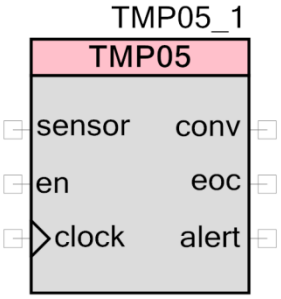
TMP05 接口组件

1.10



特性

- 可支持四个 TMP05 或 TMP06 数字温度传感器（仅采用菊花链方式相连）
- 连续运行方式与单触发运行方式
- 支持频率范围：100 至 500 kHz
- 支持温度范围：0 至 70 ℃

概述

TMP05 温度传感器接口组件简单易用，可仅通过**菊花链方式**连接模拟装置的 TMP05/06 数字温度传感器。设计人员可配置该接口组件，并通过以下两种方式的其中一种监控温度读数：1) 连续监控方式可让设计人员连续记录温度，采样率取决于温度传感器；2) 单触发方式可触发温度测量，速率可由用户控制。第一种方式用于温度发生突变且需要频繁监控的情况。每隔一段时间需采样一次温度测量值或必需尽量减小功耗的情况下，采用第二种方式。

注意：由于组件支持仅采用菊花链方式连接的数字温度传感器，所以必须采用菊花链连接方式配置设备，即使仅连接一个设备。

何时使用 TMP05 接口

TMP05 温度传感器接口组件用于热管理，要求监控多个远程子系统的温度。组件设计用于直接连接可生成 PWM 输出的 TMP05/06 温度传感器，该输出的占空比与环境温度直接相关。

组件运行方式

TMP05 温度传感器接口组件支持两种运行方式：连续方式与单触发方式。

1. 在连续方式下，温度测量值以最快速率出现，而该速率取决于传感器的转换时间。
2. 在单触发方式下，温度测量值的出现速率取决于应用固件。在该方式下，设计人员可设置温度监控速率。

输入/输出连接

本节介绍了 TMP05Intf 的各种输入和输出连接。

传感器 – 输入

就采用单个传感器而言，该输入应与 TMP05 温度传感器输出相接。上述输出是经脉冲宽度调制的数字信号，其占空比与温度成比例。

当多个 TMP05 温度传感器采用菊花链连接方式相连时，该输入应与菊花链中的最后一个 TMP05 温度传感器的输出相接。

当连接开漏输出 TMP06 传感器时，该信号需设上拉电阻，以确保正常运行。采用选定 PSoC GPIO 引脚上的内部上拉电阻或电路板上的外部上拉电阻可实现上述目的

en - 输入

有效高电平使能 将此引脚设为低电平，以禁用组件。

时钟 — 输入

时钟负责运行此模块。频率必须在 100 kHz 至 500 kHz。

conv – 输出

转换开始输出。为了启动监控温度，TMP05 传感器需要触发输入，在低电平和高电平时间有特定的时序限制。该输出向菊花链上第一个 TMP05 传感器提供了一个触发输入。

eoc – Output（输出）

结束转换输出。当新温度测量值可用时，在一个时钟周期内该信号脉冲为高电平有效。当传感器采用菊花链连接方式时，该信号说明已在所有传感器上进行测量。直至数据准备好供 API 获取（内部处理后）后才可激活 eoc 信号。

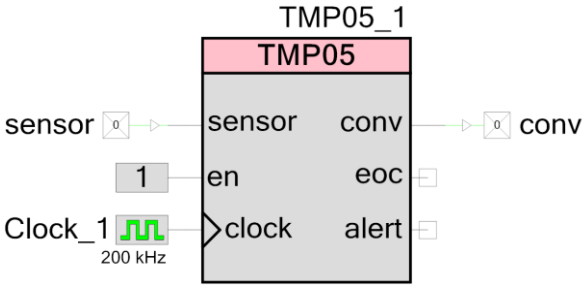
警报 – 输出

温度转换错误输出。当温度传感器测量暂停时，在一个时钟周期内该信号脉冲有效且为高电平。所连接的温度传感器中的一个或多个发生故障时，或者未与传感器连接时，可能会出现上述情况。

