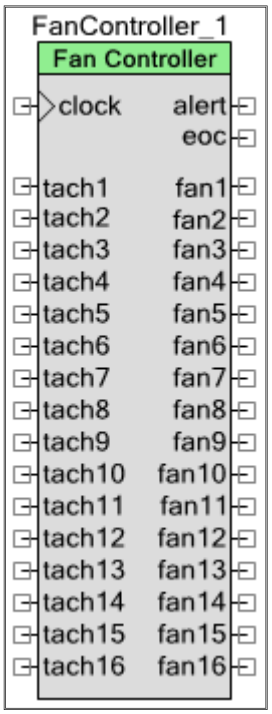
风扇控制器

2.10

特性



- 最多支持 16 个由脉冲宽度调制器控制的 4 线无刷直流风扇。
- 独立的或分组的脉冲宽度调制器输出，带转速计输入
- 支持 25 kHz、50 kHz 或用户定义的脉冲宽度调制器频率
- 支持风扇速度最高为 25,000 RPM
- 支持 4 极和 6 极电机
- 支持所有风扇上的风扇停止/电机锁定检测
- 支持由固件控制的或由硬件控制的风扇速度调节
- 用于风扇故障报告的可自定义警报引脚

概述

风扇控制器器件能够使设计人员快速、轻松地使用 PSoC 开发风扇控制器解决方案。该器件是一个系统级的解决方案，并且封装了所有必要的硬件模块，包括脉冲宽度调制器、转速计输入捕捉定时器、控制寄存器、状态寄存器和 DMA，减少开发时间和开发工作。

此器件可通过图形用户界面进行自定义，使设计人员能够输入风扇机电参数，例如占空比到 RPM 的映射和物理风扇分组组织。可通过同一用户界面配置包括脉冲宽度调制器频率和分辨率在内的性能参数以及开环或闭环控制方法。一旦输入了系统参数，此器件将在 PSoC 内提供最佳的实现，从而节省资源，以实现其他热管理和系统管理功能的整合。提供了易于使用的 API，使固件开发人员能够快速地启动和运行。

***注意：** 使用风扇控制器器件的设计不得使用 PSoC 的低功耗睡眠或休眠模式。进入这些模式会阻止风扇控制器器件控制和监控风扇。*

何时使用风扇控制器

风扇控制器器件应用于驱动和监控基于脉冲宽度调制器的 4 线直流散热风扇需要的任何热管理应用中。如果应用需要超过 16 个风扇，风扇控制器器件可以多次进行实例化（添加多个风扇控制器）。同样，如果应用中的风扇被分成各个组，设计人员可以选择每组实例化一个风扇控制器器件或实例化一个可处理所有分组的器件。

输入/输出连接

本节介绍模拟风扇控制器的各种输入和输出连接。I/O 列表中的星号 (*) 表示该 I/O 是可隐藏 I/O，其隐藏条件在该 I/O 的说明中。

clock – Input *（时钟 – 输入）*

风扇控制脉冲宽度调制器的用户定义时钟源的输入。它仅在器件自定义程序中选择了 External Clock（外部时钟）选项时才显示。

tach1..16 – 输入*

每个风扇的转速计信号，用于启用风扇控制器以测量风扇旋转速度。此器件设计用于与 4 极直流风扇一起使用，它们在其转速计输出上每次旋转生成 2 个高-低脉冲序列，以及与 6 极直流风扇一起使用，它们生成 3 个高-低脉冲序列。tach2..16 个输入是可选项。

