にした後で VDAC8_SetValue() 関数が呼び出されます。

void VDAC8_Sleep(void)

説明: これは、コンポーネントのスリープを準備するのに推奨される API です。VDAC8_Sleep() API は、現在のコンポーネントの状態を保存します。次に VDAC8_Stop() 関数、VDAC8_SaveConfig() 関数の順に呼び出して、ハードウェア設定を保存します。

CyPmSleep() または CyPmHibernate() 関数の前に VDAC8_Sleep() 関数を呼び出します。パワーマネジメント関数については、PSoC Creator *System Reference Guide (システム リファレンス ガイド)* を参照してください。

パラメータ: なし

返り値: なし

副作用: なし

void VDAC8_Wakeup(void)

説明: これは、コンポーネントを VDAC8_Sleep() が呼び出されたときの状態に復元するための推奨 API です。VDAC8_Wakeup() 関数は、設定を復元するために VDAC8_RestoreConfig() 関数を呼び出します。VDAC8_Sleep() 関数が呼び出される前にコンポーネントがイネーブルになっている場合は、VDAC8_Wakeup() 関数がコンポーネントを再度イネーブルにします。

パラメータ: なし

返り値: なし

副作用: 最初にVDAC8_Sleep() または VDAC8_SaveConfig() 関数を呼び出さないで VDAC8_Wakeup() 関数を呼び出すと、予期しない動作が発生する場合があります。

void VDAC8_Init(void)

説明: カスタマイザの [Configure (設定)] ダイアログの設定に従って、コンポーネントを初期化または復元します。VDAC8_Start() API が VDAC8_Init() 関数を呼び出すので、この関数を呼び出す必要はありません。これはコンポーネントの動作を開始する際に推奨される方法です。

パラメータ: なし

返り値: なし

副作用: すべてのレジスタが初期値に設定されます。これによって、コンポーネントが再初期化されます。レジスタの現在値以外の他の新しい値を設定する場合は、VDAC8_Init() 関数呼び出しには、VDAC8_SetValue() を呼び出す必要があります。

void VDAC8_Enable(void)

説明: ハードウェアの使用を開始し、コンポーネントの動作を開始します。VDAC8_Start() API が VDAC8_Enable() 関数を呼び出すので、この関数を呼び出す必要はありません。これはコンポーネントの動作を開始する際に推奨される方法です。

パラメータ: なし

返り値: なし

副作用: なし

void VDAC8_SaveConfig(void)

説明: この関数は、コンポーネントの設定と保持されないレジスタを保存します。この関数は、[Configure] (設定) ダイアログで定義されている、または該当する API で変更される、現在のコンポーネント パラメータ値も保存します。この関数は、VDAC8_Sleep()に呼び出されます。

パラメータ: なし

返り値: なし

副作用: なし

void VDAC8_RestoreConfig(void)

説明: この関数は、コンポーネントの設定と非保持レジスタを復元します。また、この関数はコンポーネントのパラメータ値を VDAC8_Sleep() 関数を呼び出す前の状態に復元します。

パラメータ: なし

返り値: なし

副作用: VDAC_Sleep() の呼び出し前にこの関数を呼び出すことで、予期しない動作を引き起こす場合があります。

DMA Wizard (DMA ウィザード)

VDAC8 コンポーネントは DMA 要求信号の実装は必要ありません。通常の使用は信号生成です。VDAC8 コンポーネントへのデータ レートは外部から制御されます。DMA Wizard を使って次のように DMA 処理を設定できます。

DMA ウィザードの DMA ソース/デスティネーションの名前	方向	DMA要求信号	DMA 要求タイプ	説明
VDAC8_Data_PTR	Destination (転送先)	該当せず	該当せず	DAC 値(0 ～ 255) を格納

ファームウェア ソースコードのサンプル

PSoC Creator は、[Find Example Project（サンプルプロジェクトを検索）] ダイアログに多数のサンプルプロジェクトを提供しており、そこには回路図およびサンプル コードが含まれています。コンポーネント固有の例を見るには、[Component Catalog] または回路図に置いたコンポーネント インスタンスからダイアログを開きます。一般例については、[Start Page] または **[File（ファイル）]** メニューからダイアログを開きます。必要に応じてダイアログにある **Filter Options** を使用し、選択できるプロジェクトのリストを絞り込みます。

詳しくは、PSoC Creator ヘルプの「Find Example Project（サンプルプロジェクトを検索）」を参照してください。

機能説明

VDAC8 として使用する場合は、viDAC8 アナログ ブロックを電圧 DAC として設定して、電圧ソースとして使用することができます。

VDAC として使用する場合、出力は、基準電圧を必要とする用途に対応する 8 ビット デジタル・アナログ変換電圧です。この場合、基準ソースは VREF(DAC) と呼ばれるアナログ基準ブロックの電圧基準です。
DACx_CR0 [4] レジスタを設定して電圧モードで動作するように DAC を設定することができます。このモードには、DACx_CR0[3:2] レジスタで選択される 2 つの出力範囲があります。

