

UM800Y 使用问题记录

版本：V1.1



广芯微电子（广州）股份有限公司

<http://www.unicmicro.com/>

版本修订

| 版本 | 日期 | 描述 |
|------|------------|-------------------|
| V1.0 | 2022.08.22 | 初始版 |
| V1.1 | 2022.08.24 | 删除 2 个问题记录及优化文档格式 |
| | | |
| | | |
| | | |

目录

| | | |
|------|-------------------------------------|----|
| 1 | 概述..... | 1 |
| 2 | PWM..... | 1 |
| 2.1 | 上电出现灯闪烁，或出现 PWM 无法正常关闭或输出..... | 1 |
| 2.2 | 旧版本 SDK P13 脚描述有误..... | 2 |
| 3 | Flash..... | 2 |
| 3.1 | 芯片在擦除模式下无法整长等待 Flash 标志位恢复..... | 2 |
| 4 | GPIO..... | 3 |
| 4.1 | IO 翻转异常..... | 3 |
| 4.2 | IO 配置和翻转异常..... | 3 |
| 5 | Timer0/1..... | 5 |
| 5.1 | Timer0/1 寄存器定时有误差..... | 5 |
| 6 | UART..... | 5 |
| 6.1 | 部分功能不运行/程序卡死..... | 5 |
| 6.2 | UART 接收异常，无法准确接收数据比特位..... | 5 |
| 7 | GPIO..... | 6 |
| 7.1 | 休眠功耗会有外界干扰变化..... | 6 |
| 8 | GTIMER..... | 7 |
| 8.1 | 控制死区时间输出互补 PWM 控制死区时间，无法控制..... | 7 |
| 9 | I2C..... | 7 |
| 9.1 | I2C 通信波形异常..... | 7 |
| 9.2 | I2C 输出信号异常或控制器件异常..... | 8 |
| 10 | ADC..... | 9 |
| 10.1 | ADC 采样不准，导致该通道采样数值/电压抖动..... | 9 |
| 11 | 芯片使用注意事项..... | 9 |
| 11.1 | LVD 低压检测功能..... | 9 |
| 11.2 | SPI 注意事项..... | 10 |
| 11.3 | UART2/UART3 使用单向半双工模式时，无法实现此功能..... | 10 |
| 11.4 | 芯片工作电压..... | 11 |

