

# ADP32F03X数字信号处理器

## 勘误表

编号: JXDZ7.381.013KWB

**Advancechip**



**Electronics**

**湖南进芯电子科技有限公司**

2023 年 11 月

V1.0

## 目 录

1 前言 .....	4
2 器件标识 .....	4
3 功能规范的使用说明和已知设计勘误项 .....	6
3.1 使用说明 .....	6
3.1.1 PIE: 连续PIEACK写入和手动CPU中断掩码清除后的虚假嵌套中断 .....	6
3.1.2 CAN引导程序: 内部振荡器容差不足以在高温条件下运行CAN .....	6
3.2 功能规范的已知设计勘误项 .....	8
1.勘误项 ADC: 温度传感器最小采样窗口要求 .....	9
2.勘误项ADC: 失调自校准要求 .....	9
3.勘误项ADC: 在上次转换的第14个周期时采样结束时的ADC结果转换, ACQPS=6或7 .....	10
4.勘误项 ADC: 初始转换 .....	10
5.勘误项 ADC: 在转换过程中写入 ADCNONOVERLAP或RESET时, ADC可能无响应 .....	10
6.勘误项ADC: ADC 控制位 .....	13
7.勘误项ADC: 上电时超出电流消耗规范 .....	13
8.勘误项Memory: 超出有效内存的预取 .....	14
9.勘误项GPIO: GPIO限定 .....	14
10.勘误项代码安全模块: 内存访问与代码安全模块 (CSM) PWL 寄存器冲突 .....	14
11.勘误项eCAN: 中止确认位未置位 .....	15
12.勘误项eCAN: 发送操作意外停止 .....	16
13.勘误项LIN: 多缓冲模式下的SCI操作 .....	16
14.勘误项LIN: 扩展帧模式下可能的数据损坏 .....	16
15.勘误项 LIN: 如果在单缓冲区模式中由于校验和字段期间的比特错误 (BE), 导致先前的传输未完成, 则LIN传输受到影响 .....	16
16.勘误项 LIN: 当在1.5s的等待时间内接收到新Header时, TOA3WUP被标记 .....	17
17.勘误项 LIN: 校验和字节重启接收期间的帧错误 (FE)。如果接收到新Header, 则标记帧错误 .....	17
18.勘误项LIN: 当LIN从机更快地适应传入Header时, 可能无法在预期时间帧内标记无响应错误(NRE)/超时 .....	18
19.勘误项LIN: LIN拒绝错误起始位之后的有效字节 .....	18
20.勘误项LIN: 同步字段接收不完整会导致下一个Header的丢失 .....	18

21.勘误项LIN: 从机接收到长度大于500 $\mu$ s的唤醒请求时, 它不会向唤醒后的首个 Header发送响应.....	19
22.勘误项LIN: 仅由同步中断组成的不完整帧Header将导致从机无法接收下一个完整Header.....	19
23.勘误项零引脚振荡器: 振荡器频率参数的更改.....	19
24.勘误项看门狗: WDCLK源为OSCCLKSRC2时, CPU看门狗操作不正确.....	20
25.勘误项振荡器: CPU时钟切换到INTOSC2可能导致复位后时钟丢失的情况.....	20
26.勘误项eQEP: 错过第一个索引事件.....	21
27.勘误项eQEP: 在索引期间改变方向时位置计数器复位不正确.....	21
28.勘误项eQEP: GPIO异步模式下的eQEP输入.....	22
29.勘误项ePWM: 如果Trip在屏蔽窗口结束时保持有效, 则可能产生ePWM毛刺.....	22
30.勘误项Boot ROM: Flash API [Flash_Erase() Function].....	23

## 1 前言

本档描述了ADP32F03x微控制器功能规范的更新。

更新适用于：

- 48引脚超小四方扁平无引脚(VQFN) 封装
- 56 引脚超小四方扁平无引脚(VQFN) 封装
- 64 引脚薄型四方扁平 (LQFP) 封装
- 80 引脚薄型四方扁平 (LQFP) 封装

## 2 器件标识

图1提供了ADP32F03x器件标识的示例并对标识的定义进行了说明。用户可以通过封装顶部所示的符号来确定器件的版本，如图1所示。某些原型器件的标识可能与图示标识有所不同。图2所示为器件的命名规则。

