



#### 芯片的识别

该勘误表适用于意法半导体的STM32F101xC/D/E基本型和STM32F103xC/D/E增强型大容量系列中的Z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<sup>(1)</sup>

销售型号	标注在芯片上的版本代码 <sup>(2)</sup>
STM32F103xC, STM32F103xD, STM32F103xE	“Z”
STM32F101xC, STM32F101xD, STM32F101xE	“Z”

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](#)。

表2 芯片概览

涉及到的芯片	芯片型号
STM32F101xCDE	STM32F101RC STM32F101VC STM32F101ZC
	STM32F101RD STM32F101VD STM32F101ZD
	STM32F101RE STM32F101VE STM32F101ZE
STM32F103xCDE	STM32F103RC STM32F103VC STM32F103ZC
	STM32F103RD STM32F103VD STM32F103ZD
	STM32F103RE STM32F103VE STM32F103ZE

译著：本文英文原文下载地址：<http://www.st.com/stonline/products/literature/es/14732.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大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大容量产品的影响
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SLEEPONEXIT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大容量产品的影响

只有下面涉及的局限性才对STM32F10xxx大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大容量产品(Cortex-M3 r1p1-01rel0)没有影响，也与STM32F10xxx大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：SDIO 和TIM8
	2.4.7节：SDIO 和经过重映射的TIM3
	2.4.8节：SDIO 和经过重映射的USART3以及USART4
	2.4.9节：FSMC和 I2C1以及TIM4_CH2
	2.4.10节：FSMC和 经过重映射的USART2
	2.4.11节：FSMC与 经过重映射的USART3和TIM1
2.5节：PVD和USB唤醒事件	
2.6节：SPI3在I <sup>2</sup> S从模式：在I2S3_WS和I2S3_CK之间的时序敏感	
2.7节：边缘扫描TAP：“capture IR”状态之后送出错误的的数据	
2.8节：相对于设置STRT位，Flash存储器的BSY位延后	
2.9节：I2C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	
2.12节：在FSMC存储器空间的多主访问	
2.13节：对SDIO卡的多字节支持受限	

### 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(Flash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\_AHBNR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。在这些引脚上，Cortex的EVENTOUT输出功能也不能同时作为复用功能而使用。

### 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CAN的时钟被禁止了，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 SDIO 和TIM8

#### 问题描述

下述情况时发生冲突：

- SDIO配置成1位或者4位模式，TIM8\_CH4配置成输出。

彼此冲突的信号如下：

- TIM8\_CH4和SDIO\_D1

#### 暂时解决办法

当使用SDIO时，就不能使用TIM8\_CH4的输出模式。

### 2.4.7 SDIO 和经过重映射的TIM3

#### 问题描述

当SDIO配置成1位或者4位模式，TIM3的通道被重映射到PC6到PC9，并且配置成输出模式，冲突发生在以下信号之间：

- TIM3\_CH4和SDIO\_D1

**暂时解决办法**

当使用SDIO时，就不要使用TIM3\_CH4的输出模式。

**2.4.8 SDIO 和经过重映射的USART3以及USART4****问题描述**

当SDIO配置成1位模式，如下信号会和被重映射的USART3\_TX引脚以及UART4\_TX引脚发生冲突。

- USART3\_TX和SDIO\_D2
- UART4\_TX和SDIO\_D2

**暂时解决办法**

当使用SDIO的同时，要使用USART3\_TX，就用它的默认引脚分配(PB10)或者将其重映射到PD8引脚。

当使用SDIO的同时，就不要使用UART4\_TX。

**2.4.9 FSMC和 I2C1以及TIM4\_CH2****问题描述**

当使用FSMC功能时，NADV被配置成复用输出，该信号会被默认地置位为'1'，这样就和TIM4\_CH2和I2C1的SDA信号发生冲突。

**暂时解决办法**

当使用FSMC的同时，不要使用TIM4\_CH2作为输出。如果要使用I2C1并且封装允许，就把该功能重映射到PB8/PB9。

**2.4.10 FSMC和 经过重映射的USART2****问题描述**

当使用FSMC，并且PD4、PD5和PD7被配置成复用输出时，下述信号之间会发生冲突：

- 重映射到PD4的USART2\_RTS信号与FSMC\_OEN冲突
- 重映射到PD5的USART2\_TX信号与FSMC\_WEN冲突
- 重映射到PD7的USART2\_CK信号与FSMC\_NE1/FSMC\_NCE2冲突