1 概述

本文档主要介绍 UM800Y 的使用问题记录，及芯片使用注意事项。

2 PWM

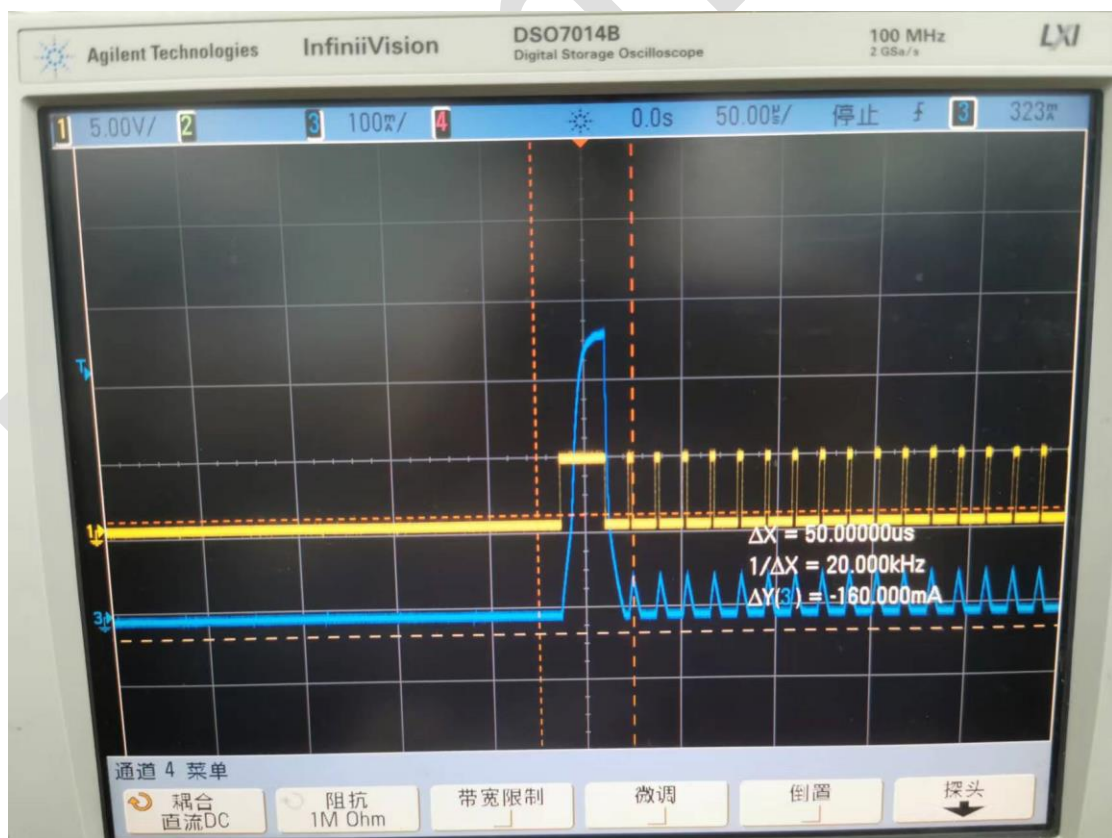
2.1 上电出现灯闪烁，或出现 PWM 无法正常关闭或输出

现象：上电出现灯闪烁，或出现 PWM 无法正常关闭或输出。

问题原因及分析：PWMxSS 和 PWMxCON 设置为 0，这两个寄存器是为了兼容上一代芯片的 PWM 输出，新的 8004 有多个 IO 复用模式，后续 8004 芯片只通过 PX_CFG 控制 PWM（此寄存器描述在 GPIO 章节）。

解决方法：后续通过 UM8004/UM800Y 芯片，通过 PX_CFG 寄存器控制输出/暂停 PWM 波形。在代码中需要控制 PXCFCG 在 delay1 个 PWM 周期再开启 PXCFCG 复用对应 PWM 模式。

下图是 PWM 输出时，瞬间高占空比异常波形图。



2.2 旧版本 SDK P13 脚描述有误

现象：P13 有 PWM 复用模式，用户手册旧版本写错成 CAN_Rx 模式

| C01BH | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|--------|---------|---|------|------|------|---------|------|------|
| P13CFG | - | - | - | - | - | P13_SEL | | |
| 读/写 | 读 | 读 | 读 | 读 | 读 | 读/写 | | |
| 复位值 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-3 | - | - | | | | | | |
| 2-0 | P13_SEL | 001: UART0_TX 010: UART2_RX 011: SPI_SCK 100: I2C_SDA 101: LPTIMER_IN 110: GTIMER0_CH 111: CAN_RX | | | | | | |

解决方法：代码和 SDK 手册更正后 P13_CFG=0x7;才是 PWM 输出模式，如下图所示：

| C01BH | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---------|---------|--|------|------|------|---------|------|------|
| P13_CFG | - | - | - | - | - | P13_SEL | | |
| 读/写 | 读 | 读 | 读 | 读 | 读 | 读/写 | | |
| 复位值 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-3 | - | - | | | | | | |
| 2-0 | P13_SEL | 000: P1_3 001: UART0_TX 010: UART2_RX 011: SPI_SCK 100: I2C_SDA 101: LP0_IN 110: GTIMER0_CH 111: PWM2 | | | | | | |

3 Flash

3.1 芯片在擦除模式下无法整长等待 Flash 标志位恢复

现象：Flash 擦除 ous 等待时间，导致芯片在擦除模式下无法整长等待 Flash 标志位恢复。

问题原因及分析：和芯片不同时钟下 ous 等待时间有关。

解决方法：16MHz 系统时钟是时钟芯片的默认 ous 等待值（已存在芯片内部）。当系统时钟使用 4MHz，设定 ous=4。注意：ous 不能随意更改。

4 GPIO

4.1 IO 翻转异常

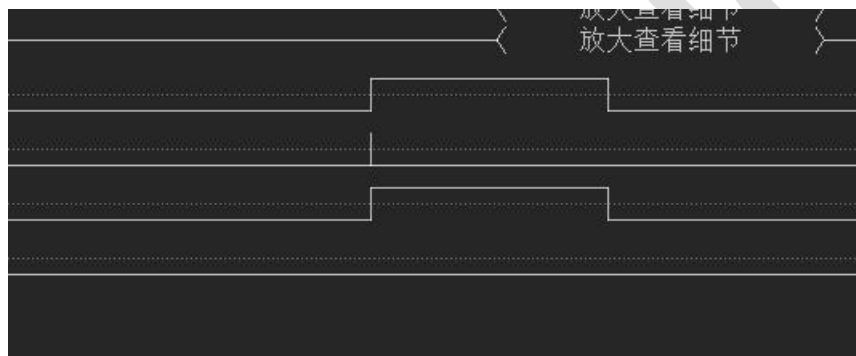
现象：IO 比特位方式设置输出连续操作翻转时，有部分 IO 翻转异常。

问题原因及分析：芯片的部分状态寄存器占用一个。

解决方法：对应 PIN 脚的 PXIE 寄存器需要设置为 1，因为输出高时，会位与这个比特位对应的 PXIE 寄存器再变成输出的电平。如果用 16 进制赋值的方式则不会。建议在 IO 使用过程中用 PXIE 寄存器保持置 1。