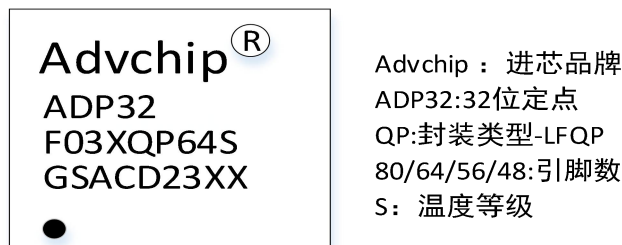


图1 器件标识示例

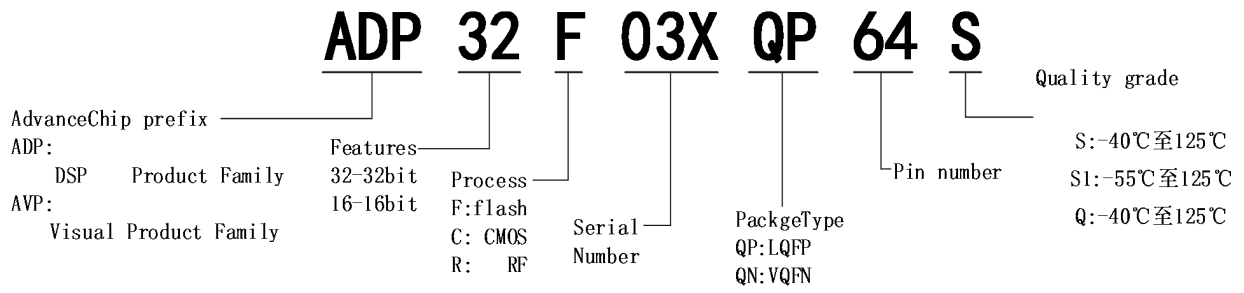


图2 器件命名规则

### 3 功能规范的使用说明和已知设计勘误项

#### 3.1 使用说明

使用说明强调并描述了器件行为可能与预期或手册所述行为不匹配的情况，可能包括影响器件性能或功能正确性的行为。这些使用说明将合并到未来器件文档更新中（如器件数据手册），且其所述行为在将来的器件版本中不会改变。

##### 3.1.1 PIE: 连续PIEACK写入和手动CPU中断掩码清除后的虚假嵌套中断

用于嵌套中断的某些代码序列允许CPU和PIE进入不一致的状态，从而触发不必要的中断。进入该状态所需的条件为：

1. 清除一个PIEACK后，紧接着是一个全局中断使能（EINT或asm（“CLRC INTM”））。
2. 嵌套中断为其组清除一个或多个PIEIER位。

是否触发不必要的中断取决于系统中其他中断的配置和时序。在大多数应用程序中，这种情况很少见甚至不存在。如果发生这种情况，不必要的中断将是嵌套中断PIE组中的第一个中断，并将在嵌套中断重新使能CPU中断（EINT或asm（“CLRC INTM”））后触发。

**应对方法：**在PIEACK写入和CPU中断使能之间添加一个NOP。代码示例如下：

```
//Bad interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF; //Enable nesting in the PIE
EINT; //Enable nesting in the CPU

//Good interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF; //Enable nesting in the PIE
asm(" NOP"); //Wait for PIEACK to exit the pipeline
EINT; //Enable nesting in the CPU
```

##### 3.1.2 CAN引导程序：内部振荡器容差不足以在高温条件下运行CAN

器件引导ROM中的CAN引导加载程序使用内部振荡器作为CAN位时钟源。在高温下，内部振荡器的频率可能会偏离到足以阻止消息接收的程度。

**应对方法：**在调用CAN引导加载程序之前，重新校准内部振荡器。这可以在应用程序代码中完成。为增加灵活性，可以将Wrapper函数编程到器件的OTP存储器中。

### 3.2 功能规范的已知设计勘误项

**表1 勘误项目录**

序号	勘误项
1	ADC: 温度传感器最小采样窗口要求
2	ADC: 失调自校准要求
3	ADC: 自上次转换的第 14 个周期时结束时的 ADC 结果转换, ACQPS = 6 或 7
4	ADC: 初始转换
5	ADC: 在转换过程中写入 ADCNONOVERLAP或RESET时, ADC 可能无响应
6	ADC: ADC 控制位
7	ADC: 上电时超出电流消耗规范
8	Memory: 超出有效内存的预取
9	GPIO: GPIO限定
10	代码安全模块: 内存访问与代码安全模块(CSM)PWL寄存器冲突
11	eCAN: 中止确认位未置位
12	eCAN: 发送操作意外停止
13	LIN: 多缓冲模式下的SCI操作
14	LIN: 扩展帧模式下可能的数据损坏
15	LIN: 如果在单缓冲区模式中由于校验和字段期间的比特错误 (BE), 导致先前传输未完成, 则LIN传输受到影响
16	LIN: 当在1.5s的等待时间内接收到新Header时, TOA3WUP被标记
17	LIN: 校验和字节重新启动接收期间的帧错误 (FE)。如果接收到新的Header, 则标记帧错误。
18	LIN:当LIN从机更快地适应传入Header时, 可能无法在预期时间帧内标记无响应错误(NRE)/超时
19	LIN: LIN拒绝错误起始位之后的有效字节
20	LIN: 同步字段接收不完整导致下一个Header丢失
21	LIN: 当从机接收到长度大于500 μs的唤醒请求时, 它不会向唤醒后的首个Header发送响应
22	LIN: 仅由同步中断组成的不完整帧Header将导致从机无法接收下一个完整Header
23	零引脚振荡器: 振荡器频率参数的更改
24	看门狗: WDCLK源为OSCCLKSRC2时, CPU看门狗操作不正确
25	振荡器: CPU时钟切换到INTOSC2可能导致复位后时钟丢失的情况
26	eQEP: 错过第一个索引事件
27	eQEP: 在索引期间改变方向时位置计数器复位不正确
28	eQEP: GPIO异步模式下的eQEP输入
29	ePWM: 如果Trip在屏蔽窗口结束时保持有效, 则可能产生ePWM毛刺
30	Boot ROM: Flash API[Flash_Erase() Function]