fan1..16 – 输出 *

脉冲宽度调制器输出，包括用于控制风扇速度的变量占空比。如果已启用风扇分组，这些输出终端将被 bank1..8 输出替换。fan2..16 是可选项。

注意，对于硬件 UDB 模式，风扇输出（和关联的转速计输入）的最大数量仅限于 12，以最大程度减少数字资源利用率。

bank1..8 – 输出*

脉冲宽度调制器输出，包括用于控制风扇分组的速度的变量占空比。这些输出仅在启用了分组时才显示。

警报 – 输出*

当检测到了风扇故障（如果已启用）时，将设置高电平有效输出终端。

eoc – 输出*

每次转速计模块测量了系统中的所有风扇后，周期末输出将处于高脉冲。它可用于通过将终端连接至状态寄存器器件或中断器件的方式，使固件算法与风扇控制器硬件同步。

元件参数

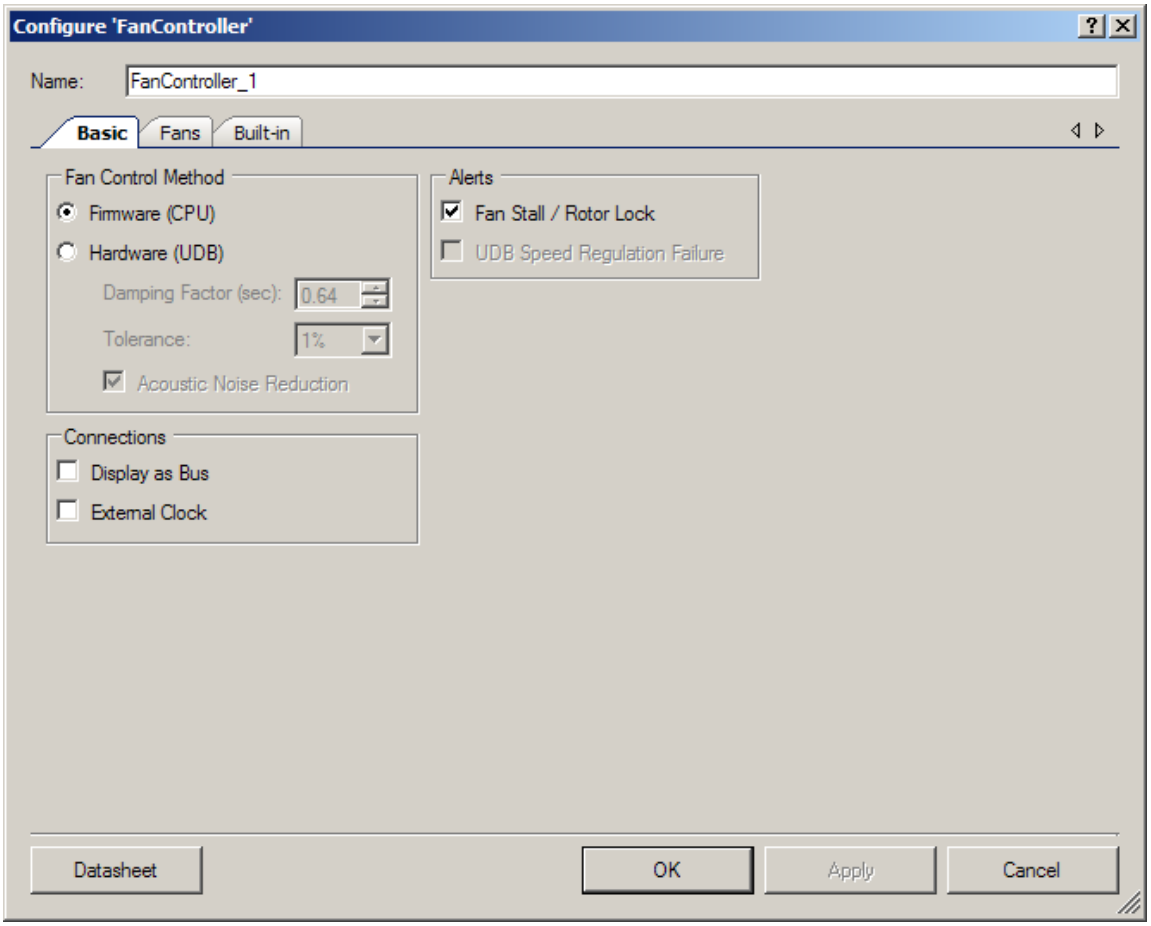
将风扇控制器拖入您的设计中，双击它打开 Configure（配置）对话框。此选项卡用于配置此器件的基本工作参数。

图 1 显示了 Configure（配置）对话框。

Basic Tab Options（基本选项卡选项）

此选项卡用于配置此器件的基本工作参数。

图 1. 基本配置对话框



Fan Control Method（风扇控制方法） – Firmware/Hardware（固件/硬件）

此参数确定如何控制风扇速度。当用户固件控制风扇速度调节时，使用固件 Firmware (CPU)（固件 (CPU)）设置。固件可设置脉冲宽度调制器占空比和读回风扇速度以确定合适的操作。

Hardware (UDB)（硬件 (UDB)）设置表示 PSoC 内部的硬件模块自动控制风扇速度调节，而不会存在任何 CPU 干预。固件为每个风扇设置需要的速度，硬件自动调节脉冲宽度调制器占空比以达到并维持指定容差范围内需要的速度。默认设置为 Firmware (CPU)（固件 (CPU)）。

在以下情况下，应选择 Firmware (CPU)（固件 (CPU)）设置：

1. 设计人员需要在 PSoC 内部的固件中实现复杂的或自定义的风扇控制算法。
2. 外部主机控制器负责管理风扇速度算法，风扇控制器器件只是用作为风扇的硬件接口。
3. 多个风扇被为各个组，共享常见的脉冲宽度调制器驱动信号。

在以下情况下，应选择 Hardware (UDB)（硬件 (UDB)）设置：设计人员需要单独控制多个风扇，且希望通过最少的固件开发执行此操作。

Hardware Control Mode（硬件控制模式）– Damping Factor（阻尼系数）

在由硬件控制的风扇模式中，此参数控制硬件控制回路的动态响应时间。此参数控制硬件间隔多久调节一次每个风扇的脉冲宽度调制器占空比。

在只有少数几个风扇的情况下，较大的阻尼系数确保控制回路可以调节至需要的速度，而不会在需要的速度目标周围波动。在有很多风扇受到控制的情况下，较小的阻尼系数确保有足够的时间来响应风扇速度的变化。此参数实现了硬件控制逻辑的微调，以符合选定风扇的机电特性。此参数的有效范围为 0..1.27（单位：秒）。默认设置为 0.64。

Hardware Control Mode（硬件控制模式）– Tolerance（容差）

在由硬件控制的风扇模式中，在指定需要的风扇速度目标时，此参数设置可接受容差。此容差被指定为与需要的速度设置相关的百分比。此参数实现了硬件控制逻辑的微调，以符合选定风扇的机电特性。

此参数的有效范围为 1..10%。默认设置为 1%。如果风扇控制器 Fans（风扇）选项卡中选择了 8-bit（8 位）脉冲宽度调制器分辨率，如图 2 所示，推荐使用 5% 的 Tolerance（容差）参数。

Hardware Control Mode（硬件控制模式）– Acoustic Noise Reduction（ 吸声降噪）

在由硬件控制的风扇模式中，此参数通过限制速度的正向变化率来限制风扇的可听噪音。如果已启用，且固件请求提高需要的风扇速度，应用于风扇的脉冲宽度调制器占比空将逐渐提高到新的设置，而不是应用突然性的改变。这消除了由于突然性的速度提高而导致的嘈杂的风扇呼呼声。已选中默认设置。