**暂时解决办法**

使用USART2的默认配置(无重映射)。

**2.4.11 FSMC与 经过重映射的USART3和TIM1****问题描述**

当在8位模式下使用FSMC，并且在未用的FSMC数据线8~15被配置成复用输出时，下述信号之间会发生冲突：

- 分别重映射到PD8和PD10的USART3\_TX和USART3\_CK信号，将分别与FSMC\_D13和FSMC\_D15冲突。
- 当TIM1\_CH2、TIM1\_CH3N、TIM1\_CH3和TIM1\_CH4用于输出并分别重映射到PE11~PE14时，它们分别与FSMC\_D8至FSMC\_D11冲突。

**暂时解决办法**

- 使用USART3的部分重映射或不用重映射功能。
- 当重映射到PE11至PE14时，不要使用TIM1\_CH2、TIM1\_CH3N、TIM1\_CH3和TIM1\_CH4作为输出功能。

## 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，导致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

如果应用程序必须需要使用PVD或USB唤醒，则使用中断源和WFI指令。

## 2.6 SPI3在I<sup>2</sup>S从模式：在I2S3\_WS和I2S3\_CK之间的时序敏感

### 问题描述

当SPI3配置为I<sup>2</sup>S从模式，如果I2S3\_WS信号相对于I2S3\_CK的有效边沿到来的太早，则将导致一个错误地通信。因此，根据时钟极性和选择的音频模式，数据或者从开始至结束错了一位，或者第一个左右数据项丢失。

### 暂时解决办法

无。

## 2.7 边缘扫描TAP：“capture IR”状态之后送出错误的数字

### 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 I<sup>2</sup>C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I<sup>2</sup>C的DMA模式，除非作为主设备时只接收一个字节。
2. 使用I<sup>2</sup>C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

I<sup>2</sup>C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. 使用警告响应地址(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<sub>LSI</sub>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。

## 2.12 在FSMC存储器空间的多主访问

### 问题描述

在FSMC存储器空间(地址空间为0x6000 0000~0xA000 0FFF)的多主访问, 可能产生错误, 或者是总线失效, 或者是DMA传输错误。

多主访问可以是:

- DMA1与DMA2
- DMA1与CPU
- DMA2与CPU

### 暂时解决办法

如果在FSMC地址空间上需要多主访问, 软件要保证在同一时间没有多主访问。

## 2.13 对SDIO卡的多字节支持受限

### 问题描述

SDIO标准允许多字节传输(传输1、2、3、至512字节的数据), 这是扩展了SD标准的块传输。

在一个多字节传输的操作中, 使用CMD53命令传输的字节数目必须是2的幂次并且介于1~512之间, 即传输的字节数目必须是1、2、4、8.....512。

### 暂时解决办法

如果应用程序需要读出的字节数目N不是2的幂次, 并且SDIO卡支持多块传输模式, 可以这样使用CMD53多块命令:

- 配置SDIO块长度为1个字节,
- 使用CMD53命令读出N个字节。

## 2.14 在SPI2/I2S2从模式和16位帧模式下意外产生TXE

### 问题描述

当使用SPI2/I2S2的从模式的发送或全双工操作时, 并且数据帧配置为16位时, TXE标志可能在设置RXNE之前变高2次, 即有可能在发送已经在数据寄存器中的之前设置TXE。这会导致发送时的数据丢失。