## 1.勘误项

### ADC：温度传感器最小采样窗口要求

#### 描述

如果使用最小采样窗口（6个ADC时钟，频率为60 MHz，116.67 ns），则温度传感器转换的结果可能会产生较大的误差，从而使系统不可靠。

#### 应对方法

1. 如果对温度传感器的进行两次采样以避免第一次采样偏差的问题，则温度传感器结果有效。这相当于给 S/H 电路足够的充电时间。
2. 在所有其他条件下，用于对温度传感器进行采样的采样保持窗口不应小于550 ns。

## 2.勘误项

### ADC：失调自校准要求

#### 描述

Device\_cal()的出厂失调校准可能无法确保 ADC 失调在客户系统的所有工作条件下都符合规范要求。

#### 应对方法

1. 为确保失调保持在数据手册的“单次重新校准”规范范围内，请在 Device\_cal()完成并配置ADC后执行AdcOffsetSelfCal()函数。
2. 为确保失调保持在数据手册的“定期重新校准”规范范围内，随着温度漂移，请定期执行AdcOffsetSelfCal()函数。

### 3.勘误项 **ADC：在上次转换的第14个周期时采样结束时的ADC结果转换**

**描述** 该问题的详细描述见：ADP32F03X数字信号处理器ADC故障勘误说明

### 4.勘误项 **ADC：初始转换**

**描述** 当ADC转换由任何触发源以顺序或同步采样模式启动时，第一个采样可能不是正确的转换结果。

**应对方法** 对于顺序模式，请在每个转换序列开始时放弃第一个采样结果。例如，如果应用程序调用给定的一系列转换 SOC0→SOC1→SOC2，定期启动，则将序列设置为 SOC0→SOC1→SOC2→SOC3，并且仅使用最后三个转换 ADCRESULT1、ADCRESULT2、ADCRESULT3，从而丢弃 ADCRESULT0。

对于同步采样模式，在每个转换序列开始时丢弃 A 和 B 通道的第一个采样结果。

用户应用程序应验证此解决方法在其应用程序中是否可接受。

对于30 MHz及以下工作频率，可通过在ADCTRL2寄存器中的ADCNONOVERLAP和CLKDIV2EN位上写入1来完全解决此问题。当CPU时钟= 60 MHz时，这将提供30 MHz ADC时钟，并且仅在ADC完成任何挂起转换时允许对ADC通道进行采样。

### 5.勘误项 **ADC：在转换过程中写入 ADCNONOVERLAP或RESET时，ADC可能无响应**

**描述** 当在转换过程中修改 ADCCTL2[ADCNONOVERLAP] 时，ADC 可能会进入无响应状态。在这种情况下，如果不进行器件复位，则无法从ADC进行进一步转换。

有两种不同的方法会导致这种情况：

- 在转换过程中写入 ADCCTL2[ADCNONOVERLAP]。
- 在转换过程中写入 ADCCTL1[RESET]。

**应对方法** 按照以下顺序修改 ADCCTL2[ADCNONOVERLAP] 或写入 ADCCTL1[RESET]：

1. 将所有 SOC 触发源 ADCSOCxCTL[TRIGSEL] 设置为 0。
2. 将所有 ADCINTSOCSEL1/2 设置为 0。

3. 确保没有另一个待处理的 SOC (这可以通过轮询 SOC 标志来完成)。
4. 等待所有转换完成。
  - a.  $ADCCTL2[CLKDIV2EN] = 0$ ,  $ADCCTL2[CLKDIV4EN] = x \rightarrow (ACQPS + 13) * 1 \text{ SYSCLKs}$
  - b.  $ADCCTL2[CLKDIV2EN] = 1$ ,  $ADCCTL2[CLKDIV4EN] = 0 \rightarrow (ACQPS + 13) * 2 \text{ SYSCLKs}$
  - c.  $ADCCTL2[CLKDIV2EN] = 1$ ,  $ADCCTL2[CLKDIV4EN] = 1 \rightarrow (ACQPS + 13) * 4 \text{ SYSCLKs}$
5. 修改  $ADCCTL2[ADCNONOVERLAP]$  或写入  $ADCCTL1[RESET]$ 。