- 0 V ～ 1.024 V
- 0 V ～ 4.096 V

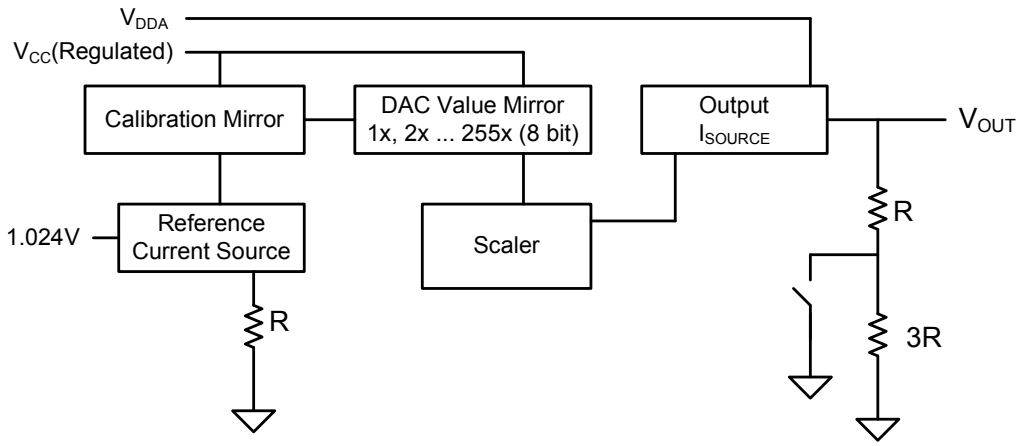
両方の出力範囲には 255 のステップ数があります。VDAC は、電流出力DACを駆動し、抵抗を介して電圧が出力されます。出力はバッファされていない為、、DACの DC 電流の変化によって出力レベルが変化します。したがって、このモードでは、出力に接続する負荷は容量性負荷となります。

VDAC は最大 1 Mspsに対応します。さらに、DAC では、4-V モードは 1-V モードより遅いです。V_{SSA} への抵抗負荷が 4 倍大きいためです。4-V モードでは、VDAC は最大 250 ksps となります。

ブロックダイアグラムと設定

図 1 は VDAC8 コンポーネントのブロック図です。

図 1. VDAC8 ブロック図



レジスタ

コンポーネントで提供される機能は、共通のランタイム関数で提供されます。次のレジスタ・リファレンスは上級者向けの簡易説明です。VDAC8_Data レジスタは API を使用せず、データを DAC へ直接書き込むために使用することができます。これは CPU または DMA のいずれかを利用する際に便利です。

VDAC8_CR0

ビット	7	6	5	4	3	2	1	0
値	RSVD			モード	Range[1:0]		hs	RSVD

- mode: DAC を電圧モードまたは電流モードのいずれかに設定します
- Range[1:0]: DAC 範囲の設定
- hs: データ スピードを設定します

VDAC8_CR1

ビット	7	6	5	4	3	2	1	0
値	RSVD		mx_data	reset_udb_en	mx_idir	idirbit	Mx_ioff	ioffbit

- mx_data: データ ソースを選択します

- reset_udb_en: DAC リセット イネーブル
- mx_idir: DAC 電流の方向コントロールの Mux 選択
- idirbit: DAC 電流の方向のレジスタ ソース
- mx_off: DAC 電流オフコントロールの Mux 選択
- ioffbit: DAC 電流オフのレジスタ ソース

VDAC8_DATA

ビット	7	6	5	4	3	2	1	0
値	Data[7:0]							

- Data[7:0]: DAC データ レジスタ

PSoC 3 の DC および AC 電気的特性

次の値は特性データに基づいています。特記されていない限り、仕様は–40°C ≤ T_A ≤ 85 °C および T_J ≤ 100 °C で有効です。下の表で別途指定されている場合を除き、すべての通常の値は T_A = 25 °C、V_{DDA} = 5.0 V、アナログ接地の基準出力 (V_{SSA})、fast モードです。

VDAC8 DC 電気的特性

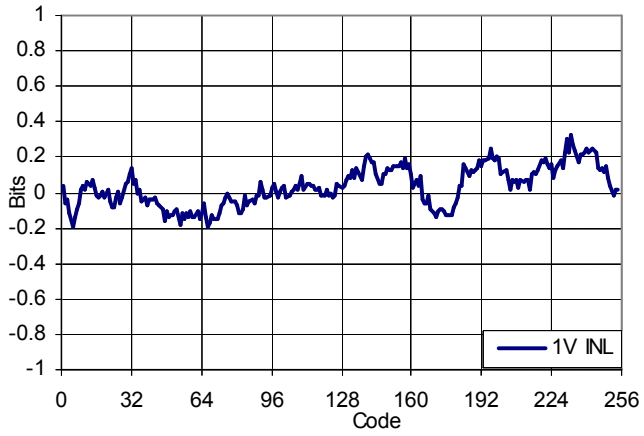
パラメータ	説明	条件	Min	Typ	Max	単位
	分解能		–	8	–	ビット
INL1	積分非直線性	1-V スケール	–	±2.1	±2.5	LSB
DNL1	微分非直線性	1-V スケール	–	±0.3	±1	LSB
R_{OUT}	出力抵抗値	1-V スケール	–	4	–	kΩ
		4-V スケール	–	16	–	kΩ
V_{OUT}	出力電圧範囲、code = 255	1-V スケール	–	1	–	V
		4-V スケール、V_{DDA} = 5 V	–	4[1]	–	V
	単調増加性		–	–	あり	–
V_{OS}	セロスケール エラー		–	0	±0.9	LSB
FSGainErr	フルスケール ゲイン エラー	1-V スケール	–	±1.6	±2.5	%