### 暂时解决办法

1. 使用8位数据帧模式, 而不是16位数据帧模式。

如果主设备可以在每次数据传输后停止时钟, 则可以使用16位模式。此时SPI2/I2S2从设备可以有足够的时间在RXNE标志变高时加载数据寄存器。

## 附录A 产品上版本号标示

图1到图6分别只显示了LFBGA144、LFBGA100、WLCSP64、LQFP144、LQFP100和LQFP64这六种不同封装的芯片上版本标志的记号。

图1 LFBGA144 封装的正面俯视图

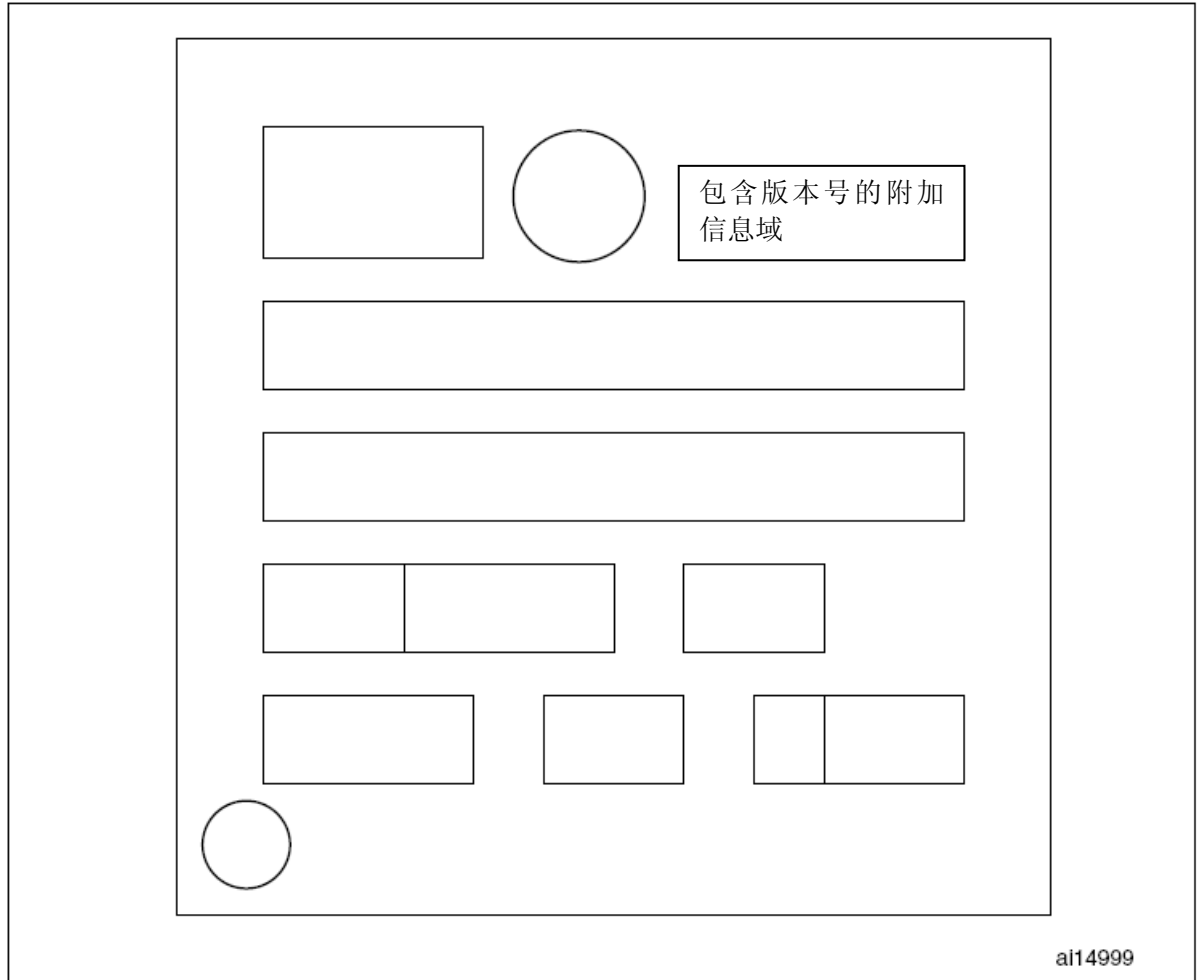


图2 LFBGA100 封装的正面俯视图

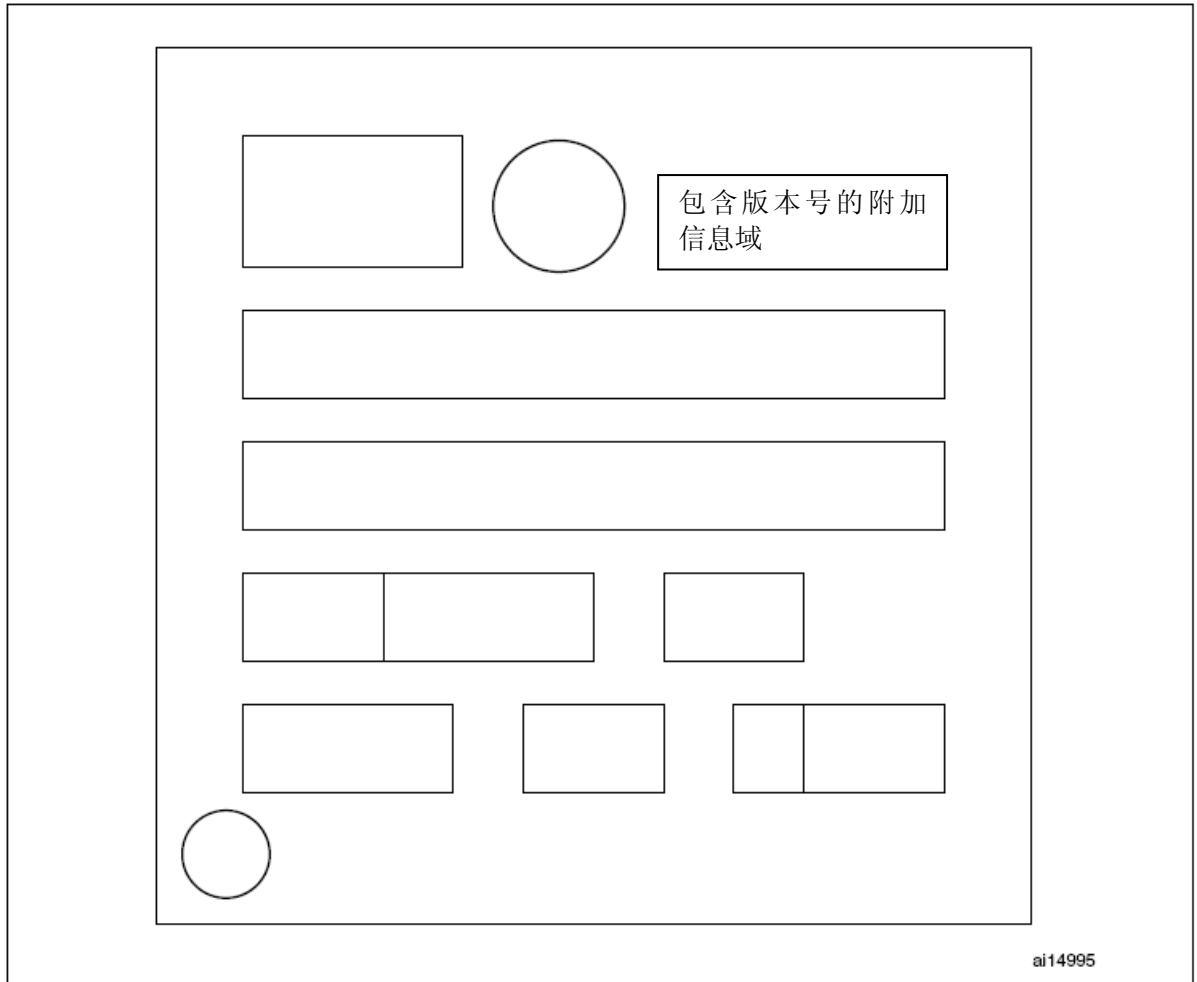