以下为一个示例代码。

```
EALLOW;
// Set all SOC trigger sources to software
AdcRegs.ADCSOC0CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC1CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC2CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC3CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC4CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC5CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC6CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC7CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC8CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC9CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC10CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC11CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC12CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC13CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC14CTL.bit.TRIGSEL = 0;
AdcRegs.ADCSOC15CTL.bit.TRIGSEL = 0;

// Set all ADCINTSOCSEL1/2 to 0.
AdcRegs.ADCINTSOCSEL1.bit.SOC0 = 0;
AdcRegs.ADCINTSOCSEL1.bit.SOC1 = 0;
AdcRegs.ADCINTSOCSEL1.bit.SOC2 = 0;
AdcRegs.ADCINTSOCSEL1.bit.SOC3 = 0;
AdcRegs.ADCINTSOCSEL1.bit.SOC4 = 0;
AdcRegs.ADCINTSOCSEL1.bit.SOC5 = 0;
AdcRegs.ADCINTSOCSEL1.bit.SOC6 = 0;
AdcRegs.ADCINTSOCSEL1.bit.SOC7 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC8 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC9 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC10 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC11 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC12 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC13 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC14 = 0;
AdcRegs.ADCINTSOCSEL2.bit.SOC15 = 0;

// Ensure there is not another SOC pending
while (AdcRegs.ADCSOCFLG1.all != 0x0);

// Wait for conversions to complete
// Delay time based on ACQPS = 6 , ADCCTL2[CLKDIV2EN] = 1, ADCCTL2[CLKDIV4EN] = 0
// 7 + 13 ADC Clocks = 20 ADCCLKS -> 40 SYSCLKS
asm(" RPT#40|NOP");
// ADCCTL2[ADCNONOVERLAP] = <new value>;
// ADCCTL1[RESET] = 1;

EDIS;
```

## 6.勘误项 ADC: ADC 控制位

**描述** ADC 控制寄存器:

1. 在0x7101地址的ADCTL2寄存器, 包含以下位:
  - CLKDIV2EN, 一种 2 分频使能, 可将 CPU 时钟频率调整为1/2 以供 ADC 使用。
  - ADCNONOVERLAP, 一种位使能, 可消除转换/采样重叠时序
2. 在现有的ADCSOCPRIORITYCTL寄存器中添加了位域ONESHOT。启用后, ONESHOT 位字段仅在同时接收多个 SOC 时让第一个 SOC 完成。这适用于处于循环优先级的所有 SOC。高优先级模式下的 SOC 不受此功能的影响。

**应对方法** 不适用

## 7.勘误项 ADC: 上电时超出电流消耗规范

**描述** 当ADC模块首次上电时, 在启动采样之前, 它可以消耗数据手册中所列电流 (IDDA) 的2倍。

**应对方法** 在ADC最初上电之前, 启动一次转换。一旦满足转换时序, 建议在设置ADC之前进行ADC复位以恢复ADC的初始状态, 就像添加此额外操作之前时所做的那样。  
(代码如下所示。如果系统可以容忍此电流量, 则无需更改设置代码。请注意, 一旦ADC处理了第一次转换, 电流将回到数据手册限值内。

```
EALLOW;  
SysCtrlRegs.PCLKCR0.bit.ADCENCLK=1;           //Enable Clocks to ADC module  
AdcRegs.ADCSOFRC1.bit.SOC0 = 1;                //Issue dummy conversion  
asm(" rpt #19 || nop");                          //Wait for conversion to propagate  
AdcRegs.SOCPRCTL.bit.SOCPRIORITY = 1;          //Change Priority Control to reset  
AdcRegs.SOCPRCTL.bit.SOCPRIORITY = 0;          //round robin pointer  
EDIS;
```

**8.勘误项**
**Memory: 超出有效内存的预取**
**描述**

CPU预取超出流水线中当前活动的指令。如果在超出有效内存之外进行预取，那么CPU可能会接收到无效的操作码。

**应对方法**

预取队列的深度为8 x16字。因此，代码不应位于有效内存末端的8字以内。

此限制适用于器件上的所有内存区域和所有内存类型（闪存、OTP、SARAM）。可以跨两个有效内存块之间的边界进行预取。

示例1：M1在地址0x7FF处结束，且其后没有另一个内存块。M1中的代码应存储在不超过地址0x7F7的范围内。地址0x7F8-0x7FF不能用于编码。

示例2：M0在地址0x3FF结束，其后为有效内存（M1）。M0中的代码最高可存储到地址0x3FF中（包含0x3FF）。代码也可以交叉存入M1，最高地址为0x7F7（包含0x7F7）。

**9.勘误项**
**GPIO: GPIO限定**
**描述**

如果使用“m”限定采样(m=3或6)，将GPIO引脚配置为“n”个SYSCLKOUT周期限定(其中  $1 \leq n \leq 510$ )，则可能获得宽度为 $[n*m-(n-1)]$ 的输入脉冲限定(而不是 $n*m$ )。这取决于异步GPIO输入信号相对于内部预分频时钟的相位对齐，因此是不确定的。这种错误限定发生的概率是“1/n”。