[1] 5 V 以下の V_{DDA} 電圧の場合、出力は出力電圧以下の仕様にのみ準拠します (V_{DDA} – 1 V)。

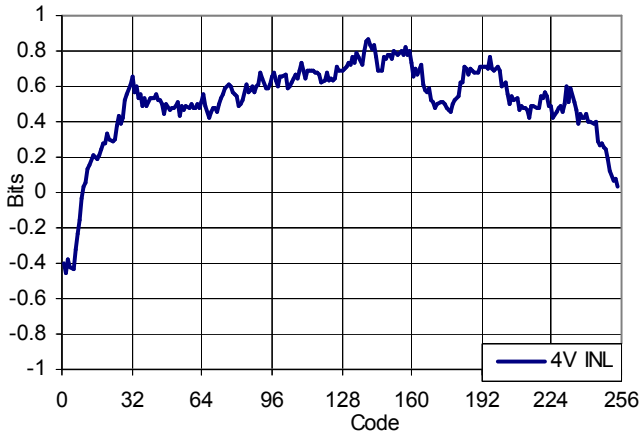
パラメータ	説明	条件	Min	Typ	Max	単位
		4-V スケール	–	±1.5	±2.5	%
TCGainErr	温度係数、ゲイン エラー	1-V スケール	–	–	0.02	%FSR/°C
		4-V スケール	–	–	0.02	%FSR/°C
I_{DD}	動作電流	Slow モード	–	–	100	µA
		Fast モード	–	–	500	µA