图3 WLCSP64 封装的正面俯视图

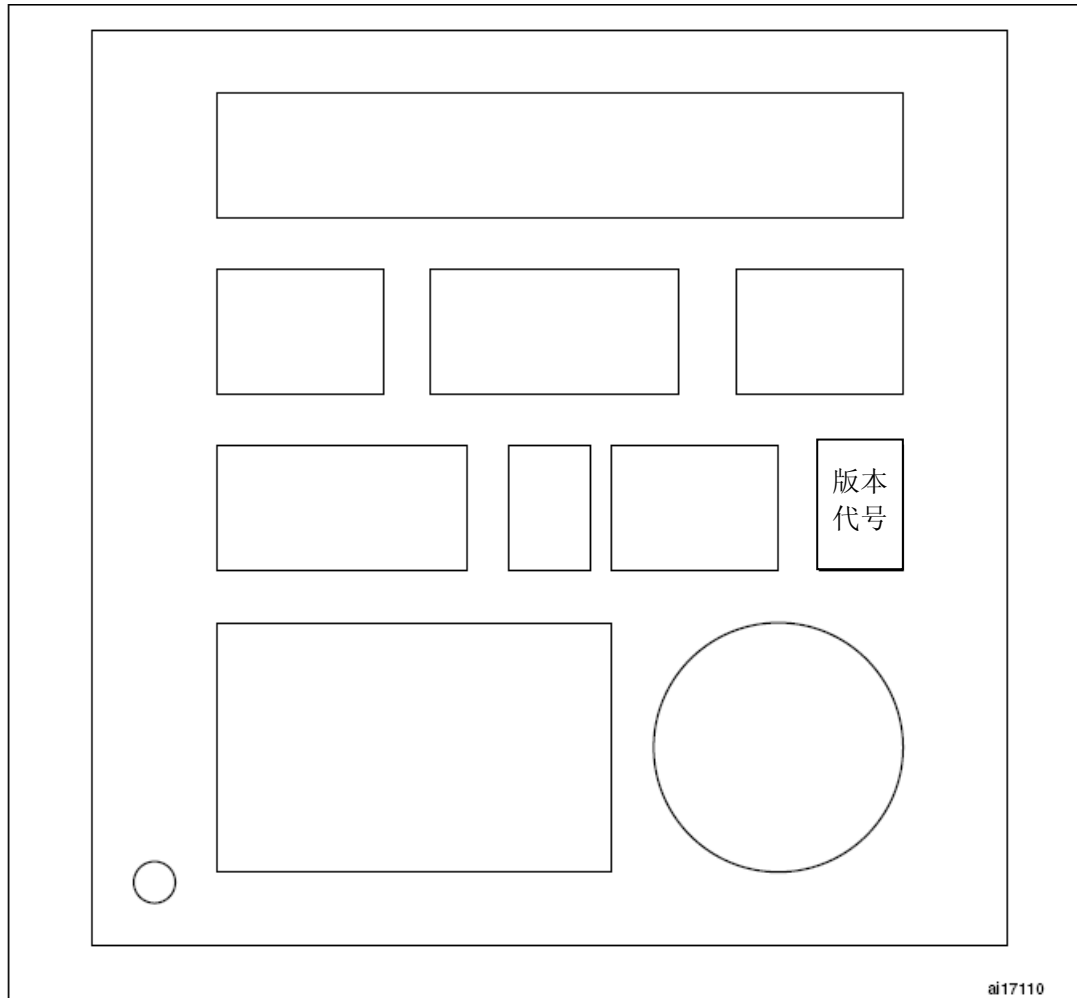


图4 LQFP144 封装的正面俯视图

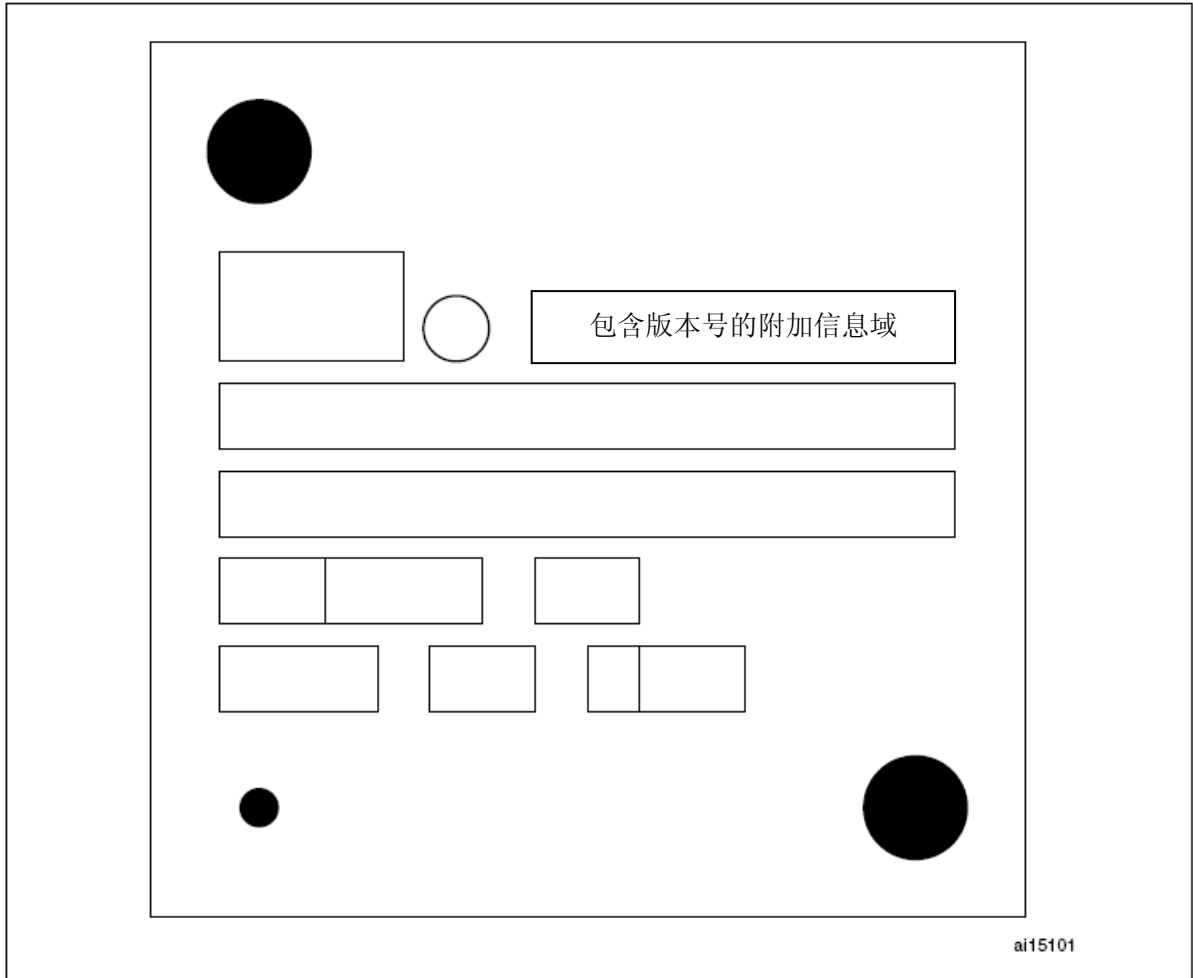