**错误示例:**

如果 $n=510$ ， $m=6$ ，则需要 $(n*m)=3060$ 个SYSCLKOUT周期的GPIO输入宽度才能通过限定。然而，由于上述问题，可能获得限定的最小GPIO输入宽度为 $[n*m-(n-1)]=3060-509=2551$ 个SYSCLKOUT周期。

**应对方法**

无。确保在输入限定设计中留有足够的余量。

**10.勘误项**
**代码安全模块: 内存访问与代码安全模块 (CSM) PWL 寄存器冲突**
**描述**

读取地址0x3D 7FF8-0x3D 7FFF将激活CSM，锁定器件。

**应对方法**

请勿读取这些地址。如果读取了这些地址，导致CSM激活，则首先可以通过复位或设置FORCESEC位来解锁器件，然后执行密码匹配流程。

## 11.勘误项

### eCAN: 中止确认位未置位

#### 描述

在设置传输请求重置(TRR)寄存器位以中止消息之后,有些罕见的情况下 TRR<sub>n</sub>和 TRS<sub>n</sub> 位将在不设置中止确认(AAn)位的情况下被清除。传输本身被正确中止,但没有断言任何中断,也没有任何挂起操作的指示。

为了使这种罕见的情况发生,必须发生以下所有情况:

1. 前一条消息不成功,可能是因为仲裁丢失,也可能是因为总线上没有节点能够确认它,或者是因为传输导致错误帧。上一条消息不必来自当前正在尝试传输中止的同一邮箱。
2. 邮箱的 TRR<sub>n</sub> 位应该在设置 TRS<sub>n</sub> 位的周期之后的一个 CPU 周期内设置。由于传输不完全而导致的 TRS<sub>n</sub> 位的剩余设置也满足这个条件;也就是说,TRS<sub>n</sub> 位可以在过去设置,但是传输仍然不完全。
3. TRR<sub>n</sub>位必须在精确的SYSCLKOUT周期设置,在这个周期中CAN模块处于空闲状态。CAN模块在不进行数据接收/传输的情况下处于空闲状态。

如果出现这些情况,则邮箱的TRR<sub>n</sub>和TRS<sub>n</sub>位将在设置TRR位之后 $t_{clr}$  SYSCLKOUT周期清除,其中:

$$t_{clr} = [(\text{mailbox\_number}) * 2] + 3 \text{ SYSCLKOUT 周期}$$

如果出现这种情况,则不会设置TAn和AAn位。通常,TA或AA位在TRR位变为零之后设置。

#### 应对方法

发生此问题时,TRR<sub>n</sub>和TRS<sub>n</sub>位将在 $t_{clr}$  SYSCLKOUT周期内清除。要检查这种情况,首先应禁用中断。设置TRR<sub>n</sub>位后,检查TRR<sub>n</sub>位 $t_{clr}$  SYSCLKOUT周期,以确保其仍被设置。TRR<sub>n</sub>位置位表示未出现问题。

如果TRR<sub>n</sub>位被清除,这可能是因为消息的正常结尾,并且相应的TAn或AAn位被设置。检查TAn和AAn位。如果其中一个位置位,则不会出现问题。如果它们都为零,则问题确实发生了。像中断服务例程一样处理这种情况,只是AAn位现在不需要清除。

如果TAn或AAn位置位,当重新启用中断时,将会进入正常的中断例程。

**12.勘误项****eCAN: 发送操作意外停止****描述**

在极少数情况下，观察到eCAN模块的消息发送停止（同时接收操作正常继续）。这种异常状态可以在总线上没有任何错误帧的情况下发生。

**应对方法**

eCAN模块的超时功能（MOTO）可以用于检测这种情况。出现这种情况时，设置并清除CCR位（使用CCE位进行验证）以消除异常情况。

**13.勘误项****LIN: 多缓冲模式下的SCI操作****描述**

当LIN节点处于SCI多缓冲区模式时，如果缓冲区长度被配置为小于8，则第一组数据（即编程的字节数）被正确传输，但随后数据组的第一个字节的传输被破坏。TX EMPTY标志在STOP位传输开始时设置，而不是在STOP位结束时设置。

此问题仅适用于缓冲SCI模式和缓冲区长度小于8字节的情况。

**应对办法**

选择SCI缓冲模式时，始终将缓冲区长度配置为“8”。

**14.勘误项****LIN: 扩展帧模式下可能的数据损坏****描述**

在扩展帧模式中，在单缓冲器模式传输期间，当TX缓冲器（TD0）在前一个数据字节的STOP位传输期间加载新数据时，下一个数据比特的传输将被破坏（即，下一数据字节的起始位将被破坏）。此问题仅在启用扩展帧模式并选择单缓冲模式传输时适用。此问题将同时出现在主节点和从节点上。