原理图宏信息

本节包括 TMP05 接口内原理图宏的专有信息。

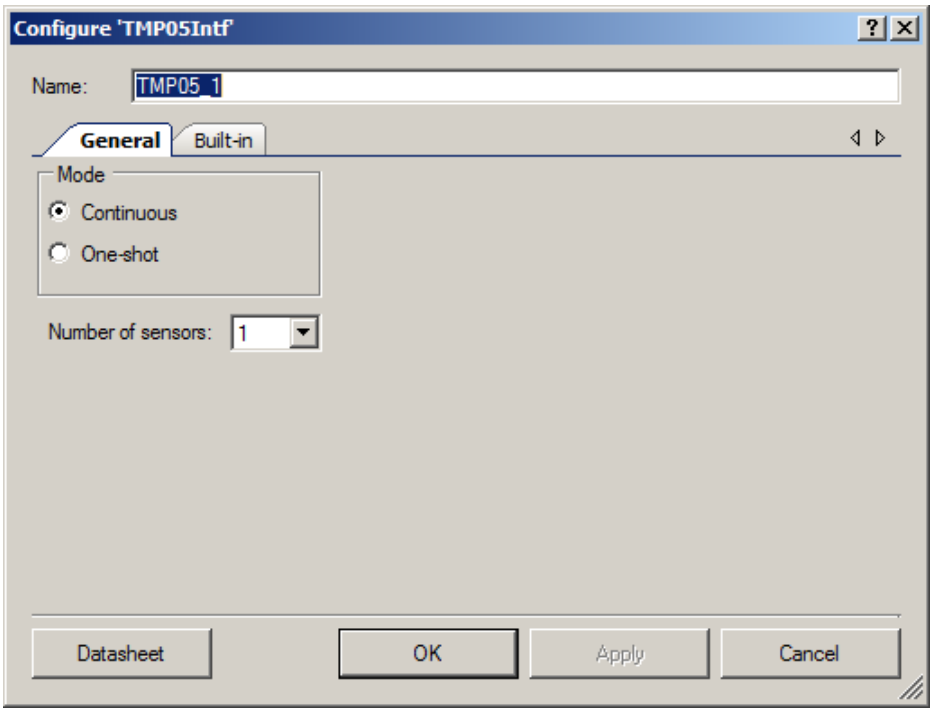
组件目录中的缺省的 TMP05 接口使用带缺省设置的 TMP05 接口组件的原理图宏。与时钟、高逻辑电平组件及数字输入/输出引脚相接。



组件参数

将一个 TMP05 接口组件拖放到设计上，然后双击，打开配置对话框。该对话框有一个选项卡，可引导您完成 TMP05 接口组件的设置过程。

一般选项卡



General（一般）选项卡提供以下参数。

连续

该参数用于规定组件的运行方式。该参数是一个布尔值，用于在连续和单触发运行方式之间作出选择。该参数真有效值和假有效值。默认设置为真，可启用连续运行方式。

传感器个数

该参数用于规定采用菊花链连接方式的 TMP05 温度传感器个数。当前，组件可支持多达四个 TMP05 温度传感器。此参数的有效范围为 1-4。默认设置为 1。

时钟选择

此组件中没有内部时钟。您必须附加时钟源。

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 “TMP05_1” 分配给指定设计中组件的第一个实例。您可以将其重命名为遵循标识符语法规则的任何唯一值。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。出于可读性考虑，下表中使用的实例名称为 “TMP05”。

函数	说明
TMP05_Start()	启动此组件
TMP05_Stop()	停止此组件
TMP05_Init()	初始化此组件
TMP05_Enable()	启用此组件
TMP05_Trigger()	触发已连接的 TMP05 传感器，以根据运行方式开始温度测量
TMP05_GetTemperature()	计算温度值（单位：℃）
TMP05_SetMode()	设置组件运行方式
TMP05_DiscoverSensors()	自动检测与组件采用菊花链连接的温度传感器个数
TMP05_ConversionStatus()	返回至温度转换的当前状态（繁忙、完成或错误）
TMP05_SaveConfig()	在输入低功耗模式前保存该组件的当前状态
TMP05_RestoreConfig()	在低功耗模式唤醒后，恢复该组件之前的状态
TMP05_Sleep()	将该组件置于低功耗模式
TMP05_Wakeup()	在低功耗模式下唤醒该组件

全局变量

变量	说明
TMP05_initVar	initVar 变量用于说明此组件的初始配置。此变量前面加有组件名称。此变量被初始化为 0，并在第一次调用 TMP05_Start() 时设置为 1。这实现了组件初始化，而无需重新初始化 TMP05_Start() 例程中的所有后续调用。 如果要求对该组件进行重新初始化，应在调用 TMP05_Init() 和 TMP05_Enable() 之前调用 TMP05_Stop() 例程。
TMP05_busyFlag	usyFlag 变量用于说明组件测量情况。在组件初始化后或者完成测量后，将该变量被初始化为 0，然后在调用 TMP05_Trigger API 后将该变量设为 1。

void TMP05_Start(void)