图5 LQFP100 封装的正面俯视图

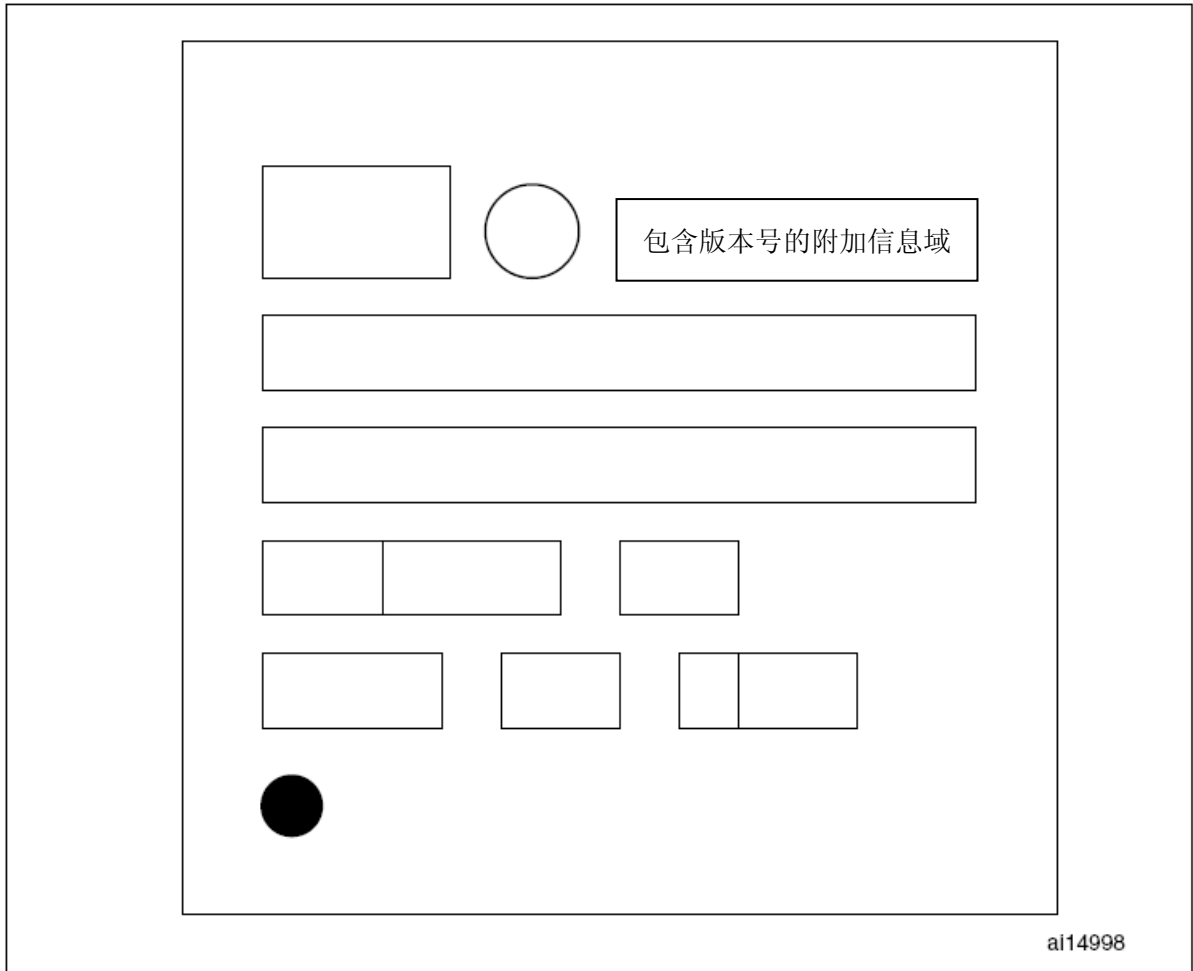
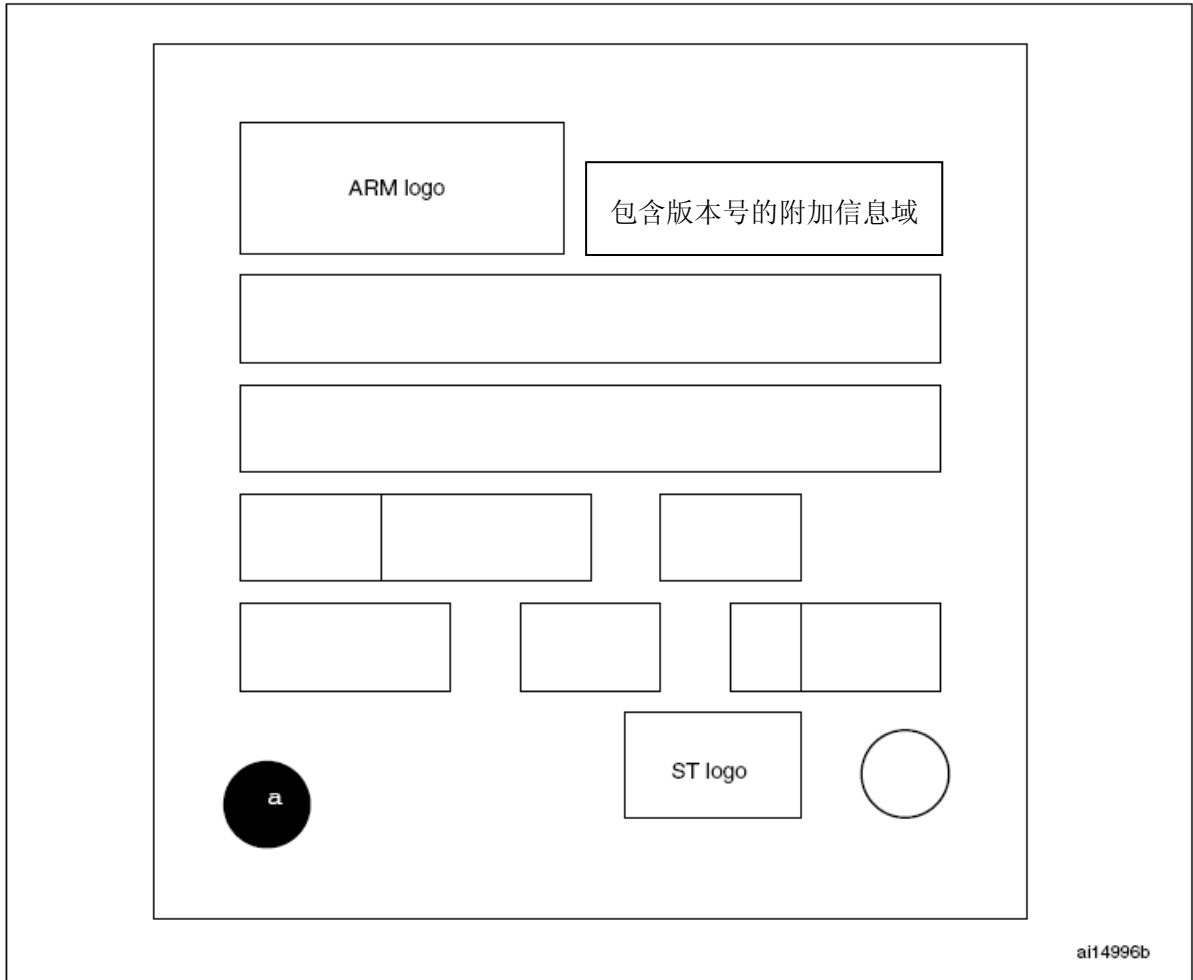