说明：当 IO 用位的方式在主循环连续操作翻转，例如：P11=1;P12=1; P13=1; 则可能会出现端口翻转异常。如果需要连续的 IO 翻转，则使用 16 进制的方式赋值 P1=0xe 可以实现。下图是出现异常翻转的 IO 状态。（这两个问题都是 IO 翻转需要读改写的原因导致。）

下图是部分 IO 异常翻转波形：



4.2 IO 配置和翻转异常

现象：IO 配置和翻转异常

问题原因及分析：旧版本 SDK 的 gpio demo code 里面，gpio.h 里面宏定义了 P00~P27 的值，这里会与系统库 um800y.h 的定义有冲突，编译可能不报错，但是在使用过程中可能会出错，导致原本想要的值被修改。

解决方法：除了 800y.h 函数的 P00~P27 不做修改。其余函数和 gpio.h 的宏定义都修改成 P0_0、P1_2、P2_7 这种格式避免重名冲突。

- 下图是 gpio.h

```

main.c  app.c  gpio.c  gpio.h
4  * Filename   : gpio.c
5  * Description : gpio driver header file
6  * Author(s)  : wanyi
7  * version    : V1.0
8  * Modify date : 2021-2-26
9  *****/
10 #ifndef _GPIO_H_
11 #define _GPIO_H_
12
13 #include "um800y.h"
14
15 #define P00      0
16 #define P01      1
17 #define P02      2
18 #define P03      3
19 #define P04      4
20 #define P05      5
21 #define P10      8
22 #define P11      9
23 #define P12     10
24 #define P13     11
25 #define P14     12
26 #define P15     13
27 #define P20     16
28 #define P22     18
29 #define P23     19
30 #define P25     21
31 #define P26     22
32 #define P27     23
33

```

- 下图是系统库:

```

15 Sbit (P05 , P0 , 5);
16 Sbit (P04 , P0 , 4);
17 Sbit (P03 , P0 , 3);
18 Sbit (P02 , P0 , 2);
19 Sbit (P01 , P0 , 1);
20 Sbit (P00 , P0 , 0);
21
22 Sfr (P1 , 0x90);
23 Sbit (P15 , P1 , 5);
24 Sbit (P14 , P1 , 4);
25 Sbit (P13 , P1 , 3);
26 Sbit (P12 , P1 , 2);
27 Sbit (P11 , P1 , 1);
28 Sbit (P10 , P1 , 0);
29
30 Sfr (P2 , 0xA0);
31 Sbit (P27 , P2 , 7);
32 Sbit (P26 , P2 , 6);
33 Sbit (P25 , P2 , 5);
34 Sbit (P23 , P2 , 3);
35 Sbit (P22 , P2 , 2);
36 Sbit (P20 , P2 , 0);
37

```

5 Timer0/1

5.1 Timer0/1 寄存器定时有误差

现象：Timer0/1 寄存器定时有误差。

解决方法：后续建议不使用（后续将屏蔽此功能），内部还有比较多的 gtimer/PWM/lptimer 可以替代。Lptimer\pwm\gtimer 可实现准确定时。

6 UART

6.1 部分功能不运行/程序卡死

现象：部分功能不运行/程序卡死。

问题原因及分析：当 UART 的 RX 端口没配置上拉时，RX 口又悬空，则可能会误触发容易进入 IO 的接收中断，影响程序测试功能。

解决方法：使用 UART 功能时，建议把 UART 的 RX 端口配置内部上拉。且 RX 配置应当不悬空。

下图为 GPIO 上拉配置寄存器：

5.1.86 PxPUN

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|------------|-------------------------|---------|--|---------|---------|---------|---------|---------|
| P0PUN(F1H) | - | - | - | P0PUN.4 | P0PUN.3 | P0PUN.2 | P0PUN.1 | P0PUN.0 |
| P1PUN(F2H) | - | - | P1PUN.5 | P1PUN.4 | P1PUN.3 | P1PUN.2 | P1PUN.1 | P1PUN.0 |
| P2PUN(F3H) | P2PUN.7 | P2PUN.6 | P2PUN.5 | - | P2PUN.3 | P2PUN.2 | - | P2PUN.0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 复位值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 位编号 | 位符号 | | 说明 | | | | | |
| 7-0 | PxPUN.y x=0-4, y=0-7 | | 端口上拉使能控制位（上拉电阻参考值 60KΩ）： 0：使能内部上拉电阻； 1：关闭内部上拉电阻。 | | | | | |

