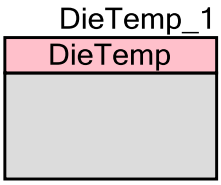
動作ダイ温度(DieTemp)
1.70



特長

- 精度: ±5 °C
- 範囲: –40 °C ～ +140 °C (0xFFD8 ～ 0x008C)
- ブロッキング、非ブロッキング API 対応
- データ コード 1005 の PSoC 3 ES2 シリコンと 、それ以降の Production PSoC 3 シリコンのみに対応
- PSoC 5 シリコンには対応していません

概要

ダイ温度 (DieTemp) コンポーネントには、ダイの温度を取得する API が備わっています。System Performance Controller (SPC) はダイの温度を取得するために使用されます。API には、ブロッキングと非ブロッキングのコールが含まれています。

DieTemp の用途

デバイスのダイ温度を測定する場合に DieTemp コンポーネントを使用します。

I/O 接続

DieTemp コンポーネントに I/O 接続はありません。ソフトウェア コンポーネントのみの API です。

コンポーネント・パラメータ

DieTemp には、標準インスタンス名と組み込みパラメータ以外の設定できるパラメータはありません。

リソース

アナログブロック	デジタルブロック					API メモリ（Bytes）		ピン数（外部I/O）
	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	Flash	RAM	
N/A	N/A	N/A	N/A	N/A	N/A	289	0	N/A

Application Programming Interface

Application Programming Interface (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 はインスタンス名「DieTemp_1」をデザイン上のコンポーネントの最初のインスタンスに割り当てます。コンポーネントのインスタンス名称は、識別子の文法ルールに従って固有の名前に変更できます。インスタンス名は、すべてのグローバル関数名、変数名、定数名の接頭語になります。便宜上、以下の表ではインスタンス名として「DieTemp」を使っています。

関数名	説明
DieTemp_Start()	SPC コマンドを起動してダイ温度を取得します
DieTemp_Stop()	温度の取得を停止します
DieTemp_Query()	温度コマンドが完了したかどうか SPC に問い合わせます
DieTemp_GetTemp()	コマンドセットにより温度を取得し、取得完了までブロックします

cystatus DieTemp_Start(void)

説明: コマンドとパラメータを SPC へ送信してダイ温度の取得を開始します。この関数は SPCが完了する前に戻ってきます。この関数の呼び出しは、常に DieTemp_Query() API の呼び出しと組み合わせてダイ温度取得を行う必要があります。

引数: なし

戻り値: CYRET_STARTED (SPC コマンドが正常に開始した場合。)
CYRET_UNKNOWN (SPC コマンドが失敗した場合。)
CYRET_LOCKED (SPC がビジーの場合。)

サイドエフェクト: なし

void DieTemp_Stop(void)

説明: 温度取得を停止します。

引数: なし

戻り値: なし

サイドエフェクト: なし

cystatus DieTemp_Query(int16 * temperature)

説明: DieTemp_Start() によって開始された SPC コマンドが終了したかどうかを確認します。コマンドが終了していなかった場合、温度データはまだ未取得で戻っていません。呼び出し元は、戻り値が CYRET_STARTED の間は関数をポーリングする必要があります。

正常に正しいダイ温度を取得するために、この関数は DieTemp_Start() API と共に使用する必要があります。

初めに DieTemp_Start() 、次に DieTemp_Query() というシーケンスで関数を呼び出し、初回に戻る値は不確かなので使用せず、同様のシーケンスをもう一度行いその値を使用する必要があります。

引数: int16 * temperature: 温度を摂氏で格納するアドレス

戻り値: CYRET_SUCCESS (温度コマンドが正常に終了した場合。)

CYRET_UNKNOWN (SPC エラーがあった場合。)

CYRET_STARTED (温度コマンドがまだ完了していない場合。)

CYRET_TIMEOUT (データ取得期間がタイムアウトした場合。)

サイドエフェクト: なし

cystatus DieTemp_GetTemp(int16 * temperature)

説明: コマンドとパラメータを SPC へ送信して、ダイ温度取得を開始し、エラーまたは正常に終了するまで待機します。この関数はブロッキング API です。

この関数はダイ温度を 2 回取得し、2回目の値を返します。これは初回取得値が不確かになるシリコンの問題に対処するためです。

引数: int16 * temperature: 温度を摂氏で格納するアドレス

戻り値: CYRET_SUCCESS (コマンドが正常に終了した場合。)

CYRET_TIMEOUT (コマンドタイム アウトの場合。)

DieTemp_Start() 、DieTemp_Query() の戻り値と同様です。

サイドエフェクト: なし

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。

DC/AC 電気特性

以下の値は、初期設定での期待される性能を示しております。

パラメータ	説明	条件	Min	Typical	Max	単位
	温度センサ精度	範囲: –40 °C ～ +85 °C	--	±5	--	°C

コンポーネント履歴

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	ダイ温度取得の問題を修正するために、2回ダイ温度を取得するように DieTemp_GetTemp() API を変更。	DieTemp_GetTemp() API は、最初の取得が成功しない場合でも温度を返していた。
1.60	パワーオンリセットのエラー出力を修正するために *DieTemp.c* ファイルの GetTemp API を変更。	パワーオンリセット時の DieTemp 出力が間違っていた。
1.50.a	データシートに特性データを追加	
	シリコン リビジョンとの互換性について知らせる情報をコンポーネントに追加。	このツールは互換性のないシリコン上でこのコンポーネントを使うと警告/エラーを返す。このコンポーネントは PSoC 3 ES2 とPSoC 5 の間で互換性がありません。
	データシートを更新	
1.50	*cydevice.h* から *cydevice_trm.h* へ切り替え。	*cydevice.h* ファイルが古く、PSoC Creator より提供されるAPIと生成コードが *cydevice_trm.h* に含まれていなかった。

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。