说明： 启动此组件 如果组件之前未初始化，则调用 TMP05_Init() API。调用使能 API。

参数： None

返回值： None

副作用： None

void TMP05_Stop (void)

说明： 禁用并停止此组件

参数： None

返回值： None

副作用： None

void TMP05_Init(void)

说明： 初始化此组件

参数： None

返回值： None

副作用： None

void TMP05_Enable(void)

说明： 启用此组件

参数： None

返回值： None

副作用： None

void TMP05_Trigger (void)

说明： 在 conv 端子上提供有效的选通/触发输出。

参数： None

返回值： None

副作用： None

int16 TMP05_GetTemperature (uint8 SensorNum)

说明： 计算温度值（单位：℃）

参数： uint8 SensorNum - TMP05 传感器个数：0..3

返回值： 温度是指定传感器的 1/100 ℃ 。

副作用： None

void TMP05_SetMode （uint8 方式）

说明： 设置组件运行方式

参数： uint8 方式 - 运行方式：

- MODE_CONTINUOUS - 连续方式
- MODE_ONESHOT - 单触发方式

返回值： None

副作用： None

uint8 TMP05_DiscoverSensors (void)

说明： 当相连温度传感器的个数为变量时可提供的API。自动检测与组件采用菊花链连接的温度传感器个数。启用算法前检查实际相连的传感器个数是否符合组件定制器的基本选项卡内 NumSensors 参数设置。如果不符合，在假设连接的传感器少了 1 个的情况下进行重试。重复此程序，直至获知已连接传感器的实际个数。

确认是否安装传感器的时间：该算法每个迭代每个传感器耗时几百毫秒。为了限制传感时间，减小该组件定制器的基本选项卡内的 NumSensors 参数设置，以将系统内可能安装的传感器个数最大化。

参数： 无

返回值： uint8 - 表示实际相连的传感器个数 (0..4)

副作用： None

uint8 TMP05_ConversionStatus (void)

说明： 使固件与硬件同步

参数： 无

返回值： uint8 状态代码：

- STATUS_IN_PROGRESS - 转换进行中
- STATUS_COMPLETE - 转换完成
- STATUS_ERROR - 传感器错误

副作用： None

void TMP05_SaveConfig (void)

说明： 保存 TMP05 非保留寄存器的用户配置。通过 TMP05_Sleep() 调用该例程，以在进入睡眠状态之前保存组件配置。

参数： None

返回值： None

副作用： None

void TMP05_RestoreConfig (void)

说明： 恢复 TMP05 非保留寄存器的用户配置。通过 TMP05_Wakeup() 调用该例程，以在退出睡眠状态时恢复组件配置。

参数： None

返回值： None

副作用： None

void TMP05_Sleep (void)

说明： 停止 TMP05 操作，保存用户配置与 TMP05 的启用状态。

参数： None

返回值： None

副作用： None

void TMP05_Wakeup (void)

说明： 恢复用户配置和启用状态。

参数： None

返回值： None

副作用： None

MISRA 合规性

本节介绍了本组件与 MISRA-C:2004 的合规和偏差情况。定义了两种类型的偏差：

- 项目偏差 - 适用于所有 PSoC Creator 组件的偏差
- 特定偏差 - 仅适用于此组件的偏差

本节提供了有关组件特定偏差的信息。*系统参考指南*的“MISRA 合规性”章节中介绍项目偏差以及有关 MISRA 合规性验证环境的信息。

TMP05 温度传感器接口组件没有任何特定偏差。

该组件配有以下嵌入式构件：中断、状态寄存器、控制寄存器 MISRA 合规性与特定偏差的相关信息请参见相应组件数据手册。

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的例子项目。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 Start Page（开始页）或 **File**（文件）菜单中的对话框。根据需要，使用对话框中的 **Filter Options**（筛选选项）可缩小可选项目的列表。

