



STM32F10xx8和STM32F10xxB 勘误手册

STM32F101x8/B、STM32F102x8/B和STM32F103x8/B 中容量产品的使用限制

芯片的识别

该勘误表适用于意法半导体的STM32F101xx基本型和STM32F103xx增强型中容量系列中的B、Z和Y版本芯片，和STM32F102xx USB基本型的Y版本芯片。

该芯片系列集成了ARM™ 32位Cortex®-M3内核，本文中也包含内核的勘误信息(详见第一章)。

表2列出了所有的产品型号。表1列出了本文涉及产品的识别：

- 通过芯片封装上产品标识下的版本号
- 通过包装盒标签上的产品标识的最后3个数字

表1 芯片的识别⁽¹⁾

销售型号	标注在芯片上的版本代码 ⁽²⁾
STM32F101xxx ⁽³⁾	“B”、“Z”或“Y”
STM32F102xxx ⁽³⁾	“Y”
STM32F103xxx ⁽³⁾	“B”、“Z”或“Y”

1. DBGMCU_IDCODE寄存器中REV_ID位指明芯片的版本号(关于版本号的详情，参见[STM32F10xxx参考手册](#))

2. 关于在不同芯片封装上识别版本号，请参考[附录A：产品上版本号标示](#)。

3. 所有32K字节闪存存储器，不带为缀A的型号也包含在其中。

表2 芯片概览

涉及到的芯片	芯片型号
STM32F101xx	STM32F101C8, STM32F101R8, STM32F101V8, STM32F101T8
	STM32F101RB, STM32F101VB, STM32F101CB
STM32F102xx	STM32F102C8, STM32F102R8
	STM32F102CB, STM32F102RB
STM32F103xx	STM32F103C8, STM32F103R8, STM32F103V8, STM32F103T8
	STM32F103RB, STM32F103VB, STM32F103CB

译著：本文英文原文下载地址：<http://www.st.com/stonline/products/literature/es/14574.pdf>

1 ARM™ 32-bit Cortex®-M3 的使用限制

STM32F10xxx内核的勘误可以从如下网址获得：

<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.eat0420a/>

还有一个勘误PDF文档的直接链接地址：

<http://infocenter.arm.com/help/topic/com.arm.doc.eat0420a/Cortex-M3-Errata-r1p1-v0.2.pdf>

描述的所有限制对应用的影响都很小，并且是针对Cortex-M3内核中的r1p1-01rel0版本。下表列出了这些局限性和它们对STM32F10xxx中容量产品所产生的影响。

表3 Cortex-M3核心局限性和对微控制器的影响

ARM 勘误ID	ARM 勘误分类	ARM勘误说明	对STM32F10xxx 中容量产品的影响
602117	2类	如果LDRD的基址在队列中，当被中断或失效时会导致不正确的基址寄存器	小
563915	2类	事件寄存器没有被中断和调试设置	小
531064	实现	SWJ_DP缺少POR复位同步	没有
511864	3类	从异常返回时，Cortex-M3可能使用不正确的优先级取指令	没有
532314	3类	DWT CPI计数器在睡眠时递增	没有
538714	3类	Cortex-M3 TPIU时钟区域交叉	没有
548721	3类	内部写缓冲区可能在睡眠时仍活跃	没有
463763	3类	BKPT在调试监控模式下能产生DFSR不匹配	小
463764	3类	核心可能停止在SLEEPONEEXIT单指令ISR	小
463769	3类	在写操作时不对齐的MPU失效可能导致写入错误的数据到一个成功的第一次操作	没有

1.1 Cortex-M3的局限性在STM32F10xxx中容量产品的影响

只有下面涉及的局限性才对STM32F10xxx中容量产品的实现产生较小的影响。

在ARM的勘误文档中所有其它的局限性(在表3列出)，对STM32F10xxx中容量产品(Cortex-M3 r1p1-01rel0)没有影响，也与STM32F10xxx中容量产品的实现无关。

1.1.1 Cortex-M3在中断或访问失效时，LDRD列表中的基址操作可能导致不正确的基址寄存器内容

问题描述

从系统总线区执行LDRD指令时，当在一个列表中的基址寄存器出现下述形式的指令时，Cortex-M3核心有一个缺陷：LDRD Ra, Rb, [Ra, #imm]。在加载了第一个目标寄存器后，可能因为加载第二个寄存器完成之前的中断，或因为加载第二个寄存器而产生总线失败时，不能完成指令的执行。

暂时解决办法

1. 当STM32F10xxx以标准的形式——从内置闪存存储器运行代码时，这个局限性不影响STM32F10xxx的运行。
2. 使用最新的编译器版本，从本文档发布之日起，编译器将不再产生这样的指令。同时，可以向你的编译器供应商索取“扫描工具”，它可以用于在早期版本编译器产生的代码中，检测这样的指令。