図

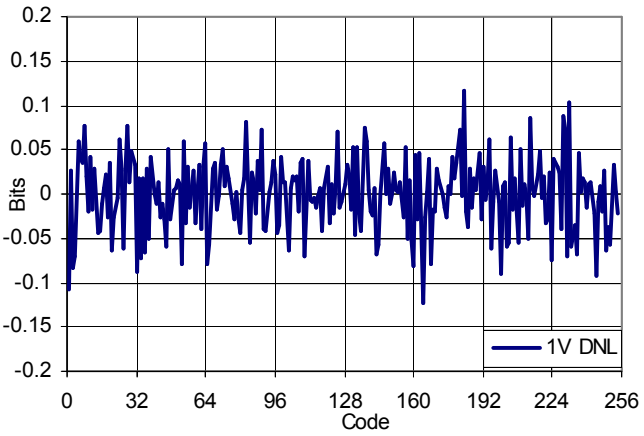
INL 対 DAC コード、1.0-V 範囲



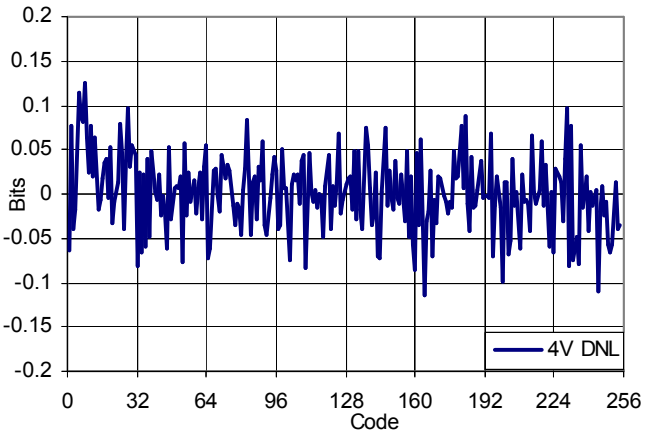
INL 対 DAC コード、4.0-V 範囲



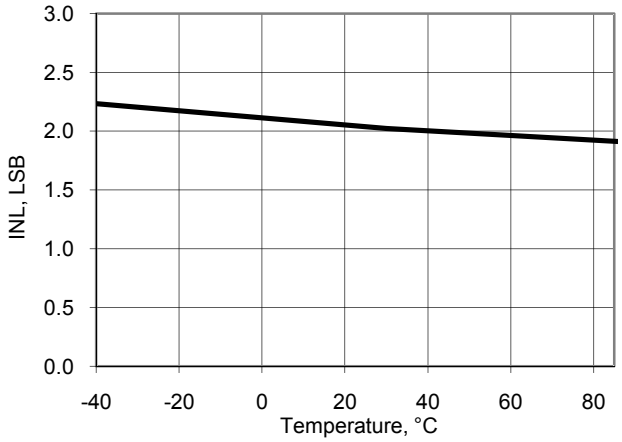
標準 DNL 対 DAC コード、1.0-V 範囲



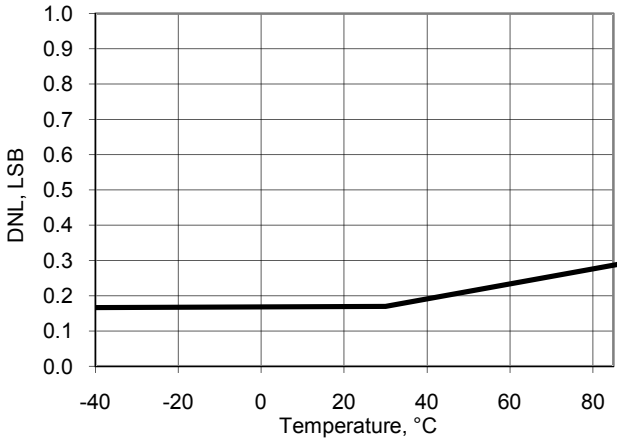
DNL 対 DAC コード、4.0-V 範囲



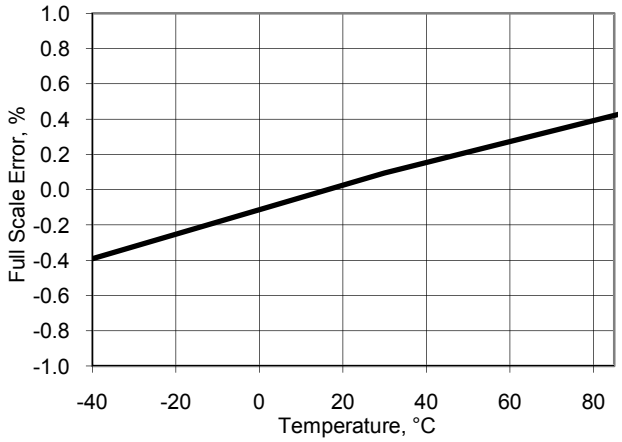
VDAC INL 対 温度、1-V モード



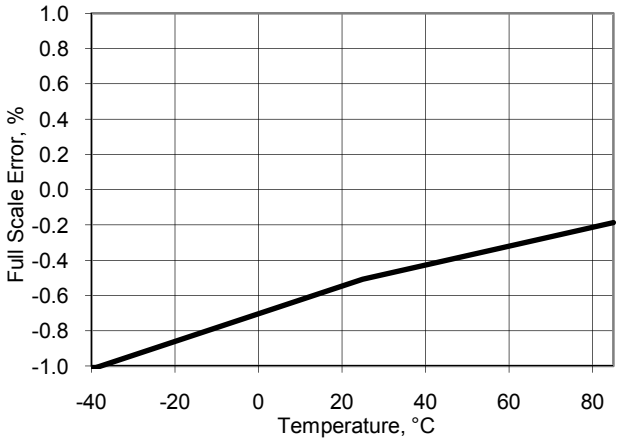
VDAC DNL 対 温度、1-V モード



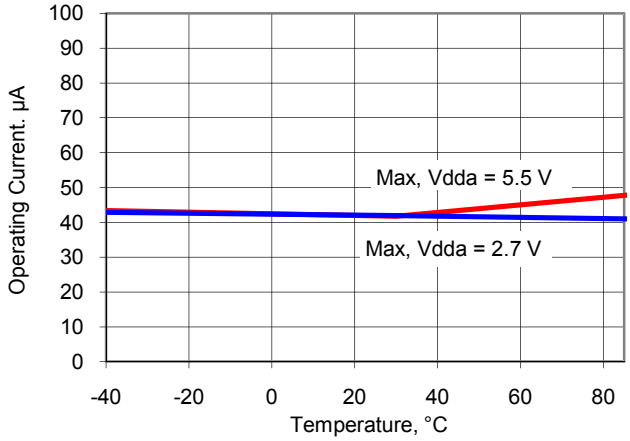
VDAC フルスケール エラー 対 温度、1-V モード



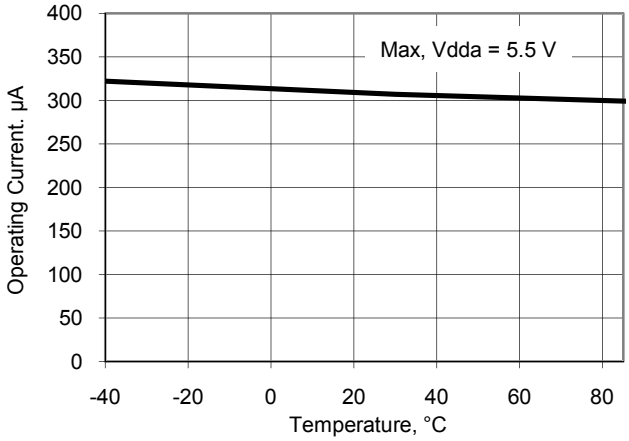
VDAC フルスケール エラー 対 温度、4-V モード



VDAC 動作電流 対 温度、1-V モード、Slow モード



VDAC 動作電流 対 温度、1-V モード、Fast モード

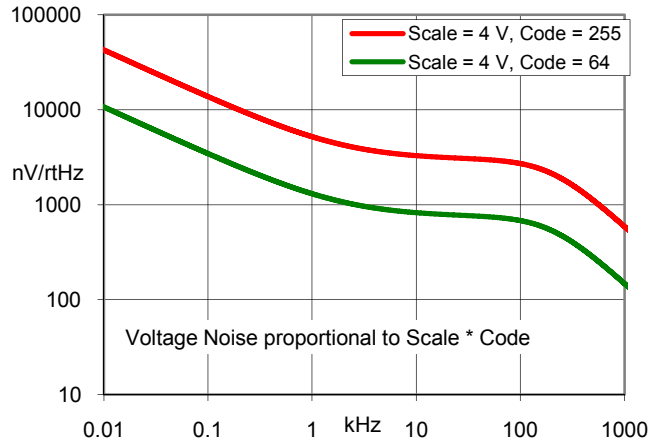


VDAC8 AC 電気的特性

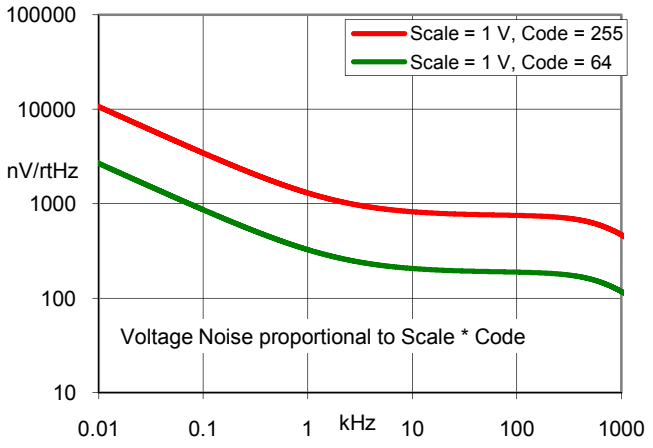
パラメータ	説明	条件	Min	Typ	Max	単位
F_{DAC}	更新レート	1-V スケール	–	–	1000	ksps
		4-V スケール	–	–	250	ksps
TsettleP	0.1% までのセトリング タイム、ステップ 25% ～ 75%	1-V スケール、C_{LOAD} = 15 pF	–	0.45	1	µs
		4-V スケール、C_{LOAD} = 15 pF	–	0.8	3.2	µs
TsettleN	0.1% までのセトリング タイム、ステップ 75% ～ 25%	1-V スケール、C_{LOAD} = 15 pF	–	0.45	1	µs
		4-V スケール、C_{LOAD} = 15 pF	–	0.7	3	µs
SRP	スルー レート、ステップ 10% ～ 90%	1-V スケール、C_{LOAD} = 15 pF	–	0.3	0.5	µs