6.2 UART 接收异常，无法准确接收数据比特位

现象：UART 接收异常，无法准确接收数据比特位。

问题原因及分析：RX 管脚的 PXIE 位不能够置 0，这样会导致 RX 的电平时序无法传递到内部 UART 模块。

解决方法：如果不使用 RX 功能，可以通过 pxcfg 寄存器把 RX 端口改成其他复用模式。

7 GPIO

7.1 休眠功耗会有外界干扰变化

现象：休眠功耗会有外界干扰变化

问题原因及分析：io-在浮空状态是不稳定的，会出现电平高低变化影响芯片功耗，尽量不允许io出现浮空

解决方法：悬空未使用的IO建议配置一个固定电平上拉/下拉，或者通过PXIE来关闭输入通道。

下图是上拉/下拉配置寄存器。或者使用外部上拉、下拉或输入静止配置。

5.1.86 PxPUN

| | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|------------|-------------------------|--|---------|---------|---------|---------|---------|---------|
| P0PUN(F1H) | - | - | - | P0PUN.4 | P0PUN.3 | P0PUN.2 | P0PUN.1 | P0PUN.0 |
| P1PUN(F2H) | - | - | P1PUN.5 | P1PUN.4 | P1PUN.3 | P1PUN.2 | P1PUN.1 | P1PUN.0 |
| P2PUN(F3H) | P2PUN.7 | P2PUN.6 | P2PUN.5 | - | P2PUN.3 | P2PUN.2 | - | P2PUN.0 |
| 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 复位值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | PxPUN.y x=0-4, y=0-7 | 端口上拉使能控制位（上拉电阻参考值 60KΩ）： 0: 使能内部上拉电阻； 1: 关闭内部上拉电阻。 | | | | | | |

| D5H | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|------|--------|--------------------------------------|------|--------|--------|--------|--------|--------|
| P0PD | - | - | - | P0_4PD | P0_3PD | P0_2PD | P0_1PD | P0_0PD |
| 读/写 | 读 | 读 | 读 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 |
| 复位值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-5 | - | | | | | | | |
| 4 | P0_4PD | 端口 P0_4 下拉配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 3 | P0_3PD | 端口 P0_3 下拉配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 2 | P0_2PD | 端口 P0_2 下拉配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 1 | P0_1PD | 端口 P0_1 下拉配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 0 | P0_0PD | 端口 P0_0 下拉配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |

8 GTIMER

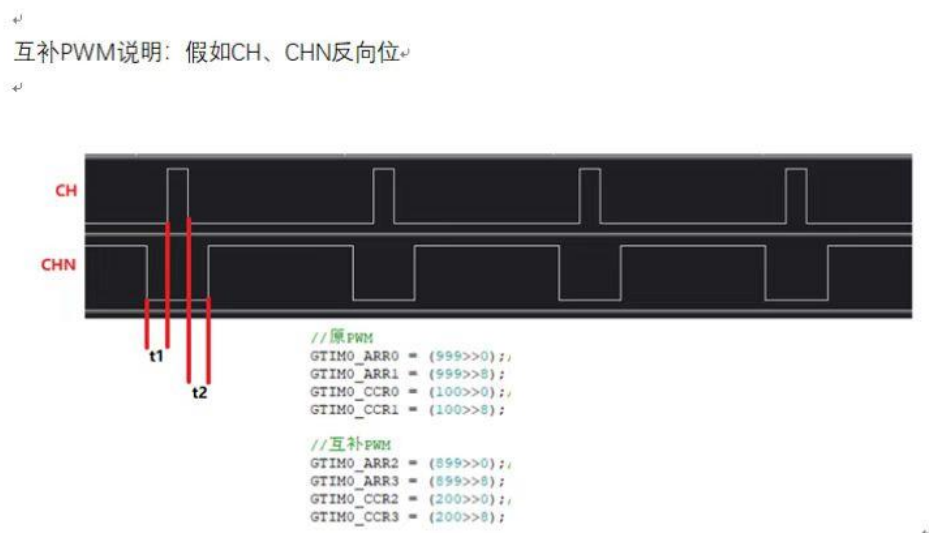
8.1 控制死区时间输出互补 PWM 控制死区时间，无法控制

现象：控制死区时间输出互补 PWM 控制死区时间，无法控制。

解决方法：可以通过寄存器配置实现死区功能使能。

| | | |
|---|----------|---|
| 1 | PWM_DEAD | PWM 死区插入功能使能： 0：避免死区功能关闭； 1：避免死区功能使能。 |
|---|----------|---|