1.1.2 Cortex-M3事件寄存器没有被中断和调试设置

问题描述

当中断与一个WFE相关，而在WFE执行之前产生了这个中断时，不能设置用于WFE唤醒事件的事件寄存器，该事件会被丢失。因此，当执行WFE时，如果没有其它事件或中断发生时，核心不能被唤醒。

暂时解决办法

配置一个外部的或内部的EXTI线为事件模式，使用STM32F10xxx的外部事件，代替中断去把核心从WFE唤醒。

1.1.3 Cortex-M3的BKPT在调试监控模式下可以导致DFSR不匹配

问题描述

在调试监控模式下有可能执行BKPT指令，这会产生调用调试监控程序，但是在地址0xE000ED30的调试失效状态寄存器(DFSR)的位1并没有被设置，该位指示调用调试监控程序的源头是BKPT指令。这个现象仅仅出现在即将执行BKPT之前，正要处理中断而不是调试监控的时候。

暂时解决办法

如果在进入调试监控程序时，DFSR寄存器中没有任何标志位，则表示出现了这种极端情况，即在调试监控模式下执行了BKPT指令。

1.1.4 Cortex-M3可能在使用SLEEPONEXIT功能时，在单指令的ISR中停止

问题描述

如果使用了Cortex-M3的SLEEPONEXIT功能，并且涉及到的中断服务程序(ISR)仅包含一条指令，则核心会停止。如果只用一个中断有效，而且这个中断被一个仅包含一条指令的中断处理程序所抢断时，核心将会停止。

但是，如果任何新的中断能够造成抢断当前的中断处理程序，则核心恢复正常。

暂时解决办法

这个现象不会发生在真实地应用系统中，因为任何中断服务程序至少会包含一条以上的指令。如果确实只有一条指令，可以在退出指令(BX或BLX)之前插入一个NOP或任何其它指令。

2 STM32F10xxx 芯片的使用限制

下表是所有已经发现的局限性概览：

表4 芯片局限性列表

2.1节：在ADC输入0上的电压毛刺
2.2节：WFI/WFE指令之后读Flash存储器
2.3节：软件不能读调试寄存器
2.4节：复用功能
2.4.1节：USART1_RTS和CAN_TX
2.4.2节：SPI1处于从模式并且USART2处于同步模式
2.4.3节：SPI1处于主模式并且USART2处于同步模式
2.4.4节：SPI2处于从模式并且USART3处于同步模式
2.4.5节：SPI2处于主模式并且USART3处于同步模式
2.4.6节：I ² C2与SPI2和
2.4.7节：SPI1经过重映射并用于主模式时的I ² C1
2.4.8节：I ² C1和经过重映射的
2.5节：PWD和USB唤醒事件
2.6节：最新的编译器兼容问题
2.7节：边缘扫描TAP：“capture IR”状态之后送出错误的数据
2.8节：相对于设置STRT位，Flash存储器的BSY位延后
2.9节：I ² C外设
2.9.1节：某些软件事件必须在发送当前字节之前处理
2.9.2节：不完全支持SMBus标准
2.10节：通用定时器
2.10.1节：捕获标志丢失
2.10.2节：过早检测到捕获溢出
2.10.3节：通用定时器：占空比为100%的PWM
2.11节：LSI时钟稳定时间

2.1 在ADC输入0上的电压毛刺

问题描述

当以注入触发进行ADC转换时，在PA0引脚(ADC输入0)上，可能会产生一个低振幅的电压毛刺。这个毛刺是由内部耦合和同步到注入序列的开始和结束产生的，与要转换的通道无关。

毛刺的振幅低于150mV，典型的持续时间是10ns(在I/O引脚配置为高阻输入并悬空时测量的)。如果PA0是作为数字输出，这将不会影响到输出信号。如果PA0是配置为数字输入，只要驱动PA0的另一端的阻抗低于5kΩ，这个毛刺将不会被检测为意外的转换。在单ADC配置时，这个毛刺不会对端口A的其它引脚产生影响，也不会对ADC注入转换的结果产生影响。

当使用ADC的双模式，并使用注入触发时，为了避免任何可能的副作用，建议分配模拟通道时，把通道0配置为注入通道。

暂时解决办法

无。

2.2 WFI/WFE指令之后读Flash存储器

配置条件

- 打开Flash的预取指
- Flash的时序设置为2个等待周期
- 在停止模式下，FLITF(Flit接口)时钟停止

问题描述

如果在访问Flash存储器时，执行了WFI/WFE指令，而睡眠的时间又非常的短(少于2个时钟周期)，则在下一个唤醒事件时，从Flash存储器的取指操作可能被干扰。

暂时解决办法

当使用2个等待周期的Flash存储器配置并打开预取指功能时，在睡眠模式下不要停止FLITF(Flash接口)的时钟，即保持RCC_AHBENR寄存器的FLITFEN位为“1”(默认的复位状态)。