**应对办法**

在扩展帧模式下，在单缓冲区模式传输期间，在用新数据加载TD0缓冲区之前，始终等待TX\_EMPTY标志。

**15.勘误项****LIN: 如果在单缓冲区模式中由于校验和字段期间的比特错误（BE），导致先前的传输未完成，则LIN传输受到影响****描述**

在单缓冲模式传输中，如果在校验和字节的传输过程中发生位错误，那么在接收下一个报头之后，在第一个数据字节传输之后将有10 Tbit的空闲时间。这导致传输比配置的响应长度少1个字节，也会影响从机的无响应错误（NRE）计数。此问题仅适用于单缓冲模式传输期间。这适用于主节点和从节点。

**应对方法**

出现位错误时发出软件重置。



**16.勘误项****LIN: 当在1.5s的等待时间内接收到新Header时, TOA3WUP被标记****描述**

只有在应用程序中使用“睡眠”功能时才会出现此问题。当LIN节点将唤醒信号传输到LIN总线上时, 它将遵循以下顺序:

- 发送第一个唤醒信号→ 等待设置TOAWUP标志
- 发送第二个唤醒信号→ 等待设置TOAWUP标志
- 发送第三个唤醒信号→ 等待设置TOAWUP标志
- 等待1.5秒, 如果没有接收到Header, 然后设置TOA3WUP标志

由于这种错误, 即使在1.5秒的等待期内接收到有效的Header, TOA3WUP标志也会被设置, 从而触发中断。这适用于主节点和从节点。

**应对办法**

如果可以接受, 应用程序可以忽略TOA3WUP中断。

**17.勘误项****LIN: 校验和字节重启接收期间的帧错误 (FE)。如果接收到新Header, 则标记帧错误****描述**

当校验和字节接收过程中出现帧错误 (FE) 时, 下一个传入Header将被视为数据, 另一个FE将被标记。传入Header将被正确接收, 通信也将正确进行。这适用于主节点和从节点。

**应对办法**

关键性和软件变通方法的适用性取决于应用程序对FE标志的响应。以下是可能出现的情况:

- a. 在校验和字节接收过程中发生FE, 随后出现无响应错误 (NRE)。  
注释: 如果应用程序在无响应错误中断服务例程 (NRE ISR) 中发出软件重置, 可能会导致传入Header丢失。应用程序必须在软件中处理这种情况。
- b. FE发生在校验和字节接收期间, 随后是NRE; 并且如果下一个报头出现在所分配的时隙之前 (即, 在服务FE和NRE之前), 则将发生另一个FE。

注释: 应用程序必须忽略第二个FE。

不会对传入Header的接收或通信产生任何影响。

## 18.勘误项

### LIN: 当LIN从机更快地适应传入Header时, 可能无法在预期时间帧内标记无响应错误(NRE)/超时

**描述** 如果响应接收未在TFRAME\_MAX时间范围内完成, 则不应标记“无响应错误”(NRE), 如果4秒钟内没有总线活动, 则应标记超时。然而, 当LIN从机适应更快的传入Header时, NRE/超时可能不会在预期的时间范围内被标记。此问题仅适用于SLAVE模式和ADAPTIVE模式(ADAPT位设置)。

**应对办法** 将BRSR寄存器编程到比预期总线波特率值更快的波特率。

## 19.勘误项

### LIN: LIN拒绝错误起始位之后的有效字节

**描述** LIN拒绝在具有错误起始位的数据字节之后的第一个字节。当在标识符或响应数据期间接收到错误的起始位时, 下一个接收到的字节将被拒绝。

**应对办法** 以下是在这种情况下可能发生的情况:

- a. 在数据字节接收期间出现错误的起始位。  
在这种情况下, 除了现有的字节外, 下一个数据字节也将被拒绝。在这种情况下, 将出现无响应错误(NRE)。  
措施: 在无响应错误中断服务例程(NRE ISR)中发出软件重置。
- b. 校验和字节期间出现错误起始位。  
在这种情况下, 下一个有效的起始位评估将在接收到下降沿或接收到传入的同步中断时发生。在这种情况下, 校验和字节将被拒绝, CE标志可能会根据预期的校验和字节进行设置。在任何一种情况下, 下一个传入Header都将被正确接收, 并且模块将重新同步到传入Header。在这种情况下, 将发生NRE。  
措施: 在NRE ISR中发布软件重置。  
注意: 如果在为NRE提供服务之前接收到下一个Header, 则会发生帧错误(FE)(因为下一个同步中断被视为校验和字节)。该FE应被软件忽略。  
应用程序必须确保在接收下一个Header之前, 所有未决的LIN中断都得到了处理。
- c. 在标识符字段期间出现错误的起始位。  
在这种情况下, 传入的数据字节将被视为ID, 并发生ID奇偶校验错误(PE)。  
措施: 在ID PE中断服务例程(ISR)中发出软件重置。  
在上述所有的情况下, LIN模块都不会错过输入的Header, 并始终与输入的Header重新同步。