Alerts（警报）– FanStall / RotorLock

可配置风扇控制器在风扇由于机械性阻塞而延迟或停止时生成高电平有效警报信号。已选中默认设置。

Alerts（警报）– UDB SpeedRegulationFailure

可配置风扇控制器在由硬件控制的模式中当硬件控制回路无法达到需要的速度时生成高电平有效警报信号。未选中默认设置。

Connections（连接）– Display as Bus（显示为总线）

如果未选中，转速计输入和风扇/分组输出将显示为总线。否则，它们将显示为独立终端。

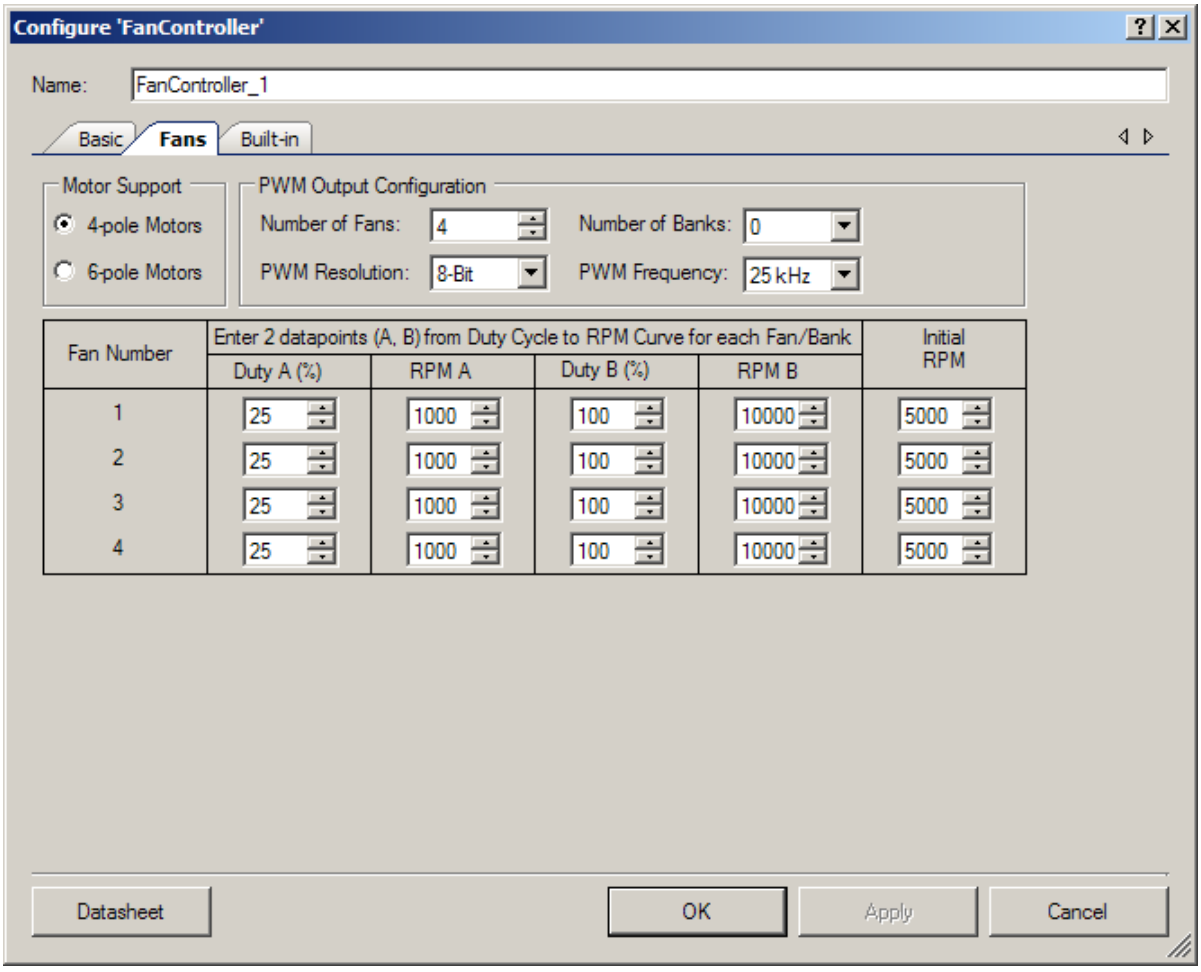
Connections（连接）– External Clock（外部时钟）

如果已选中，它允许用户将外部时钟源连接至器件时钟输入。如果未选中，则将使用内部时钟源。

Fans Tab Options（风扇选项卡选项）

此选项卡用于配置特定于风扇的参数。

图 2. 风扇配置对话框



Motor Support（电机支持）

此参数指定每转在风扇的转速计输出上显示的高-低脉冲的数量。4 极选项意味着每次风扇旋转一周存在 2 个高脉冲和 2 个低脉冲，而 6 极选项意味着每次风扇旋转一周存在 3 个高脉冲和 3 个低脉冲。

脉冲宽度调制器 Output Configuration（脉冲宽度调制器输出配置）– Number of Fans（风扇数量）

此参数显示系统中有多少个风扇。此参数的有效范围为 1..16。默认设置为 4。

脉冲宽度调制器 Output Configuration（脉冲宽度调制器输出配置 ）– Number of Banks（分组数量）

此参数仅在固件控制模式下可见，用于指定系统中有多少个风扇分组。假定当风扇被分成各个组时每个组都有相同数量的风扇。因此，风扇的数量必须除以分组的数量。值 0 表示风扇未进行分组。针对分组操作，此参数的有效值为 1 到（风扇数量/ 2）。默认设置为 0。

脉冲宽度调制器输出配置——脉冲宽度调制器分辨率

此参数指定用于驱动风扇以控制旋转速度的调制脉冲宽度调制器信号的占空比的分辨率。此参数的有效选项为 8-bit（8 位）或 10-bit（10 位）。默认设置为 8-bit（8 位）。