2.3 软件不能读调试寄存器

问题描述

DBGMCU_IDCODE和DBGMCU_CR调试寄存器只能在调试模式下被访问，不能通过用户软件操作。当试图在用户模式读取这些寄存器时，返回值是0x00。

暂时解决办法

无

2.4 复用功能

在某些特殊的情况下，映射到同一个引脚上的多个复用功能可能会有一些使用限制。

2.4.1 USART1_RTS和CAN_TX

配置条件

- USART1时钟使能
- CAN时钟关闭
- I/O端口的引脚PA12被配置为复用功能的输出

问题描述

即使没有用到CAN_TX，但是如果PA12被配置成了复用功能的输出，PA12会被默认设置为“1”。这样就不能使用USART1_RTS了。

暂时解决办法

如果要使用USART1的RTS功能，不管是否要用到CAN功能，都必须将CAN进行重映射：如果要使用CAN的话，就把CAN重映射到另外的I/O(把CAN_REMAP[1:0]设置成“10”或者“11”)；如果不使用CAN的话，就把CAN的重映射配置成不使用(即CAN_REMAP[1:0]设置成“01”)。

2.4.2 SPI1处于从模式并且USART2处于同步模式

配置条件

- SPI1和USART2的时钟都使能
- PA4被配置为复用输出

问题描述

如果SPI1处于从模式，则USART2(由于USART2_CK)不能工作于同步模式。

暂时解决办法

无

2.4.3 SPI1处于主模式并且USART2处于同步模式

配置条件

- SPI1和USART2的时钟都使能
- PA4被配置为复用输出

问题描述

如果SPI1处于主模式，并且SPI1_NSS配置为软件控制，则USART2不能工作于同步模式。因为以上的情况造成USART2_CK引脚上不能输出。

暂时解决办法

为了输出USART2_CK，必须设置SPI1_CR2寄存器中的SSOE位为'1'以使引脚处于输出模式。

2.4.4 SPI2处于从模式并且USART3处于同步模式

配置条件

- SPI2和USART3的时钟都使能
- PB12被配置为复用输出

问题描述

如果SPI2处于从模式，则USART3(由于USART3_CK)不能工作于同步模式。

暂时解决办法

无

2.4.5 SPI2处于主模式并且USART3处于同步模式

配置条件

- SPI2和USART3的时钟都使能
- PB12被配置为复用输出

问题描述

如果SPI2处于主模式，并且SPI2_NSS配置为软件控制模式，则USART3不能工作于同步模式。因为以上情况造成USART3_CK引脚上不能输出。

暂时解决办法

为了输出USART3_CK，必须设置SPI2_CR2寄存器中的SSOE位为'1'以使引脚处于输出模式。

2.4.6 I²C2与SPI2和USART3

配置条件

- 同时使能了I²C2和SPI2的时钟，或同时使能了I²C2和USART3的时钟。
- I/O端口PB12配置为复用输出。

问题描述

- 在I²C2的SMBALERT信号(即使不用这个功能)和SPI2_NSS在输出模式冲突。
- 在I²C2的SMBALERT信号(即使不用这个功能)和USART3_CK冲突。
- 在这些情况下，如果选择了I/O端口为复用输出并且使能了I²C2的时钟，则I/O端口PB12被默认置为'1'。

暂时解决办法

如果SPI2配置为主模式并且NSS是软件模式，则I²C2的SMBALERT信号可以作为输出。

如果SPI2配置为主模式或从模式并且NSS由软件管理，则I²C2的SMBALERT信号可以用于输入模式。

当使用I²C2时，SPI2不能用于其它配置。

当使用I²C2时，一定不能使用USART3的同步模式。

2.4.7 SPI1经过重映射并用于主模式时的I²C1

配置条件

- 同时使能了I²C1和SPI1的时钟。
- SPI1的引脚被重映射。
- I/O端口PB5配置为复用输出。

问题描述

在SPI1的MOSI信号和I²C1的SMBALERT信号(即使不用这个功能)之间有冲突。

暂时解决办法

不要在重映射SPI1后同时使用SPI1主模式和I²C1的功能。

当使用SPI1重映射功能时，不要使能I²C1的时钟。

2.4.8 I²C1和经过重映射的TIM3_CH2

配置条件

- 同时使能了I²C1和TIM3的时钟。
- I/O端口PB5配置为复用输出。

问题描述

在TIM3_CH2信号和I²C1的SMBALERT信号(即使不用这个功能)之间有冲突。

在这些情况下，如果选择了I/O端口为复用输出并且使能了I²C1的时钟，则I/O端口PB5被默认置为'1'，TIM3_CH2不能用于输出模式。

暂时解决办法

为避免这个冲突，TIM3_CH2只能用在输入模式。

2.5 PVD和USB唤醒事件

问题描述

PVD和USB唤醒，在内部是分别连到了EXTI线16和EXTI线18，它们不能作为Cortex-M3核心的事件源，因此这些信号不能作为退出(WFE)睡眠或停止模式的事件源。

