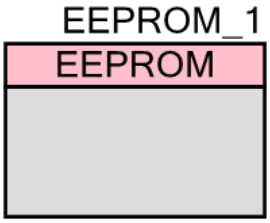
EEPROM
2.10



特性

- 512 B 到 2 KB EEPROM 存储器
- 1,000,000 次循环，数据保留时间为 20 年
- 一次读取/写入 1 字节
- 一次编程 16 字节（一行）

概述

EEPROM 组件提供了一个 API 函数库，以便擦除并向非易失性 EEPROM 存储器写入数据。术语“写入”指的是在一个操作中先擦除，然后编程。

在 PSoC 组件上，EEPROM 存储器以阵列作为组织单位。根据器件，PSoC 3 和 PSoC 5LP 器件可以提供大小分别为 512 字节、1 KB 或 2 KB 的 EEPROM 阵列。每个阵列被分为各行，每一行的大小为 16 个字节。EEPROM 组件的 API 函数库支持字节和行级别的写操作以及扇区级别的擦除操作。在 EEPROM 中，一个扇区包括 64 行。

EEPROM 组件没有初始化 EEPROM 存储器：存储器的初始状态在器件数据手册中定义。

EEPROM 组件可与 cy_boot 组件中包含的各种系统元件紧密连接。这些元件在成功编译后生成。有关 cy_boot 组件及其各种元件的更多信息，请参考*系统参考指南*部分的内容。

何时使用 EEPROM

EEPROM 组件可以用于以下各目的：

- 用于数据的附加存储（从而释放片上 RAM）
- 用于只读（或极少更改的）程序数据
- 用于电源通断变化时也不会丢失的数据（例如，校准表或组件配置）

输入/输出接口

没有用于 EEPROM 组件的 I/O 接口。它只是一个 API 函数。

组件参数

除了标准实例名称和内置参数外，EEPROM 没有其他的可配置参数。

应用编程接口（API）

通过应用编程接口（API），您可以使用软件对组件进行配置。下表列出并说明了每个函数的接口。以下各节将更加详细地介绍每个函数。

从 EEPROM 中读取时，不需要 API 函数。EEPROM 的全部内容映射到存储器空间，可以直接读取。EEPROM 允许字节级别的读访问。下面的参数定义用于读取 EEPROM：

- CYDEV_EE_BASE — EEPROM 存储器的基本指针
- CYDEV_EE_SIZE — EEPROM 存储器空间的大小（以字节为单位）
- CYDEV_EEPROM_ROW_SIZE — EEPROM 的行大小（以字节为单位）

通过使用基本指针，可以读取 EEPROM 存储器中的各单独字节。为了能够定位到特定行，基本指针必须以 EEPROM 行大小的次数而增加。

对于 PSoC 3 器件，当浮点整数被存储于 EEPROM 时，指向 EEPROM 变量的指针类型应转换为（CYXDATA *）：

```
(volatile float CYXDATA *)CYDEV_EE_BASE
```

可以使用 SIZEOF_EEPROM_ROW 的定义，而不是使用 CYDEV_EEPROM_ROW_SIZE 的定义。

EEPROM_1_EEPROM_SIZE 还被定义为 EEPROM 存储器空间的大小（以字节为单位，其中 EEPROM_1 是 EEPROM 组件的实例名称）。

进行一系列的 EEPROM 写操作前，必须通过调用 CySetTemp()获取 Die 的温度。CySetTemp() 函数将查询 SPC 以获取 Die 的温度并将其存储在全局变量中，后续执行 EEPROM 写入操作时，可以用到此变量。如果该应用程序使用环境中 Die 的温度变化了 10° C 或以上，则应刷新该温度以调整写入时间，从而达到最佳性能。

写入 EEPROM 最多需要 20 ms 的时间。在这个时间内不能复位器件，否则将导致 EEPROM 或闪存部分的意外更改。复位源包括 XRES 引脚、软件复位和看门狗。需要保证这些复位源不被无意激活。另外，需要配置低电压检测电路，使之生成中断而不是复位。

默认情况下，PSoC Creator 将实例名称“EEPROM_1”分配给指定设计中组件的第一个实例。您可以将其重新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“EEPROM”。

函数	说明
EEPROM_Enable()	使能EEPROM模块操作
EEPROM_Start()	启动EEPROM
EEPROM_Stop()	停止并关闭EEPROM的供电
EEPROM_EraseSector()	擦除EEPROM扇区
EEPROM_Write()	向EEPROM中写入一行时处于阻塞状态
EEPROM_StartWrite()	开始向EEPROM中写入一行数据
EEPROM_QueryWrite()	检查写入EEPROM的数据的状态
EEPROM_ByteWrite()	向EEPROM中写入一个字节的数据

void EEPROM_Enable(void)