脉冲宽度调制器输出配置——脉冲宽度调制器频率

此参数指定用于驱动风扇的调制 脉冲宽度调制器信号的频率。当使用了内部时钟时，此参数的有效配置为 25 kHz 到 50 kHz。默认设置为 25 kHz。当使用了外部时钟时，此参数显示为灰色，因为脉冲宽度调制器频率取决于输入时钟源。有关详细信息，请参见“功能说明”一节。

Fan Specifications（风扇规范）– Duty A (%)（占空比 A (%)），RPM A

综合这些参数为选定的风扇或风扇分组在占空比到 RPM 的传输函数上指定一个数据点。RPM A 参数指定当风扇由脉冲宽度调制器使用占空比 Duty A (%)（占空比 A (%)） 驱动时风扇正常运行时的速度。此信息可从风扇制造商的基本介绍中获得。应注意风扇控制器器件可能会将脉冲宽度调制器占空比下压至 0%，即使 Duty A (%)（占空比 A (%)）设置为非零值。

Duty A (%)（占空比 A (%)）参数的有效范围为 0-99。默认设置为 25。

RPM A 参数的有效范围为 500..24,999。默认设置为 1,000。

Fan Specifications（风扇规范）– Duty B (%)（占空比 B (%)），RPM A

综合这些参数为选定的风扇或风扇分组在占空比到 RPM 的传输函数上指定另一个数据点。RPM B 参数指定当风扇由脉冲宽度调制器使用占空比 Duty B (%)（占空比 B (%)） 驱动时风扇正常运行时的速度。此信息可从风扇制造商的基本介绍中获得。应注意风扇控制器器件可能会将脉冲宽度调制器占空比上提至 100%，即使 Duty B (%)（占空比 A (%)）设置低于 100%。

Duty B (%)（占空比 A (%)）参数的有效范围为 1-100。默认设置为 100。

RPM B 参数的有效范围为 501..25,000。默认设置为 10,000。

Fan Specifications（风扇）– Initial RPM（初始 RPM）

此参数指定单个风扇的初始 RPM。Initial RPM（初始 RPM）的值将被转换为占空比，并被设置为单个风扇的初始占空比。Initial RPM（初始 RPM）参数可设置低于 RPM A 参数。

时钟选择

此器件使用时钟树资源进行其操作。这些时钟为：总线时钟、转速计时钟 (500 kHz) 和脉冲宽度调制器时钟（6、12 或 24 MHz，取决于配置）。此器件有一个连接外部时钟源而非脉冲宽度调制器时钟的选项。

应用程序编程接口

应用程序编程接口 (API) 子程序允许您使用固件与器件交互。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 FanController_1 分配给提供的设计中的第一个器件实例。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 FanController_1。

函数

函数	说明
FanController_Start()	启动此器件
FanController_Stop()	停止此器件并禁用硬件模块
FanController_Init()	初始化此器件
FanController_Enable()	启用此器件内部的硬件模块
FanController_EnableAlert()	启用此器件中的警报
FanController_DisableAlert()	禁用此器件中的警报
FanController_SetAlertMode()	配置警报资源
FanController_GetAlertMode()	返回当前启用的警报源
FanController_SetAlertMask()	启用每个风扇的警报的掩码
FanController_GetAlertMask()	返回每个风扇的警报掩码状态
FanController_GetAlertSource()	返回待处理警报源
FanController_GetFanStallStatus()	返回表示每个风扇的状态的位掩码

函数	说明
FanController_GetFanSpeedStatus()	返回表示每个风扇在硬件控制模式中的速度调节状态的位掩码
FanController_SetDutyCycle()	为指定风扇或风扇分组设置脉冲宽度调制器占空比
FanController_GetDutyCycle()	返回指定风扇或风扇分组的脉冲宽度调制器占空比
FanController_SetDesiredSpeed()	设置指定风扇在硬件控制模式中需要的风扇速度
FanController_GetDesiredSpeed()	返回指定风扇在硬件控制模式中需要的风扇速度
FanController_GetActualSpeed()	返回指定风扇的实际速度
FanController_OverrideHardwareControl()	启用用于覆盖硬件 (UDB) 风扇控制的固件

全局变量

变量	说明
FanController_initVar(static)	initVar 变量用于说明此器件的初始配置。此变量前面加有器件名称。此变量被初始化为 0，并在第一次调用 FanController_Start() 时设置为 1。这实现了器件初始化，而无需重新初始化 FanController_Start() 例程中的所有后续调用。 如果需要重新初始化此器件，应首先调用 FanController_Stop() 例程，然后再调用 FanController_Init() 和 FanController_Enable()。

voidFanController_Start(void)

说明： 启用器件。如果此器件未提前进行初始化，调用 Init() API。调用 Enable() API。

参数： None

返回值： None

副作用： None

voidFanController_Stop(void)

说明： 禁用此器件 所有脉冲宽度调制器输出都将被驱使为 100% 占空比，以确保当此器件不工作时持续散热。

参数： None

返回值： None

副作用： 警报引脚已解除。

voidFanController_Init(void)

说明： 初始化此器件。

参数： None

返回值： None

副作用： None

voidFanController_Enable(void)

说明： 启用此器件内部的硬件模块。

参数： None

返回值： None

副作用： None

voidFanController_EnableAlert(void)

说明： 启用警报信号的生成。特别是，使用 FanController_SetAlertMode() 和 FanController_SetAlertMask() API 配置启用哪些警报源。

参数： None

返回值： None

副作用： None

voidFanController_DisableAlert(void)