暂时解决办法

如果应用程序必须要使用PVD或USB唤醒，则使用中断源和WFI指令。

2.6 最新的编译器兼容问题

问题描述

市场上已经出现了针对STM32F10xxx改进了优化的编译器。版本Z和B的中容量STM32F10xxx产品(STM32F10xx8/B)不能支持某些经这些编译器优化产生的代码序列，版本Y的产品没有这个限制。

暂时解决办法

这种现象是完全可以预测的，并且能够在代码的开发和测试阶段检测出来。因此已经开发、验证完成并交付的系统，在使用早期版本的产品时不会产生影响。

对于已经交付的、使用版本Z和B的系统升级时，不要使用这些新的编译器。目前已知产生这种指令序列的编译器有：

- IAR EWARM版本5.20和以后的版本
- GNU版本4.2.3和以后的版本

对于使用这些编译器进行新的开发时，必须使用版本Y的STM32F10xx8/B产品。

2.7 边缘扫描TAP: "capture IR"状态之后送出错误的数据

问题描述

当边缘扫描TAP进入"capture IR"状态之后，指令寄存器中2个最低位应该被加载为"01"，以便在移入下一个指令时移出去。

但是，边缘扫描TAP会移出最后进入指令寄存器的数值，可以是"00"、"01"、"10"或"11"。

暂时解决办法

在边缘扫描的流程中，应该丢弃"capture IR"状态之后移出的数据，软件不能仅仅检查最低2位(XXXX01)，而应该检查所有的寄存器位(XXXXXX)。

2.8 相对于设置STRT位，Flash存储器的BSY位延后

问题描述

当设置了Flash存储器控制寄存器的STRT位(启动擦除操作)，Flash存储器状态寄存器的BSY位晚一个周期变高。

如果在写入了FLASH_CR寄存器(设置STRT位)之后，立即读取FLASH_SR寄存器，则BSY位为'0'。

暂时解决办法

在设置了STRT位，至少一个周期之后再读BSY位。

2.9 I²C外设

2.9.1 某些软件事件必须在发送当前字节之前处理

问题描述

如果没有在传输当前字节之前处理EV7、EV7_1、EV6_1、EV2、EV8和EV3事件，有可能产生问题，如收到一个额外的字节、两次读到相同的数据或丢失数据。

暂时解决办法

当不能在传输当前字节之前和当改变ACK控制位送出相应脉冲之前，处理EV7、EV7_1、EV6_1、EV2、EV8和EV3事件时，建议如下操作：

1. 使用I²C的DMA模式，除非作为主设备时只接收一个字节。
2. 使用I²C的中断并把它的优先级设为最高级别，使得它不能被中断。

2.9.2 不完全支持SMBus标准

问题描述

I²C外设不完全兼容SMBus v2.0标准，它不能对非法的字节或命令发送一个NACK。

暂时解决办法

需要实现高层次的处理机制，验证在目标设备上执行的写操作是正确的，例如：

1. 如果主机支持，使用SMBAL引脚。
2. 使用警报响应地址(ARA)协议。
3. 使用主机提醒协议。

2.10 通用定时器

2.10.1 捕获标志丢失

问题描述

在捕获模式下，当发生捕获时正好在读CCRx寄存器，捕获标志(CCxIF)可能被清除而没有设置捕获溢出标志(CCxOF)。实际上，新的数据已经捕获至捕获寄存器中。

暂时解决办法

在处理捕获中断时，可以在读出捕获寄存器之前，使能捕获引脚上的外部中断，然后在读出捕获数据后关闭这个外部中断。这样有可能通过EXTI设备检测到错过的捕获事件。

2.10.2 过早检测到捕获溢出

问题描述

在捕获模式下，可能在没有丢失数据的情况下设置了捕获溢出标志(CCxOF)。

条件

当发生捕获时正好在读捕获寄存器，即使前面捕获的数据已被读出并且新的数据已经存入捕获寄存器，也会设置捕获溢出标志。

暂时解决办法

无

2.10.3 通用定时器：占空比为100%的PWM

问题描述

当使用OCREF_CLR功能时，当在OCREF_CLR信号上施加一个高电平，OCxREF信号变为无效(导致OCx无效并且OCxN有效)，PWM会在下一次计数器溢出时重新开始(输出再使能)。