有关更多信息，请参见 PSoC Creator 帮助中的“Find Example Project（查找示例项目）”主题。

中断服务子程序

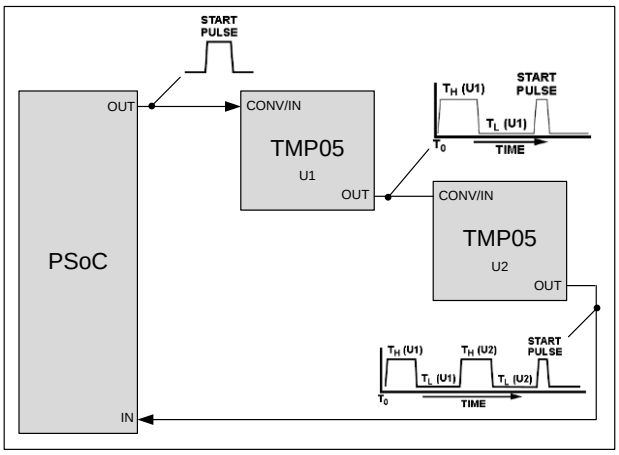
在已生成的中断服务例程位置,用户输入自定义 ISR 代码时的 C 语言示例如下：

```
/* `#START TMP05_INT_HEADER` */

/* `#END`  */
```

功能描述

该组件采用一个输入引脚和一个输出引脚通过菊花链方式连接单个或多个 TMP05 传感器（如下所示）。



TMP05 传感器有三种运行方式：

- 连续转换方式
- 菊花链方式
- 单触发方式

通电时会采样三态功能控制输入引脚 (FUNC) 可确定该装置的运行方式。将 FUNC 引脚设为高电平状态，这样便可采用菊花链方式连接多个 TMP05。在该方式下，可连接多个 TMP05 温度传感器，这样可使 TMP05 接口组件通过同一个两线接口读出所有传感器。

在这种配置下，TMP05 接口组件会产生一个“启动”脉冲，以开始温度至 PWM 转换的新周期。第一个 TMP05 传感器的输出以其自带的 PWM 输出作为开始，紧接着的是内部生成的新启动脉冲。第二个 TMP05 传感器的输出以前一个传感器的 PWM 输出作为开始，紧接着的是第二个传感器自带的 PWM 输出，最后以内部生成的新启动脉冲作为结束。

当多个传感器采用菊花链方式连接时，最终返回至 PSoC 的信号包括所有传感器的 PWM 输出，以第一个传感器的输出作为开始，紧接着的是第二个传感器的输出，以此类推，直到发现新脉冲为止。随后，直到 TMP05 接口组件产生另一个新脉冲，温度至 PWM 的转换停止。系统及连接方案与预期波形如上所述。

采用 TMP05/06 数据手册所给等式算出的温度：

温度 = 421 – 751 * (TH/TL)

资源

TMP05 组件放置在整个 UDB 阵列中。该组件利用以下资源。

配置	资源类型					
	数据路径单元	宏单元	状态单元	控制单元	DMA 通道	中断
全部	2	12	1	1	–	1

API 存储器使用

根据编译器、组件、所用 API 数量和组件配置的不同，组件内存使用会出现较大变化。下表提供指定组件配置中可用的 API 的存储器使用。

已利用释放模式中配置的相关编译器进行了测量，大小采用了优化设定。对于特定设计，可以分析编译器生成的映射文件以确定存储器使用。

配置	PSoC 3 (Keil_PK51)		PSoC 4 (GCC)		PSoC 5LP (GCC)	
	闪存字节	SRAM 字节	闪存字节	SRAM 字节	闪存字节	SRAM 字节
连续	807	8	654	11	640	11
单触发	805	8	650	11	636	11

直流和交流电气特性

下面的值表示了预计性能，它们基于初始特性数据。

TMP05 接口交流规范

参数	说明	最小值	典型值	最大值	单位
f_{CLOCK}	组件时钟频率	100	–	500	kHz

性能

已使用发布模式中配置的关联编译器使用 24 MHz 的 CPU 速度收集以下测量值。这些数字应视为近似值，并应用于确定必要的权衡。

GetTemperature API

计算周期数 (8051)	计算周期数 (ARM cortex M0)	计算周期数 (ARM cortex M3)
1230	100	60

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
1.10	添加了 PSoC4 支持。	
	更新了数据手册的特性、概述、资源、内存使用情况以及性能部分。	
1.0	版本 1.0 是 TMP05Intf 组件的首次发行版。	

赛普拉斯半导体公司，2013-2016 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion徽标，及上述项目的组合，及PSoC、CapSense、EZ-USB、F-RAM和Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。