说明： 禁用警报信号的生成。

参数： None

返回值： None

副作用： 警报引脚已解除。

voidFanController_SetAlertMode(uint8 alertMode)

说明： 配置此器件的警报源。有两个警报源：1) 风扇停止或电机锁定； 2) 硬件控制模式速度调节失败。

参数： uint8 alertMode

位字段	启用警报源
FanController_STALL_ALERT	1=启用 fanstall / rotorlockalert
FanController_SPEED_ALERT	1=启用 ClosedLoopspeedregulationfailurealert

返回值： None

副作用： None

uint8FanController_GetAlertMode(void)

说明： 返回已启用的警报源。

参数： None

返回值： uint8 alertMode

位字段	启用警报源
FanController_STALL_ALERT	1=启用 fanstall / rotorlockalert
FanController_SPEED_ALERT	1=启用 ClosedLoopspeedregulationfailurealert

副作用： None

voidFanController_SetAlertMask(uint16 alertMask)

说明： 通过掩码启用或禁用来自于每个风扇的警报。掩码同时应用于风扇停止警报和速度调节失败警报。

参数： uint16 alertMask

位字段	启用警报源
bit0	1=启用 Fan1 的警报
bit1	1=启用 Fan2 的警报
…	…
bit15	1=启用 Fan16 的警报

返回值： None

副作用： None

uint16FanController_GetAlertMask(void)

说明： 返回每个风扇的警报掩码状态。掩码同时应用于风扇停止警报和速度调节失败警报。

参数： None

返回值： uint16 alertMask

位字段	启用警报源
bit0	1=启用 Fan1 的警报
bit1	1=启用 Fan2 的警报
…	…
bit15	1=启用 Fan16 的警报

副作用： None

uint8FanController_GetAlertSource(void)

说明： 返回此器件的待处理警报源。此 API 可用于轮询此器件的警报状态。或者，如果警报引脚用于生成 PSoC 的 CPU 内核中断，中断服务例程可使用此 API 确定警报源。在任何一种情况下，如果此 API 返回了非零值，FanController_GetFanStallStatus() 和 FanController_GetFanSpeedStatus() API 可提供有关哪个（些）风扇出现了故障的进一步信息。

参数： uint8 alertMode

位字段	待处理警报
FanController_STALL_ALERT	1=Fanstall / rotorlockalert 待处理
FanController_SPEED_ALERT	1=Closed Loopspeedregulationfailurealert 待处理

返回值： None

副作用： None

uint16FanController_GetFanStallStatus(void)

说明： 返回所有风扇的停止/电机锁定状态。

参数： None

返回值： uint16 stallStatus

位字段	状态
bit0	Fan1 停止状态（1=停止，0=正常）
bit1	Fan2 停止状态
…	…
bit15	Fan16 停止状态

副作用： 调用此 API 将清除所有待处理的风扇停止警报，且将解除警报引脚。如果停止警报是待处理警报，将重新设置警报引脚与下一个周期末 (eoc) 脉冲保持同步。

uint16FanController_GetFanSpeedStatus(void)

说明： 返回所有风扇的硬件风扇控制模式速度调节状态。在以下两种情况下，会出现速度调节失败：1)如果需要的风扇速度超过了当前实际风扇速度，但风扇的占空比已经为 100%；2)如果需要的风扇速度低于当前实际风扇速度，但风扇的占空比已经为 0%。

参数： None

返回值： uint16 speedStatus

位字段	状态
bit0	Fan1 速度调节状态（1=失败，0=正常）
bit1	Fan2 速度调节状态
…	…
bit15	Fan16 速度调节状态

副作用： 调用此 API 将清除所有处理的速度调节警报，且将解除警报引脚。如果速度调节警报是待处理警报，将重新设置警报引脚与下一个周期末(eoc)脉冲保持同步。

voidFanController_SetDutyCycle(uint8 fanOrBankNumber, uint16 dutyCycle)

说明： 设置指定风扇或风扇分组的脉冲宽度调制器占空比（单位：万分比）。在硬件风扇控制模式中，如果需要手动占空比控制，在调用此 API 之前首先调用 FanController_OverrideHardwareControl()。

参数： uint8 fanOrBankNumber
有效范围为 1..16，但不得超过系统中的风扇或分组的数量。

uint16 dutyCycle
占空比（单位：万分比）。例如，50% 的占空比 = 5000。有效范围为 0..10000。

返回值： None

副作用： None

uint16FanController_GetDutyCycle(uint8 fanOrBankNumber)

说明： 返回指定风扇或风扇分组的脉冲宽度调制器占空比（单位：万分比）。

参数： uint8 fanOrBankNumber
有效范围为 1..16，但不得超过系统中的风扇或分组的数量。

返回值： 占空比（单位：万分比）。例如，50% 的占空比 = 5000。

副作用： None

voidFanController_SetDesiredSpeed(uint8 fanNumber, uint16 rpm)

说明： 设置指定风扇需要的速度（单位：每分钟转数 (RPM)）。在硬件风扇控制模式中，RPM 参数被传送至控制回路硬件，作为调节的新目标风扇速度。在固件风扇控制模式中，RPM 参数将基于自定义程序的 Fans（风扇）选项卡中输入的风扇参数被转换为占空比，并被写入到适当的脉冲宽度调制器中。这向固件提供了启动粗粒度级速度控制的方法。则可使用 FanController_SetDutyCycle() API 达成细粒度固件速度控制。

参数： uint8 fanNumber

有效范围为 1..16，但不得超过系统中风扇的数量。

uint16 rpm

有效范围为 500..25,000，但不得超过风扇能够运行的最大 RPM。否则会导致速度调节失败。

返回值： None

副作用： None

uint16FanController_GetDesiredSpeed(uint8 fanNumber)