说明： 使能EEPROM模块操作。

参数： 无

返回值： 无

其他影响： 对EEPROM_Start()的调用将调用EEPROM_Enable()。您可以直接调用EEPROM_Start()或EEPROM_Enable()函数；两者的作用相同。调用API后，EEPROM的启动时间需要5 µS，这时不要进行任何写操作。

void EEPROM_Start(void)

说明： 启动EEPROM。使用各写入/擦除API并读取EEPROM前，必须调用它们。

参数： 无

返回值： 无

其他影响： 对EEPROM_Start()的调用将调用EEPROM_Enable()。您可以直接调用EEPROM_Start()或EEPROM_Enable()函数；两者的作用相同。调用API后，EEPROM的启动时间需要5 µS，这时不要进行任何写操作。

void EEPROM_Stop(void)

说明： 停止并关闭EEPROM的供电。

参数： 无

返回值： 无

其他影响： 无

cystatus EEPROM_EraseSector(uint8 sectorNumber)

说明： 通过将扇区的各位设为零来擦除存储器的扇区（64行）。该函数一直处于阻塞状态，直到操作完成为止。

参数： uint8 sector。要擦除的扇区编号

返回值： 如果操作成功，则返回CYRET_SUCCESS
如果参数超过了范围，将返回CYRET_BAD_PARAM
如果EEPROM控制模块处于繁忙状态，则返回CYRET_LOCKED
如果存在EEPROM控制模块错误，则返回CYRET_UNKNOWN

其他影响： 无

cystatus EEPROM_Write(const uint8 * rowData, uint8 rowNumber)

说明： 将一行数据（16 字节）写入到EEPROM。这是一个阻塞调用。它不会在函数成功或失败之前返回任何值。

参数： const uint8 * rowData。写入EEPROM的数据的地址

uint8 rowNumber。要编程的EEPROM行号

返回值： 如果操作成功，则返回CYRET_SUCCESS
如果参数超过了范围，将返回CYRET_BAD_PARAM
如果EEPROM控制模块处于繁忙状态，则返回CYRET_LOCKED
如果存在EEPROM控制模块错误，则返回CYRET_UNKNOWN

其他影响： 无

cystatus EEPROM_StartWrite(const uint8 * rowData, uint8 rowNumber)

说明： 启动SPC写入函数。该函数不会处于阻塞状态；它在指令开始SPC写入函数后将立即返回值。此函数必须与EEPROM_QueryWrite()结合使用。一旦调用该函数，SPC将处于锁定状态，直到EEPROM_QueryWrite()返回CYRET_SUCCESS为止。

参数： const uint8 * rowData。写入EEPROM的数据的地址

uint8 rowNumber。要编程的EEPROM行号

返回值： 如果成功开始要写入的指令，则返回CYRET_STARTED

如果参数超过了范围，将返回CYRET_BAD_PARAM

如果EEPROM控制模块处于繁忙状态，则返回CYRET_LOCKED

如果存在EEPROM控制模块错误，则返回CYRET_UNKNOWN

其他影响： 无

cystatus EEPROM_QueryWrite(void)

说明： 检查写入EEPROM的数据的状态必须调用此函数，直到返回值不是CYRET_STARTED为止。

参数： 无

返回值： 如果操作成功，则返回CYRET_SUCCESS

如果仍在处理要写入的指令，则返回CYRET_STARTED

如果存在EEPROM控制模块错误，则返回CYRET_UNKNOWN

其他影响： 无

cystatus EEPROM_ByteWrite(unit8 dataByte, uint8 rowNumber, uint8 byteNumber)

说明： 向EEPROM中写入一个字节的数据。这是一个阻塞调用。它不会在函数成功或失败之前返回任何值。

参数： uint8 dataByte。要写入EEPROM的字节数据

uint8 rowNumber。要编程的EEPROM行号

uint8 byteNumber。需要在行内编程的字节数量

返回值： 如果操作成功，则返回CYRET_SUCCESS

如果各参数超过了范围，将返回CYRET_BAD_PARAM

如果EEPROM控制模块处于繁忙状态，则返回CYRET_LOCKED

如果存在EEPROM控制模块错误，则返回CYRET_UNKNOWN

其他影响： 无

MISRA 合规性

本节介绍了MISRA-C:2004合规性和本器件的偏差情况。有两种差异的类型，如下定义：

- 项目偏差 — 适用于所有 PSoC Creator 组件的偏差
- 特定偏差 — 仅适用于该组件的偏差

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”章节中介绍了项目偏差以及有关 MISRA 合规性验证环境的信息。

EEPROM 组件没有任何特定偏差。

示例固件源代码

在“Find Example Project”对话框中，PSoC Creator 提供了大量的示例项目，包括原理图和示例代码。要获取组件特定的示例，请打开器件目录中的对话框或原理图中的器件实例。要查看通用示例，请打开“Start Page”或 **File** 菜单中的对话框。根据要求，可以通过使用对话框中的 **Filter Options** 选项来限定可选的项目列表。