但是如果PWM被配置为100%(CCxR > ARR)，则不能重新开始，同时OCxREF保持无效。

暂时解决办法

无

2.11 LSI时钟稳定时间

问题描述

当设置了LSIRDY标志，时钟仍有可能不在给定的频率范围内(见产品数据手册中的LSI振荡器特性， f_{LSI} 参数)。

暂时解决办法

为得到在给定范围内完全稳定的时钟，使用软件加入一个100μs的稳定时间。

附录A 产品上版本号标示

图1到图5分别显示了LFBGA100、LQFP100、LQFP64、LQFP48和VFQFPN36这五种不同封装的芯片上版本标志的记号。

图1 LFBGA100 封装的正面俯视图

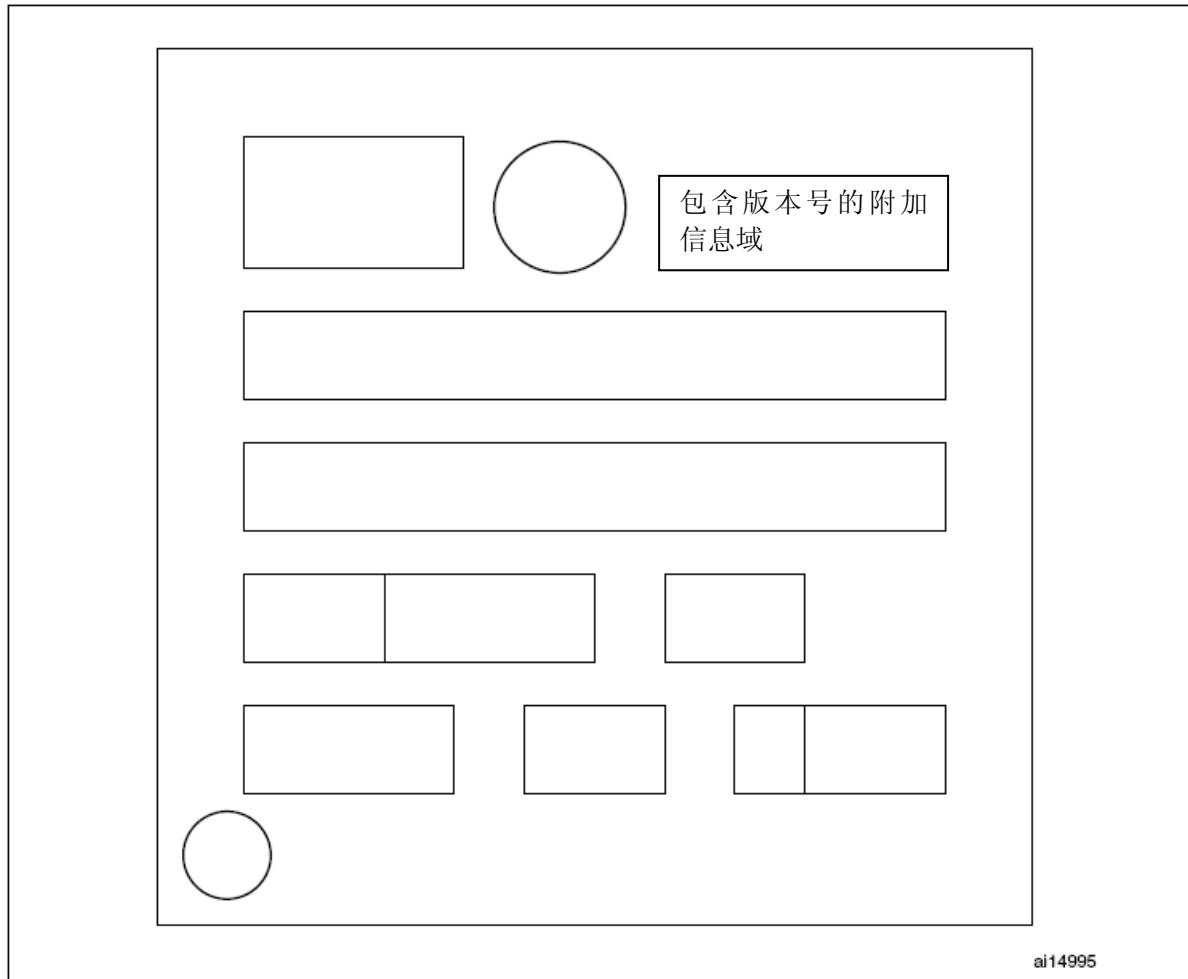


图2 LQFP100 封装的正面俯视图

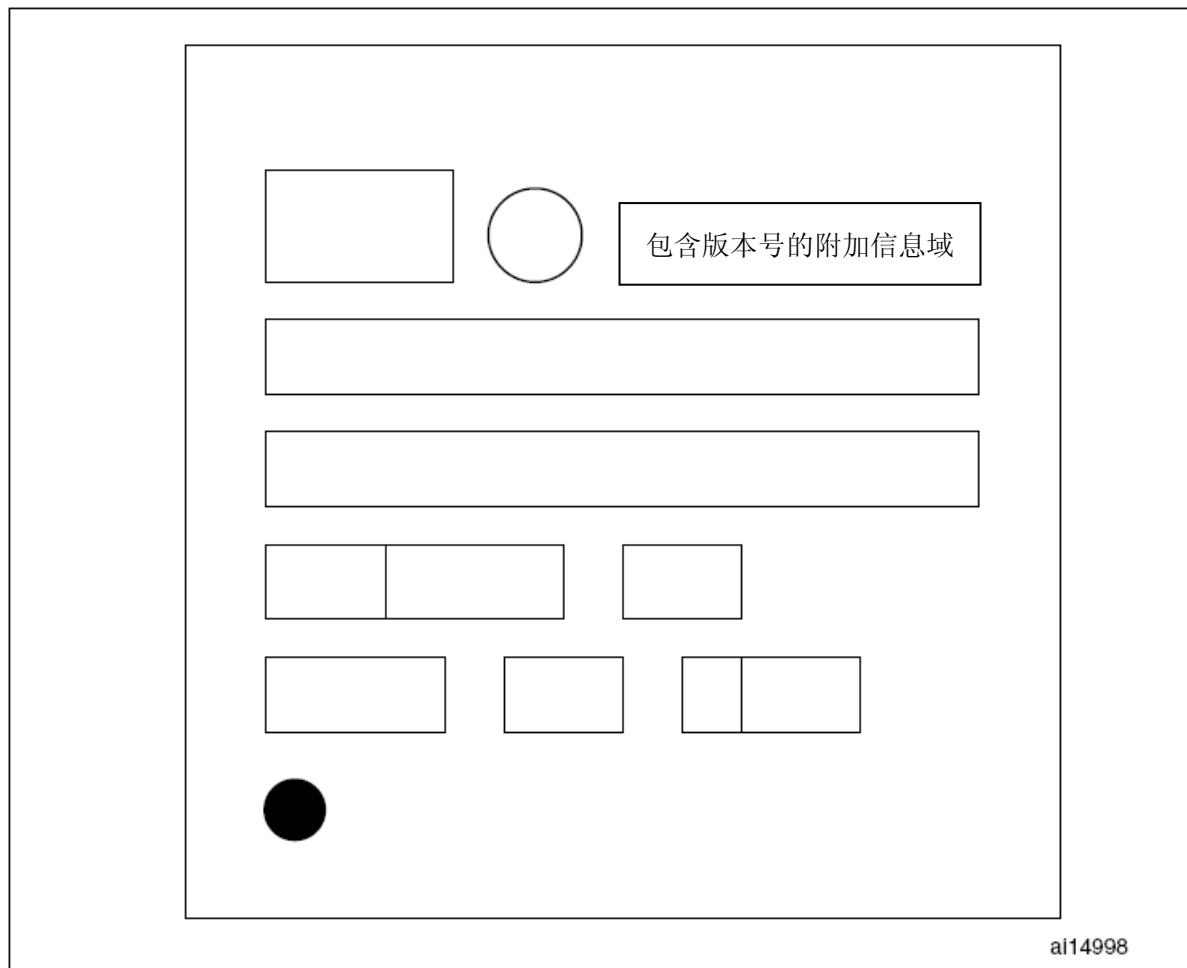
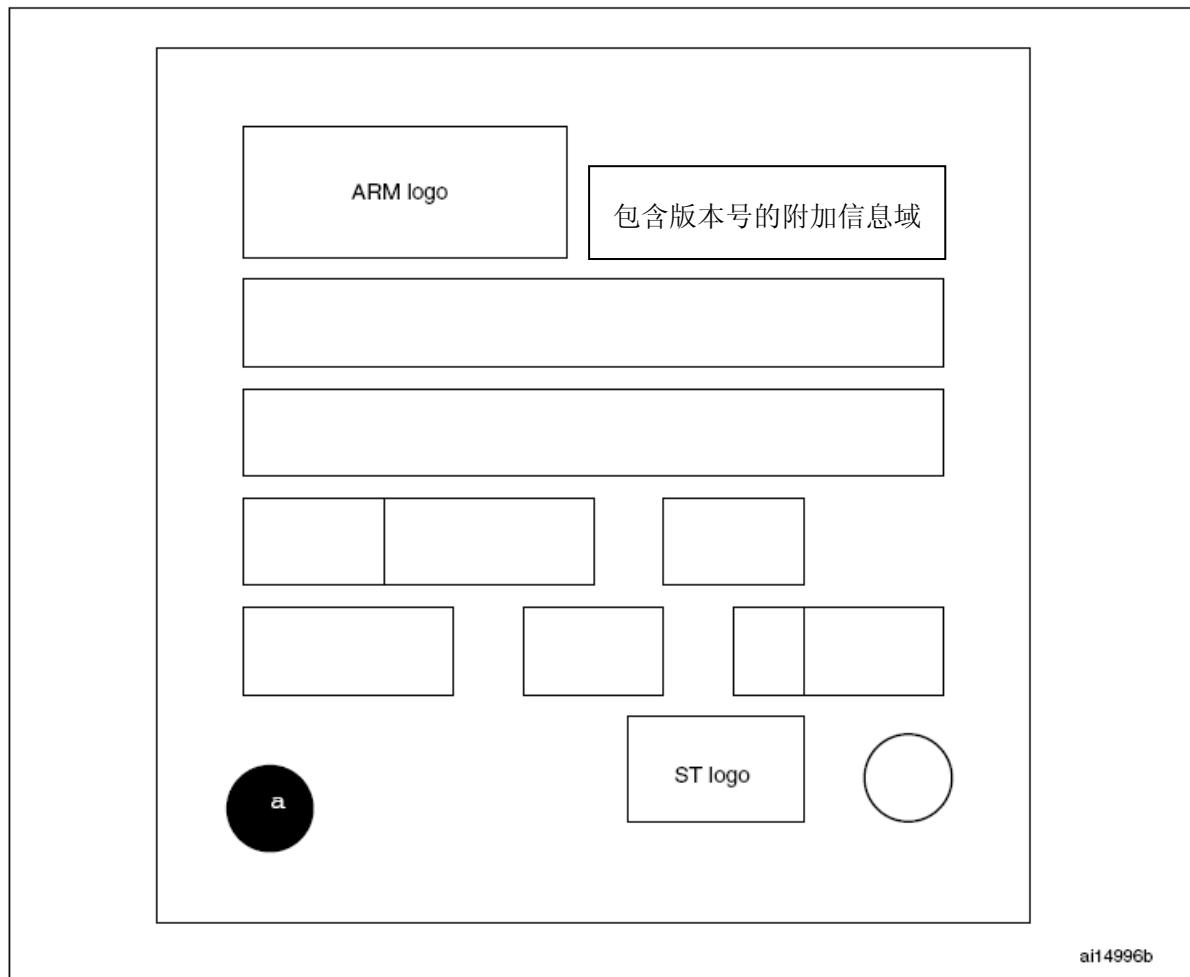