## 20.勘误项

### LIN: 同步字段接收不完整会导致下一个Header的丢失

- 描述** 当节点正在接收Header时，如果同步字段接收不完整，则将错过下一个Header。当接收到下一个Header时，节点将标记不一致同步字段错误（ISFE），并丢弃该Header。后续Header将被正确接收。因此，总的来说，在错误的同步字段接收之后，节点将错过一个Header。此问题适用于主和从模式（因为在Header传输期间，主节点将在总线上接收该值）。
- 应对办法** 主模式：在发出下一个Header前进行软件重置。  
从模式：无。
- 21.勘误项** **LIN：从机接收到长度大于500 μs的唤醒请求时，它不会向唤醒后的首个 Header 发送响应**
- 描述** 唤醒后，从机要么不发送响应，要么发送损坏的响应。
- 应对办法** 无
- 22.勘误项** **LIN：仅由同步中断组成的不完整帧Header将导致从机无法接收下一个完整Header**
- 描述** 如果从机接收到一个仅有同步中断（没有同步字段）的Header，而下一个传入Header是完整的Header，则从机会被卡住，无法接收最新的传入Header（并且接收缓冲区不会显示最新传入Header的ID）。如果不完整的Header由同步中断+同步字段组成，则可以无错误地接收下一个传入Header。问题是当不完整的Header仅为同步中断时。单独的同步中断可能会导致比特错误，然后是帧错误（因为节点也在接收）。当传输下一个ID时，将看到一个不一致同步字段错误（ISFE），该ID将被忽略。
- 应对办法** 主模式：在发出下一个Header前进行软件重置。  
从模式：无
- 23.勘误项** **零引脚振荡器：振荡器频率参数的更改**
- 描述** 零引脚振荡器现指定了特定温度下的中心频率和温度系数，以计算在任意工作温度下的绝对频率。客户需要检查其温度特性，以确保零引脚振荡器符合客户的要求。
- 应对方法** 如果频率输出不满足所需容差，可使用软件补偿在任意温度下实现优于1%的频率扩展（约10 MHz）。

**24.勘误项****看门狗：WDCLK源为OSCCLKSRC2时，CPU看门狗操作不正确****描述**

当CPU看门狗时钟源是OSCCLKSRC2时，看门狗可能不会产生周期性的复位。

**应对方法**

看门狗的时钟源只能是OSCCLKSRC1 (INTOSC1)。CPU时钟源可以是OSCCLKSRC2或者OSCCLKSRC1 (INTOSC1)。

**25.勘误项****振荡器：CPU时钟切换到INTOSC2可能导致复位后时钟丢失的情况****描述**

在至少两次系统复位（不包括上电复位）后，当应用程序代码试图将CPU时钟源切换到内部振荡器2时，将出现丢失时钟的情况，并且在以下情况下时钟切换将失败：

- X1和X2未使用（X1在未使用时总是连接到低电平）。
- GPIO38（与TCK和XCLKIN多路复用）仅用作JTAG TCK引脚。
- JTAG模拟器已断开连接。

当出现故障情况时，只有上电复位后，时钟丢失情况才会恢复。

**应对方法**

在通过CLKCTL寄存器中的OSCCLKSRCSEL和OSCCLKSRC2SEL位将CPU时钟源切换到INTOSC2之前，用户必须切换CLKCTL寄存器中的XCLKINOFF和XTALOSCOFF位，如下所示：

```
CLKCTL |= 0x6000; // XCLKINOFF = 1, XTALOSCOFF = 1
CLKCTL &=~0x6000; // XCLKINOFF = 0, XTALOSCOFF = 0
CLKCTL |= 0x6000; // XCLKINOFF = 1, XTALOSCOFF = 1
CLKCTL &=~0x6000; // XCLKINOFF = 0, XTALOSCOFF = 0
CLKCTL |= 0x6000; // XCLKINOFF = 1, XTALOSCOFF = 1
```

一旦执行了上述程序，就可以配置OSC2选择开关。

如果连接了JTAG仿真器，并且GPIO38（TCK）正在切换，则无需执行上述程序，但亦无坏处。

如果GPIO38没有时钟，建议在GPIO38上为VDDIO增加一个强上拉电阻。

**26.勘误项**
**eQEP: 错过第一个索引事件**
**描述**

如果QEPI输入的第一个索引事件边沿在相应的QEPA/QEPB边沿之前的一个系统时钟周期到相应的QEPA/QEP边沿之后的两个系统时钟周期的任何时间发生，则eQEP模块可能会错过该索引事件，可能会导致以下行为：

- 如果QEPCTL[PCRM]=00b或10b (在索引事件上重置位置计数器或在第一个索引事件上重置位置计数器)，则QPOSCNT将不会在第一个索引事件上重置。
- 不会设置第一索引事件标记标志(QEPSTS[FIMF])。

**应变方法**

通过延迟索引信号以使QEPI事件边沿出现在相应的QEPA/QEPB信号边沿之后至少两个系统时钟周期来实现可靠的操作。对于编码器可能给予QEPI信号相对于相应的QEPA/QEPB信号的负延迟(TD)的情况(即，QEPI边沿出现在相应的QEPA/QEPB边沿之前)，QEPI信号延迟应大于“ $t_d + 2 * SYSCLKOUT$ ”。