说明： 返回选定风扇当前需要的速度。

参数： uint8 fanNumber

有效范围为 1..16，但不得超过系统中风扇的数量。

返回值： RPM 中选定风扇当前需要的速度。

副作用： None

uint16 FanController_GetActualSpeed(uint8 fanNumber)

说明： 返回选定风扇当前的实际速度。

参数： uint8 fanNumber
有效范围为 1..16，但不得超过系统中风扇的数量。

返回值： RPM 中选定风扇当前的实际速度。

副作用： None

注意：在 PSoC3 器件上，在硬件模式中运行时，应在特定的时间窗口中调用 FanController_GetActualSpeed()。有关详细信息，请参见“功能说明”一节。

voidFanController_OverrideHardwareControl(uint8 override)

说明： 允许在硬件风扇模式中控制风扇。注意，在固件风扇控制模式下，无法调用此 API。

参数： uint8 override
0 = 硬件负责风扇的控制
1 = 固件负责风扇的控制
有效范围为 0..1。默认值为 0

返回值： None

副作用： None

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。要获取器件特定的示例，请打开器件目录中的对话框或原理图中的器件实例。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，请参考 PSoC Creator 帮助中的“查找示例项目”主题。

中断服务子程序

风扇控制器器件不自动提供任何中断服务例程 API。要为此器件使用中断，添加 PSoC Creator 中 Cypress（赛普拉斯）标准目录中的中断器件。

功能描述

自定义时钟

此器件具有允许连接外部时钟的功能。时钟源的频率确定脉冲宽度调制器输出频率，且内部时钟频率 (f_{CLK}) 与输出脉冲宽度调制器频率 ($f_{脉冲宽度调制器}$) 之间关系如以下公式所示：

$$f_{脉冲宽度调制器} = f_{CLK} / 240$$，针对 8 位分辨率；

$$f_{脉冲宽度调制器} = f_{CLK} / 960$$，针对 10 位分辨率。

（其中，常量 240 和 960 分别是针对 8 位分辨率模式和 10 位分辨率模式的最大周期数）。

实际速度

当此器件用于 PSoC 3 设计中时，需要在生成“eoc”脉冲的特定时间帧 (tACT) 内调用此 API，以确保可靠地捕捉 16 位速度值。此限制是由于 8051 CPU 内核一次性读取 16 位值和 8 位值。在此区域之外，可能存在以下情况：实际速度 MSB 与实际速度 LSB 读数可能包含风扇速度测量（之前的和当前）中的两个不同的数据。