图3 LQFP64 封装的正面俯视图



ai14996b

图4 LQFP48 封装的正面俯视图

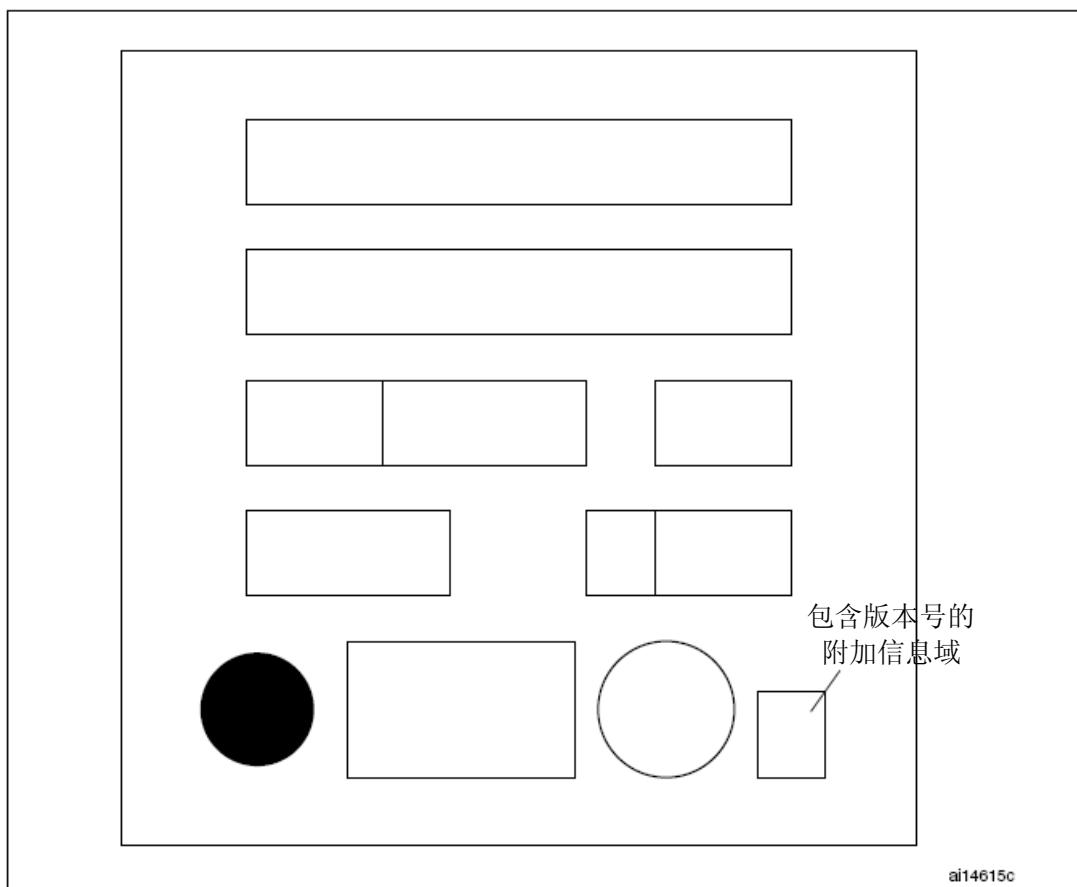
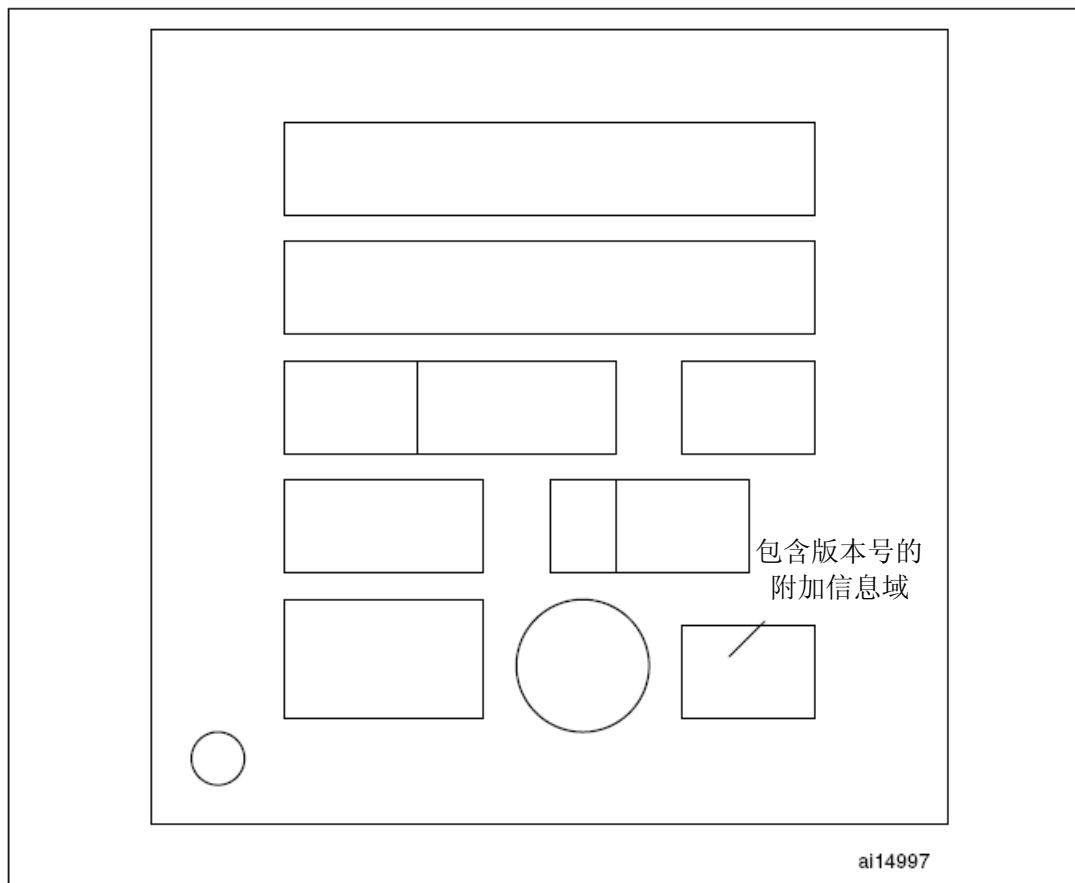


图5 VFQFPN36 封装的正面俯视图



版本修订记录

表5 文档版本记录

日期	版本	变化
2008年3月28日	1	初始版本
2008年4月7日	2	增加2.2节: WFI/WFE指令之后读Flash存储器 在2.4.1节: USART1_RTS和CAN_TX, 给出了暂时解决办法
2008年5月23日	3	勘误手册提供Y版本的信息。 增加第2.3节: 软件不能读调试寄存器。少量文字修改。
2008年7月18日	4	增加第2.5节: PVD和USB唤醒事件。
2008年10月1日	5	勘误手册提供STM32F102xx中容量产品信息。尽管32K闪存的产品不再属于中容量产品, 但本勘误手册仍适用于销售代码不带尾缀"A"的产品。 增加第2.6节: 最新的编译器兼容问题。 增加图1: LFBGA100封装的正面俯视图, 图3: LQFP64封装的正面俯视图。 更正了和图4: LQFP48封装的正面俯视图。
2009年2月11日	6	第1章: ARM™ 32-bit Cortex®-M3 的使用限制, 增加表3: Cortex-M3核心局限性和对微控制器的影响。 增加下列内容: - 第2.7节: 边缘扫描TAP: "capture IR"状态之后送出错误的数据 - 第2.8节: 相对于设置STRT位, Flash存储器的BSY位延后 - 第2.9节: I2C外设 - 第2.10节: 通用定时器 - 第2.11节: LSI时钟稳定时间 - 表4: 芯片局限性列表