		4-V スケール、C_{LOAD} = 15 pF	–	0.5	1.3	µs
SRN	スルー レート、ステップ 90% ～ 10%	1-V スケール、C_{LOAD} = 15 pF	–	0.3	0.5	µs
		4-V スケール、C_{LOAD} = 15 pF	–	0.3	1.3	µs
Vn4V	ノイズ密度[2]	4-V スケール、コード 255	–	3	–	µV/rtHz
Vn1V	ノイズ密度[2]	1-V スケール、コード 255	–	750	–	nV/rtHz

図

ノイズ レベル、スケール 4 V



ノイズ レベル、スケール 1 V



[2] 出力ノイズはコード値に比例します。

PSoC 5 DC/ AC 電気的特性

次の値は特性データに基づいています。特記されていない限り、仕様は –40°C ≤ T_A ≤ 85 °C および T_J ≤ 100 °C で有効です。下の表で別途指定されている場合を除き、すべての標準値は T_A = 25 °C、V_{DDA} = 5.0 V、アナログ接地の基準出力 (V_{SSA})、fast モードで有効です。

VDAC8 DC 電気的特性

パラメータ	説明	条件	Min	Typ	Max	単位
	分解能		–	8	–	ビット
INL1	積分非直線性	1-V スケール	–	±2.1	±2.5	LSB
DNL1	微分非直線性	1-V スケール	–	±0.3	±1	LSB
R_{OUT}	出力抵抗値	1-V スケール	–	4	–	kΩ
		4-V スケール	–	16	–	kΩ
V_{OUT}	出力電圧範囲、code = 255	1-V スケール	–	1.02	–	V
		4-V スケール、V_{DDA} = 5 V	–	4.08[3]	–	V
	単調増加性		–	–	あり	–
V_{OS}	セロスケール エラー		–	0	±0.9	LSB
Eg	ゲイン エラー	1-V スケール	–	–	±5	%
		4-V スケール	–	–	±5	%
TC_Eg	温度係数、ゲイン エラー	1-V スケール	–	–	0.03	%FSR/°C
		4-V スケール	–	–	0.03	%FSR/°C
I_{DD}	動作電流	4-V Slow モード	–	–	100	µA
		4-V Fast モード	–	–	500	µA
		1-V Slow モード			300	µA
		1-V Fast モード			600	µA