**27.勘误项**
**eQEP: 在索引期间改变方向时位置计数器复位不正确**
**描述**

在使用PCRM=0配置时，如果在索引输入有效时发生方向改变，位置计数器(QPOSCNT)可能会错误地复位，从而导致计数器的值意外改变。这可能导致与位置计数器的预期值相差最多 $\pm 4$ 个计数，并导致随后意外设置错误标志。

使用PCRM=0配置 [即索引事件上的位置计数器复位(QEPCTL[PCRM]=00)] 时，如果索引事件在顺时针移动期间发生，则位置计数器在下一个eQEP时钟时复位为0。如果在逆时针移动期间发生索引事件，则位置计数器在下一个eQEP时钟时复位为QPOSMAX寄存器中的值。EQEP外设QEPI在QEPISTS寄存器中记录第一次索引标记(QEPSTS[FIMF])的出现和第一次索引事件标记(QEPSTS[FIDF])上的方向。它还会记住第一个索引标记上的正交边沿，以便将相同的正交边沿用于索引事件复位操作。

如果在索引脉冲有效时发生方向改变，则模块仍将继续寻找用于执行位置计数器复位的相对正交转换。这会导致位置计数器的值发生意外更改。

**应变方法**

不要使用PCRM=0配置，如果在索引处于活动状态时可能发生方向更改，并且由此产生的位置计数器值更改可能会影响应用。

用于执行位置计数器复位的其他选项 [如索引事件初始化(IEI)]，如果适用于应用程序，则不存在此问题。

28.勘误项

**eQEP: GPIO异步模式下的eQEP输入**

描述

如果通过GPxQSELn寄存器将任何eQEP输入引脚配置为GPIO异步输入模式，则eQEP模块可能无法正常工作。例如，QPOSCNT可能无法正确复位或锁存，并且可能会错过输入引脚上的脉冲。这是因为eQEP外设假定模块输入端存在与SYSCLKOUT的外部同步。

为使eQEP模块正常运行，输入GPIO引脚应通过GPxQSELn寄存器配置为同步输入模式(带或不带限定)。这是复位时GPxQSEL寄存器的默认状态。进芯电子提供的所有现有eQEP外设示例也将GPIO输入配置为同步输入模式。

eQEP模块输入引脚不应使用异步模式。

应变方法

将配置为eQEP引脚的GPIO输入配置为非异步模式(除“11b=异步”之外的任何GPxQSELn寄存器选项)。

29.勘误项

**ePWM: 如果Trip在屏蔽窗口结束时保持有效，则可能产生ePWM毛刺**

描述

屏蔽窗口通常用于屏蔽转换期间的任何PWM触发事件，这些事件对系统来说是错误的触发。在屏蔽窗口结束后，如果ePWM触发事件保持小于3个ePWM时钟周期，ePWM输出可能会产生不必要的毛刺。

图3说明了可能导致不必要的ePWM输出的时间段。

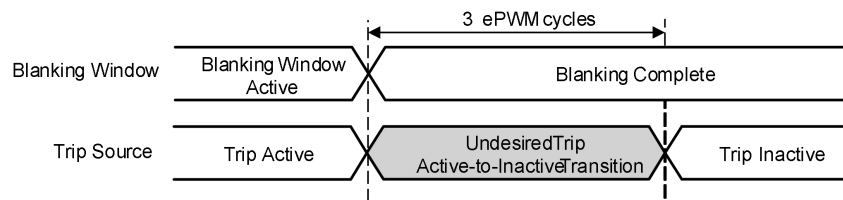


图3 不必要的触发事件和屏蔽窗口的结束

图4说明如果触发事件在屏蔽窗口关闭之前的1个周期或之后的3个周期内结束，则可能存在的两个潜在ePWM输出。

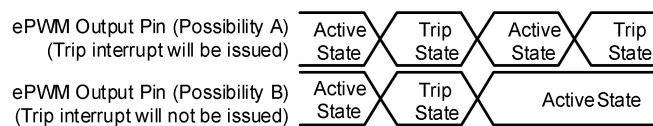


图4 可能产生不必要的ePWM输出

**应对方法** 为了避免不必要的触发行为必须增加或者减小屏蔽窗口。

**30.勘误项** **Boot ROM: Flash API [Flash\_Erase() Function]**

**描述** 器件引导ROM中的Flash API在Flash\_Erase()函数中包含一个编码错误。此问题不影响闪存擦除操作。擦除失败时，函数将失败并按预期返回错误状态，但未正确更新错误地址字段。

**应对方法** 无



## 联系方式

公司网址: [www.advancechip.com](http://www.advancechip.com)

联系邮箱: [sales@advancechip.com](mailto:sales@advancechip.com)

销售联系电话: **0731-88731027** (长沙)

公司总部地址: 长沙市湘江新区东方红街道北斗产业园·黄金园A5栋

南京销售中心: 南京市秦淮区卡子门大街**19**号紫云智慧广场**6**号楼**15**层