为了保证不出现这种情况，在生成了“eoc”之后的这段时间内必须调用 GetActualSpeed() API：

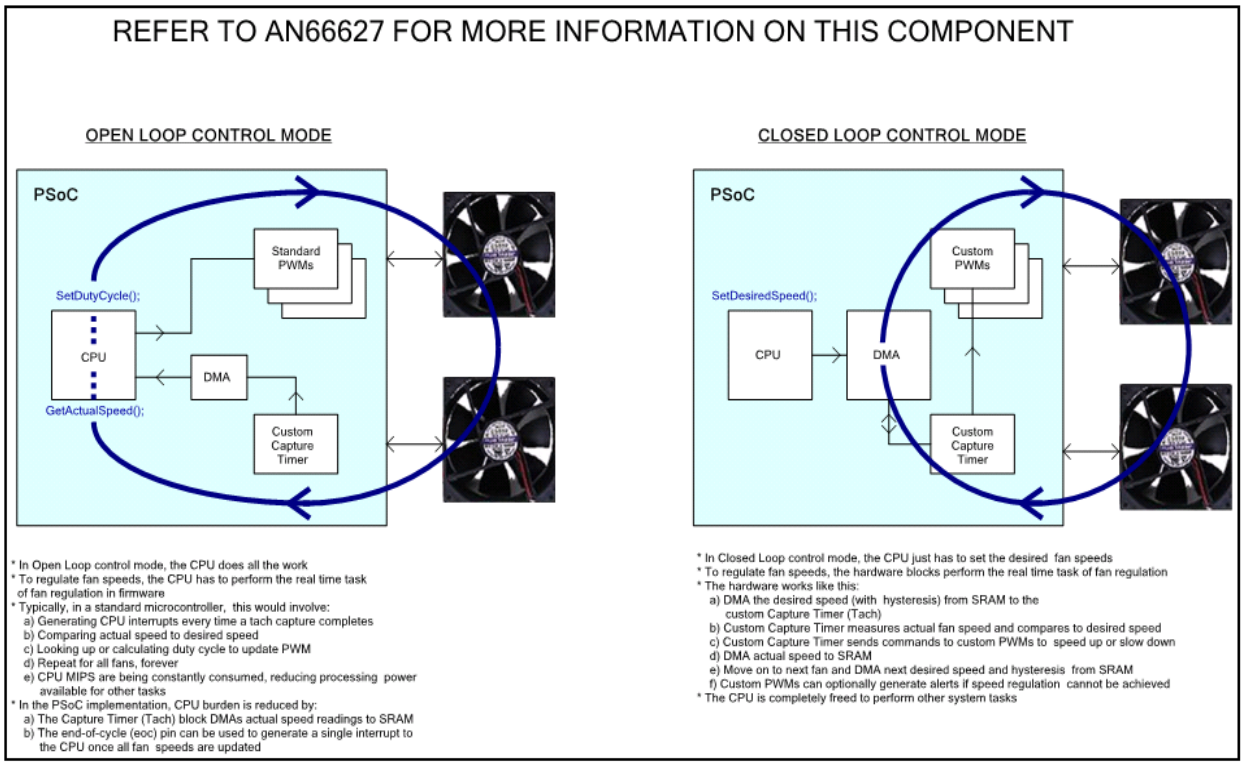
$$t_{ACT} = 2 * 60/RPMn_{MAX}$$

注意 $RPMn_{MAX}$ 是 fanN - 配置中使用的最后一个风扇的最大旋转速度。

框图和配置

图 3 中所示的原理图显示了此器件的两个基本工作模式的高级框图：1) 开环控制模式（固件调节风扇速度）；2) 闭环控制模式（硬件调节风扇速度）。

图 3. 风扇控制器模框图



寄存器

风扇控制器有多个控制寄存器和状态寄存器，它们被固件 API 用于控制操作和监控状态。用户固件不可直接访问上述任何寄存器。

资源

风扇控制器器件放置在整个 UDB 阵列中。此器件利用以下资源：

配置[1]	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
固件模式，8 位，4 个风扇	4	28	3	4	1	–
固件模式，8 位，8 个风扇	6	36	3	4	1	–
固件模式，8 位，12 个风扇	8	46	4	5	1	–
固件模式，8 位，16 个风扇	10	56	4	5	1	–
固件模式，10 位，4 个风扇	6	28	3	4	1	–
固件模式，10 位，8 个风扇	10	36	3	4	1	–
固件模式，10 位，12 个风扇	14	46	4	5	1	–
固件模式，10 位，16 个风扇	18	56	4	5	1	–
硬件模式，8 位，4 个风扇	6	57	4	8	2	–
硬件模式，8 位，8 个风扇	10	93	4	12	2	–
硬件模式，8 位，12 个风扇	14	132	6	17	2	–
硬件模式，10 位，4 个风扇	10	57	4	8	2	–
硬件模式，10 位，8 个风扇	18	93	4	12	2	–

API 内存使用情况

器件使用情况显著不同，取决于编译器、设备、使用 API 的数量以及器件配置。下表提供了给定的器件配置中的所有 API 的内存使用情况。

已使用 Release（发布）模式中配置的关联编译器进行了测量，此编译器使用了 Size（大小）的最佳设置。有关特定的设计，可分析编译器生成的映射文件以确定内存使用情况。

1. 对于硬件模式，不包括阻尼系数功能使用的资源。

配置[2]	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)		PSoC 5LP (GCC)	
	Flash（闪存）字节	SRAM 字节	Flash（闪存）字节	SRAM 字节	Flash（闪存）字节	SRAM 字节
固件模式，8 位，4 个风扇	1396	87	856	106	856	106
固件模式，8 位，8 个风扇	1396	171	840	206	840	206
固件模式，8 位，12 个风扇	1413	255	872	306	872	306
固件模式，8 位，16 个风扇	1413	339	876	406	880	406
固件模式，10 位，4 个风扇	1425	87	872	106	868	106
固件模式，10 位，8 个风扇	1425	171	852	206	852	206
固件模式，10 位，12 个风扇	1442	255	884	306	888	306
固件模式，10 位，16 个风扇	1442	339	888	406	892	406
硬件模式，8 位，4 个风扇	2222	129	1380	163	1380	163
硬件模式，8 位，8 个风扇	2222	253	1380	319	1380	319
硬件模式，8 位，12 个风扇	2243	377	1428	475	1424	475
硬件模式，10 位，4 个风扇	2239	129	1388	163	1388	163
硬件模式，10 位，8 个风扇	2239	253	1388	319	1388	319

2. 对于硬件模式，不包括阻尼系数功能使用的资源。

直流和交流电气特性

除非另有说明，否则这些规范的适用条件是 -40 °C ≤ T_A ≤ 85 °C 且 T_J ≤ 100 °C。除非另有说明，否则这些规范的适用范围为 1.71 V 到 5.5 V。

直流特性（脉冲宽度调制器时钟 – 6 MHz）

参数	说明	最小值	典型值[3]	最大值	单位
Idd	器件电流消耗（固件模式，4 个风扇）				
	8 位分辨率	–	108	–	µA
	10 位分辨率	–	139	–	µA
	器件电流消耗（固件模式，8 个风扇）				
	8 位分辨率	–	194	–	µA
	10 位分辨率	–	260	–	µA
	器件电流消耗（固件模式，12 个风扇）				
	8 位分辨率	–	307	–	µA
	10 位分辨率	–	395	–	µA
	器件电流消耗（固件模式，16 个风扇）				
	8 位分辨率	–	392	–	µA
	10 位分辨率	–	505	–	µA
	器件电流消耗（硬件模式，4 个风扇）				
	8 位分辨率	–	225	–	µA
	10 位分辨率	–	269	–	µA
	器件电流消耗（硬件模式，8 个风扇）				
	8 位分辨率	–	417	–	µA
	10 位分辨率	–	518	–	µA
	器件电流消耗（硬件模式，12 个风扇）				
	8 位分辨率	–	636	–	µA

3. 未包括设备 IO 和时钟分配的电流。路由条件、转速计输入的频率、温度等其他因素对电流消耗也有影响。这些值是在 25 °C 时的值。

直流特性（脉冲宽度调制器时钟 – 12 MHz）

参数	说明	最小值	典型值	最大值	单位
Idd	器件电流消耗（固件模式，4 个风扇）				
	8 位分辨率	–	169	–	µA
	10 位分辨率	–	233	–	µA
	器件电流消耗（固件模式，8 个风扇）				
	8 位分辨率	–	310	–	µA
	10 位分辨率	–	439	–	µA
	器件电流消耗（固件模式，12 个风扇）				
	8 位分辨率	–	498	–	µA
	10 位分辨率	–	677	–	µA
	器件电流消耗（固件模式，16 个风扇）				
	8 位分辨率	–	641	–	µA
	10 位分辨率	–	871	–	µA
	器件电流消耗（硬件模式，4 个风扇）				
	8 位分辨率	–	403	–	µA
	10 位分辨率	–	492	–	µA
	器件电流消耗（硬件模式，8 个风扇）				
	8 位分辨率	–	756	–	µA
	10 位分辨率	–	951	–	µA
	器件电流消耗（硬件模式，12 个风扇）				
	8 位分辨率	–	1162	–	µA