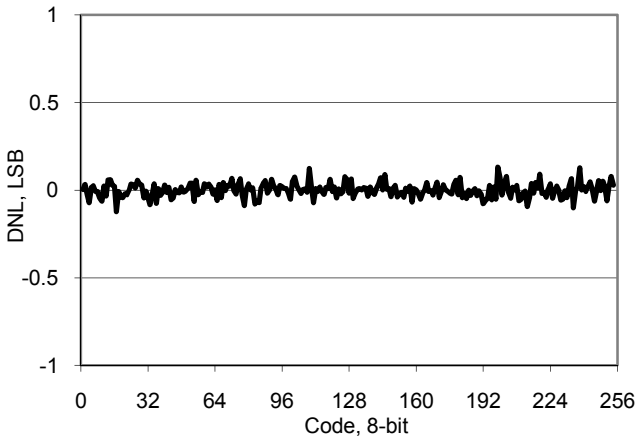
[3] 5 V 以下の V_{DDA} 電圧の場合、出力は出力電圧以下の仕様にのみ準拠します (V_{DDA} – 1 V)。

図

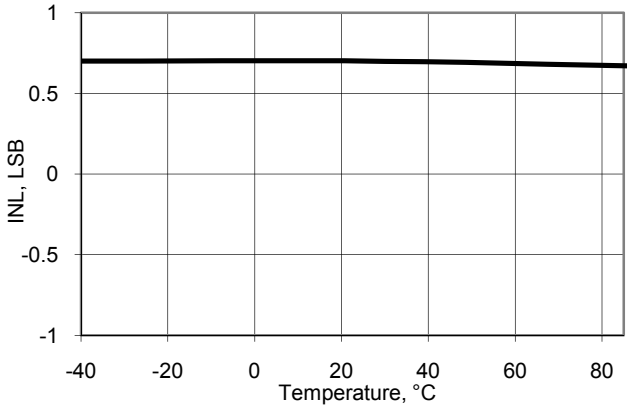
VDAC INL 対 入力コード、1-V モード



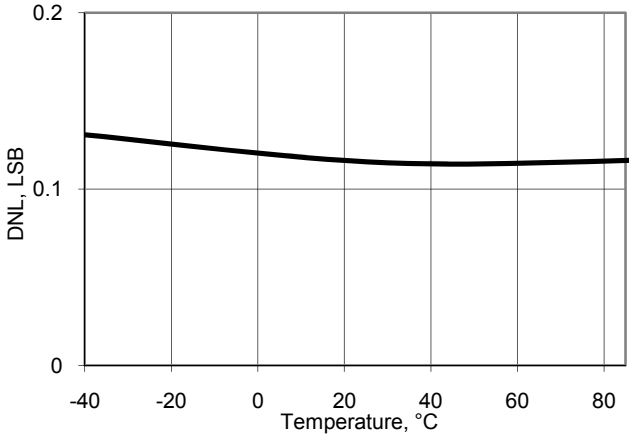
VDAC DNL 対 入力コード、1-V モード



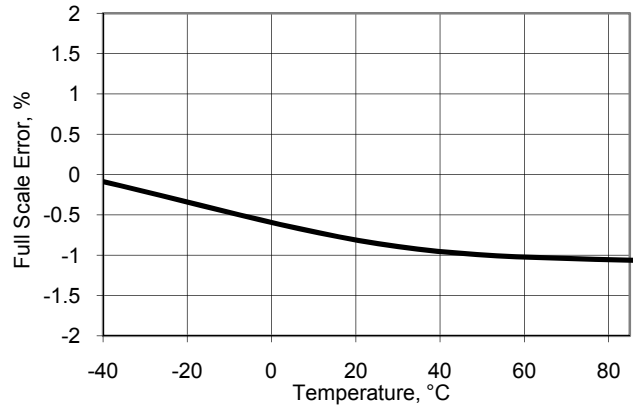
VDAC INL 対 温度、1-V モード



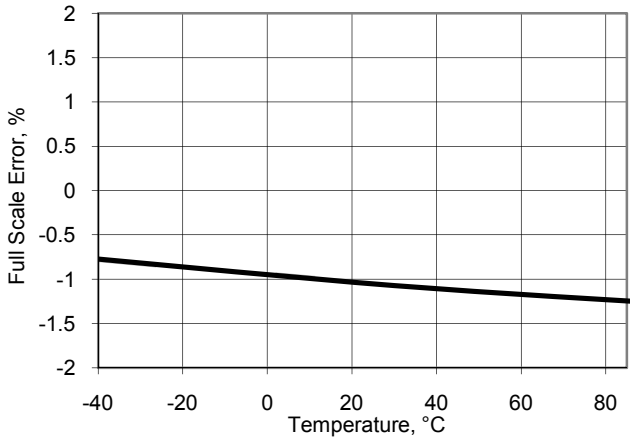
VDAC DNL 対 温度、1-V モード



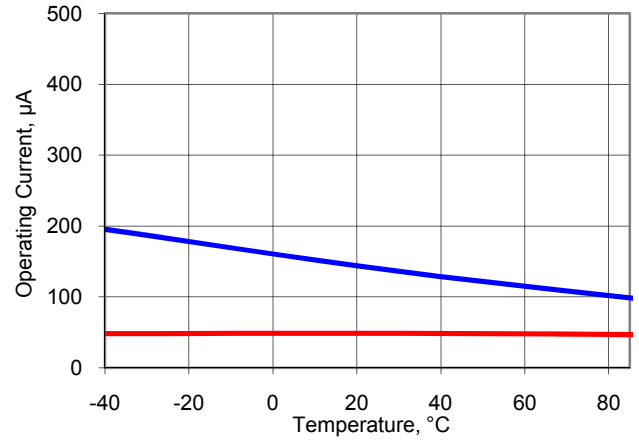
VDAC フルスケール エラー 対 温度、1-V モード



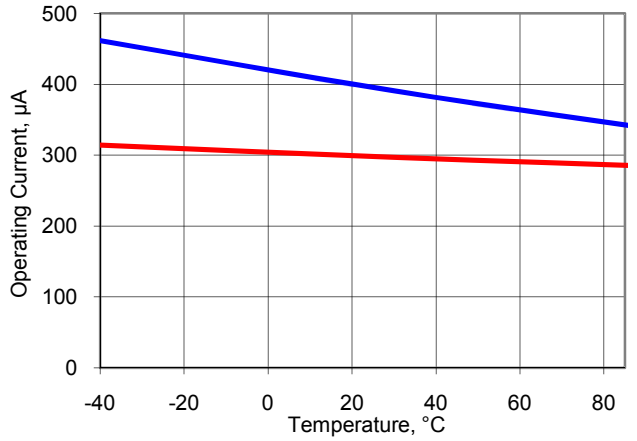
VDAC フルスケール エラー 対 温度、4-V モード



VDAC 動作電流 対 温度、1-V モード、Slow モード



VDAC 動作電流 対 温度、1-V モード、Fast モード