图6 LQFP64 封装的正面俯视图





## 版本修订记录

表5 文档版本记录

日期	版本	变化
2008年5月23日	1	初始版本
2008年7月21日	2	增加2.5节：PVD和USB唤醒事件
2008年9月29日	3	更正图6：LQFP64封装的正面俯视图
2009年2月11日	4	增加第1章：ARM™ 32-bit Cortex®-M3 的使用限制
		增加2.6节：SPI3在I <sup>2</sup> S从模式：在I2S3_WS和I2S3_CK之间的时序敏感
		增加下述局限性： - 边缘扫描TAP：“capture IR”状态之后送出错误的数 - 相对于设置STRT位，Flash存储器的BSY位延后 - I2C外设 - 通用定时器 - LSI时钟稳定时间 - 2.12节：在FSMC存储器空间的多主访问 - 2.13节：对SDIO卡的多字节支持受限
		增加表4：芯片局限性列表
2009年6月22日	5	修改了第2.4节的说明文字 更新了第2.4.10节：FSMC和 经过重映射的USART2 增加了第2.4.11节：FSMC与 经过重映射的USART3和TIM1 澄清了第2.13：对SDIO卡的多字节支持受限 增加了第2.14：在SPI2/I2S2从模式和16位帧模式下意外产生TXE 增加了图3：WLCSP64封装的正面俯视图