更多有关信息，请参考《PSoC Creator 帮助》部分中主题为“Find Example Project”的内容。

参考

另请参考 Die 温度组件数据手册和*系统参考指南*。

资源

EEPROM 组件使用了组件的 EEPROM 功能。

API 存储器大小

根据编译器、器件、所使用的 API 数量以及组件的配置不同，组件对存储资源的占用也不一样。下表提供了在某种器件配置中所有 API 占用存储器的大小。

数据是在将编译器设置为 Release 模式并将优化等级设置为 Size 的情况下测得的。对于特定的设计，分析完编译器生成的映射文件后可以确定存储器的使用情况。

配置	PSoC 3（Keil_PK51）		PSoC 5LP（GCC）	
	闪存 字节	SRAM 字节	闪存 字节	SRAM 字节
默认值	768	0	524	0

直流和交流电的电气特性

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。

直流规范

参数	说明	条件	最小值	典型值	最大值	单位
	擦除和编程电压		1.71	—	5.5	V

交流规范

参数	说明	条件	最小值	典型值	最大值	单位
T_{WRITE}	一行擦除/写入循环时间		—	2	20	ms
	EEPROM数据保留时间，即从最后一个擦除周期开始测量的保留时间的长度	平均环境温度，T_A ≤ 25 °C，1M擦除/编程循环	20	—	—	年
		平均环境温度，T_A ≤ 55 °C，100K擦除/编程循环	20	—	—	
		平均环境温度，T_A ≤ 85°C，10K擦除/编程循环	10	—	—	

组件更改

本节列出了该组件各版本中的主要更改内容。

版本	更新内容	更改原因/影响
2.10.b	更新了数据手册。	移除了已停产的PSoC 5器件的参考内容。
2.10.a	更新了EEPROM_Start/EEPROM_Enable API的介绍内容，添加访问PSoC 3器件上的浮点整数的说明内容。	用以说明组件的操作。
2.10	添加了MISRA合规性章节。	该组件没有任何特定偏差。
2.0.a	更新了交流和直流电气特性部分。	保持交流和直流特性与组件的数据手册一致。
	新增了有关EEPROM存储器的初始状态的信息。	详细说明组件的操作，并参考组件的数据手册，了解EEPROM存储器的初始状态的信息。
2.0	增加对PSoC 5LP芯片的支持。 增加了一个新的API EEPROM_ByteWrite()。	以便支持字节写入功能。
	更改了EEPROM_Write()、EEPROM_StartWrite()、EEPROM_QueryWrite() 和 EEPROM_EraseSector()的代码。	以便支持SPC代码的更改。
	移除从EEPROM_Write()和EEPROM_StartWrite()等API进行的CySetTemp()调用。	以便获得更好的性能。
1.60	对API EEPROM_Write()、EEPROM_StartWrite() 和EEPROM_QueryWrite()的代码稍作更改。	以便获得更好的性能。
	更改EEPROM_EraseSector() API中的代码以支持PSoC 5。	PSoC 5芯片支持擦除扇区指令。
	更改了数据手册中的EEPROM_EraseSector() API说明。	
1.50.b	在数据手册中添加了如何获取温度的说明。	清晰化
1.50.a	向数据手册中添加了特性数据。	
	请注意，数据手册中的EEPROM_EraseSector() API仅可适用于PSoC 3 Production或更高版本。	
	对数据手册进行了少量编辑和更新。	
1.50	修改了*EEPROM.c*文件以将include文件从*cydevice.h*文件切换为*cydevice_trm.h*。	*cydevice.h*文件是过时的文件。因此，生成的源代码以及由PSoC Creator提供的API应当使用*cydevice_trm.h*。

版本	更新内容	更改原因/影响
	更新了示例代码的EEPROM_EraseSector()部分。	在PSOC 3 ES1和ES2芯片或PSOC 5芯片的EEPROM上，“擦除扇区”指令不起作用。此API可在PSOC 3 Production或更高版本的EEPROM上使用。
	添加了EEPROM_Enable()、EEPROM_Start()和EEPROM_Stop() API。	旨在支持默认情况下关闭EEPROM电源的PSoC 3 Production芯片要求。 对EEPROM_Start()的调用将调用EEPROM_Enable()。您可以直接调用EEPROM_Start()或EEPROM_Enable()函数；两者的作用相同。
1.20.a	将组件移动到组件目录的子文件夹中。	
	向组件中添加了信息，以说明它与芯片版本的兼容性。	如果组件在不兼容的芯片上使用，该工具将报告错误/警告。如果发生该情况，请更新到支持您的目标组件的版本。
1.20	更新了“Configure”对话框。	在原理图中，数字端口已更改为引脚组件。

©赛普拉斯半导体公司，2010-2015。此处，所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品内嵌的电路外，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，否则赛普拉斯不保证产品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，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。该处引用的所有其它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。除上述指定的用途外，未经赛普拉斯明确的书面许可，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。对于合理预计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于赛普拉斯软件许可协议的限制。