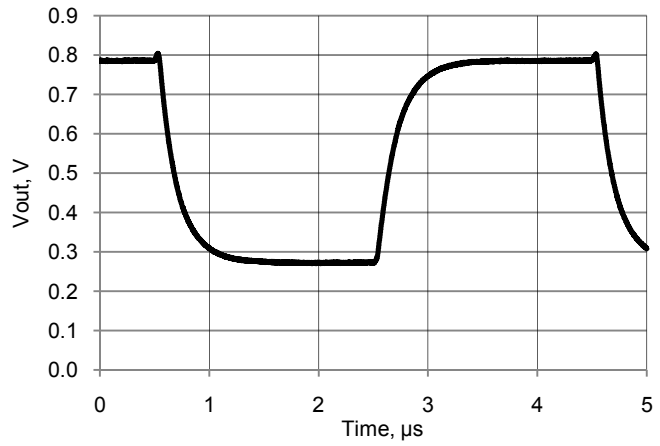


VDAC8 AC 電気的特性

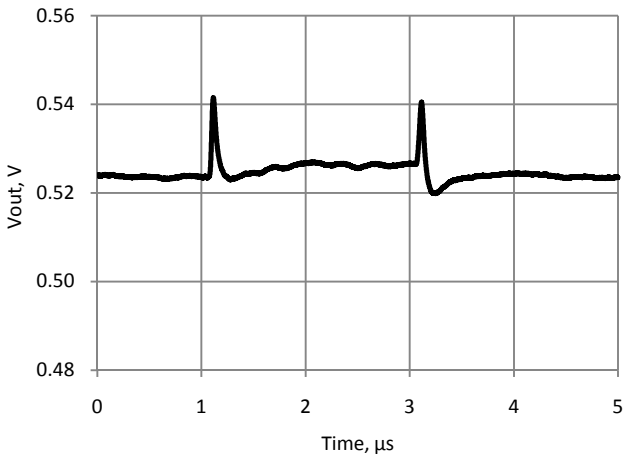
パラメータ	説明	条件	Min	Typ	Max	単位
F_{DAC}	更新レート	1-V スケール	–	–	1000	ksps
		4-V スケール	–	–	250	ksps
TsettleP	0.1% までのセトリング タイム、ステップ 25% ～ 75%	1-V スケール、C_{LOAD} = 15 pF	–	0.45	1	µs
		4-V スケール、C_{LOAD} = 15 pF	–	0.8	4	µs
TsettleN	0.1% までのセトリング タイム、ステップ 75% ～ 25%	1-V スケール、C_{LOAD} = 15 pF	–	0.45	1	µs
		4-V スケール、C_{LOAD} = 15 pF	–	0.7	4	µs
	電圧ノイズ	範囲 = 1 V、fast モード、V_{DDA} = 5 V, 10 kHz	–	750	–	nVsqrtHz

図

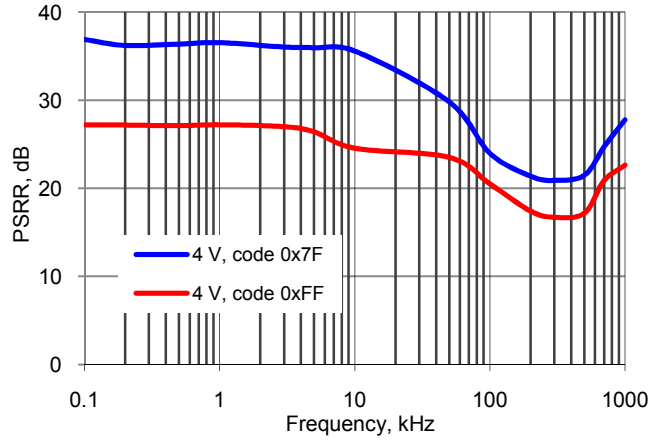
VDAC ステップ応答、コード 0x40 ～ 0xC0、1-V モード、Fast モード、V_{DDA} = 5 V