直流特性（脉冲宽度调制器时钟 – 24 MHz）

参数	说明	最小值	典型值	最大值	单位
Idd	器件电流消耗（固件模式，4 个风扇）				
	8 位分辨率	–	290	–	µA
	10 位分辨率	–	417	–	µA
	器件电流消耗（固件模式，8 个风扇）				
	8 位分辨率	–	545	–	µA
	10 位分辨率	–	798	–	µA
	器件电流消耗（固件模式，12 个风扇）				
	8 位分辨率	–	887	–	µA
	10 位分辨率	–	1243	–	µA
	器件电流消耗（固件模式，16 个风扇）				
	8 位分辨率	–	1150	–	µA
	10 位分辨率	–	1607	–	µA
	器件电流消耗（硬件模式，4 个风扇）				
	8 位分辨率	–	753	–	µA
	10 位分辨率	–	938	–	µA
	器件电流消耗（硬件模式，8 个风扇）				
	8 位分辨率	–	1446	–	µA
	10 位分辨率	–	1826	–	µA
	器件电流消耗（硬件模式，12 个风扇）				
	8 位分辨率	–	2220	–	µA

交流电规范

参数	说明	最小值	典型值	最大值	单位
f_{pwm_clk}	输入时钟频率[4]				
	固件模式，8 位，4 个风扇	–	–	54	MHz
	固件模式，8 位，8 个风扇	–	–	46	MHz
	固件模式，8 位，12 个风扇	–	–	39	MHz
	固件模式，8 位，16 个风扇	–	–	28	MHz
	固件模式，10 位，4 个风扇	–	–	42	MHz
	固件模式，10 位，8 个风扇	–	–	42	MHz
	固件模式，10 位，12 个风扇	–	–	41	MHz
	固件模式，10 位，16 个风扇	–	–	37	MHz
	硬件模式，8 位，4 个风扇	–	–	57	MHz
	硬件模式，8 位，8 个风扇	–	–	57	MHz
	硬件模式，8 位，12 个风扇	–	–	40	MHz
	硬件模式，10 位，4 个风扇	–	–	50	MHz
	硬件模式，10 位，8 个风扇	–	–	47	MHz
f_{tach_clk}	转速计时钟频率	–	500	–	kHz
转速计分辨率	转速计计数器分辨率	–	16	–	位
f_{tach}	转速计速度	500[5]	–	25000	RPM

4. 这些值提供了所有风扇的最大安全工作脉冲宽度调制器频率。可以在更高的时钟频率运行器件，在该频率将需要使用静态时序分析结果验证时序要求。

5.低于最小速度的速度意味着停止。

器件更改

本节介绍器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
2.10	更新了器件特性数据。	
	添加了 PSoC 5LP 支持	
2.0	添加了器件特性数据。	
	更改了阻尼系数行为。	在之前的版本中，阻尼系数用于指定各个风扇之间的速度测量的延迟，且未使用任何单位指定延迟值。在本版本中，阻尼系数用于指定所有风扇的速度测量的开始时间之间的延迟。此外，此延迟现在使用秒为单位进行指定。
	在配置中添加了新参数每个风扇的 Initial RPM（初始 RPM）。	此参数指定在器件启动时特定风扇的旋转的近似速度。在之前的版本中，所有风扇都使用最大数据进行初始化。
	添加了用于选择此器件支持的电机类型的新功能。	此参数指定了每次风扇旋转一周存在的高-低脉冲的数量。对于 4 极电机，存在 2 个高低脉冲；对于 6 极电机，存在 3 个高低脉冲。
	已使用添加外部时钟源以驱动内部脉冲宽度调制器的功能扩展了器件符号。可在自定义程序的 GUI 中访问此选项。	新增器件功能。允许连接外部时钟。基于配置中源时钟的值和脉冲宽度调制器的分辨率，可调节脉冲宽度调制器输出频率。
	添加了新功能，此功能可在自定义程序的 GUI 中进行选择，用于将器件输入和输出显示为总线。	限制一系列的 tach1-tachN、fan1-fanN 和 bank1-bankN 连接可以显示为总线。这实现了原理图上空间的节约，因为此选项减小了器件符号的大小。
	向 API 中添加了 Keil 功能重新进入支持。	添加此功能，以便使客户能够指定各个生成函数可重入。
	删除了过时函数名称 - FanController_OverrideClosedLoop()，它曾经是 FanController_OverrideHardwareControl() 的简单 #define。	由于它在之前的版本中被标记为已过时，因此本版本中删除了它。
1.20	1. 进行了更新，以兼容 PSoC Creator v2.0	

版本	更改说明	更改/影响原因
	2. 被重新分类为“概念”软件 3. 修改了 SetDesiredSpeed() API 以提高准确性	
1.10	1. 修正了 SetDesiredSpeed() API 2. 向转速计输入添加了短时脉冲滤波器 3. 风扇控制方法术语由开/闭环更改为固件/硬件控制方法 4. 更新了符号颜色和大小 5. 更新（降低）了资源利用率 6. 删除了对电源管理 API 的参考（不支持）	
1.0	首次发行版	

© 赛普拉斯半导体公司，2012。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器件。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器件。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。