VDAC グリッチ応答、コード 0x7F ～ 0x80、1-V モード、Fast モード、V_{DDA} = 5 V



VDAC PSRR 対周波数



専門用語

積分非直線性（INL）

INL (積分比直線性)は、DAC の動作範囲上の最良適合直線からの LSB 単位の最大偏差。

微分非直線性（DNL）

DNL (微分非直線性) は、2 つの隣接コード間の測定した変化と理想的な 1LSB 変化の差です。この VDAC はデザインにより単調増加性が保証されています。出力は「サーモメータ エンコード」です。各連続ステップは、以前に有効にされた出力ソースに合計される別の出力ソースをオンにして行われます。

単調増加性

A DAC は、出力が増加するか、デジタル コードの入力値の増加と同じ状態である場合は、単調増加性として定義されます。VDAC8 コンポーネントは、電圧および温度のフル動作範囲において単調増加性。

セロスケール エラー

ゼロ スケール エラーは、コード 0x00 の測定値とコード 0x00 の最良適合直線の値との差です。

フルスケール ゲイン エラー

フルスケール ゲイン エラーは最大コードでの測定値と公称値との差です。最大値は、コード = 255 での 1.020 V または 4.080 V のいずれかです。

フルスケール ゲイン温度係数 (TC)

フルスケール ゲイン温度係数は、温度の変動によるフルスケール値 (最大コード 0xFF) の変動です。低ゲインでのゲイン変動はコード値に比例します。

電源電圧変動除去比 (PSRR)

電源電圧変動除去比は、VDAC の出力のアイソレーションを電源から測定します。

セトリング タイム

セトリング タイムは、特定のデジタル入力変動の特定レベルに安定させるために出力に必要な期間です。

スルー レート

スルー レートは、VDAC 出力の最大変動率です。スルー レートは、フルスケール値の 10 % ～ 90 %が測定されます。

グリッチ振幅

グリッチ振幅は、入力コードがミッドスケール (0x7F ～ 0x80) での単一カウントを変更する場合に、出力に投入されるパルス ピーク振幅です。パルスは、データの変動の前後のスタティック値の差より大きいです。

電圧ノイズ

電圧ノイズは、VDAC の出力抵抗と、VDAC の電流ノイズ x 出力抵抗の合計です。このノイズは、コード値の機能に応じて変わります。

コンポーネントの変更

ここでは、過去のバージョンからコンポーネントに加えられた主な変更を示します。

バージョン	変更の説明	変更の理由 / 影響
1.70.a	PSoC 5 の DC と AC 電気的特性データを追加。	
	データシートのマイナーな編集と更新	
1.70	PSoC 5 の VDAC8_Stop() API を変更	PSoC 5 と併用する場合、コンポーネントの停止時に独立したアナログ信号に影響を与えることを防ぐには、変更が必要です。
	VDAC カスタマイザを更新。	▪ VDAC レイアウトを IDAC レイアウトと同じにするため。 ▪ データ ソースが DAC Bus として選択された場合、ストローブ モードを External 設定に強制するため。
1.60	GUI Configuration Editor を追加	以前の設定ウィンドウは、使用しやすい十分な情報がありませんでした。
	データシートに特性データを追加	
	データシートのマイナーな編集と更新	
1.50	Sleep/Wakeup (スリープ/ウェイクアップ) と Init/Enable (初期化/イネーブル) API を追加しました。	ローパワー モードをサポートし、ほとんどのコンポーネントの初期化と有効化の制御を分離する共通インターフェースを提供するため。
	コンポーネントに DMA 能力ファイルを追加しました。	このファイルにより、PSoC Creator の DMA Wizard ツールでサポートされるようになります。

Copyright © 2005-2012 Cypress Semiconductor Corporation 本文書に記載される情報は、予告なく変更される場合があります。Cypress Semiconductor Corporation は、サイプレス製品に組み込まれた回路以外のいかなる回路を使用することに対しても一切の責任を負いません。特許又はその他の権限下で、ライセンスを譲渡又は暗示することもありません。サイプレス製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、又は安全の用途のために使用することを保証するものではなく、また使用することを意図したものでもありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことを合理的に予想される、生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を提供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

PSoC Designer™ 及び Programmable System-on-Chip™ は、Cypress Semiconductor Corp.の商標、PSoC® は同社の登録商標です。本文書で言及するその他全ての商標又は登録商標は各社の所有物です。

全てのソースコード(ソフトウェア及び/又はファームウェア)は Cypress Semiconductor Corporation (以下「サイプレス」)が所有し、全世界(米国及びその他の国)の特許権保護、米国の著作権法並びに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によるライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであって、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタムソフトウェア及び/又はカスタムファームウェアを作成する目的に限って、サイプレスのソースコードの派生著作物を複製、使用、変更、そして作成するためのライセンス、並びにサイプレスのソースコード及び派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソースコードを複製、変更、変換、コンパイル、又は表示することは全て禁止されます。

免責条項:サイプレスは、明示的又は黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性又は特定目的への適合性の黙示的な保証が含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品又は回路を適用又は使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を提供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレスソフトウェアライセンス契約によって制限され、かつ制約される場合があります。