或者需要调整死区时间用下图描述的方式实现死区功能使能；或通过死区间隔，设置间隔调整寄存器配置实现，效果如下：



CH、CHN的周期按周期大的PWM

原PWM设置周期=1000，互补PWM设置周期=900，最后互补PWM的周期会往左拉大

$t1 = \text{原PWM设置周期} - \text{互补PWM设置周期} = 1000 - 900 = 100\text{次计数时间}$

$t2 = \text{互补PWM设置占空比} - \text{原PWM设置占空比} = 200 - 100 = 100\text{次计数时间}$

9 I2C

9.1 I2C 通信波形异常

现象：I2C 通信波形异常

解决方法：使用硬件 I2C 时要先确认已配置为开漏模式，并且有外部上拉。开漏模式可以单独配置 IO 的模式去开启或 I2C 寄存器开启（只需配置其中一个寄存器即可）。

| CC03H | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---------|-------------|--|---------|------|--------------|------------|-------------|------------|
| I2C_CR1 | RFU | MTX_ANTO_EN | OD_MODE | RFU | MAAS2_INT_EN | WBT_INT_EN | RXNE_INT_EN | TXE_INT_EN |
| 读/写 | 读 | 读/写 | 读/写 | 读 | 读/写 | 读/写 | 读/写 | 读/写 |
| 复位值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | RFU | 保留。 | | | | | | |
| 6 | MTX_ANTO_EN | <ul style="list-style-type: none"> 当此模块 SLAVE 模式下，传输数据和地址 SDA 线自动切换。 当此模块 SLAVE 模式下，此位设置为 1，主设备传输完地址信息后，根据总线上的 RW 位，自动切换 SDA 传输方向。 0: 不使能自动切换功能； 1: 使能自动切换功能。 | | | | | | |
| 5 | OD_MODE | SCL 与 SDA 输出模式选择： 0: push-pull 模式输出； 1: open-drain 模式输出。 | | | | | | |
| 4 | RFU | 保留。 | | | | | | |

| ECH | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|------|--------|--|--------|------|--------|--------|------|--------|
| P2OD | P2_7OD | P2_6OD | P2_5OD | - | P2_3OD | P2_2OD | - | P2_0OD |
| 读/写 | 读/写 | 读/写 | 读/写 | 读 | 读/写 | 读/写 | 读 | 读/写 |
| 复位值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | P2_7OD | 端口 P2_7 开漏输出配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 6 | P2_6OD | 端口 P2_6 开漏输出配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 5 | P2_5OD | 端口 P2_5 开漏输出配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 4 | - | - | | | | | | |
| 3 | P2_3OD | 端口 P2_3 开漏输出配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 2 | P2_2OD | 端口 P2_2 开漏输出配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |
| 1 | - | - | | | | | | |
| 0 | P2_0OD | 端口 P2_0 开漏输出配置寄存器： 0: 禁止； 1: 使能。 | | | | | | |

9.2 I2C 输出信号异常或控制器件异常

现象：I2C 输出信号异常或控制器件异常

解决方法：配置 P04 和 P10 为 0 再设置当前 GPIO 为 I2C 模式。（为了兼容上一代 UM8005 问题。）配置及注意事项如下：

PCLK 和 PRESET 寄存器属于模块时钟和模块复位是否保持寄存器。如果操作了写 0，那么原来在模块 initial 的配置，寄存器将会变成默认值，需要重新配置。建议在一般情况下不操作写 0（休眠模式例外，但也需要唤醒之后第一个动作写回 1）。

15.5.5 作为从接收器

1. 向 I2C_SLAVE_ADDR1 寄存器或 I2C_SLAVE_ADDR2 寄存器写入 7 位地址作为自己在从机状态下被寻址的地址。
2. 写 I2C_CR 寄存器的 MEN 为 1，使能 I2C 模块。
3. 等待 MAAS1 或 MASS2 (ADDR2_EN=1) 标志是否有效。地址匹配无效则重复 3。
4. 地址匹配有效，判断 SRW 位是否为 1。为 0 表示从接收，为 1 表示从发送。
5. 等待字节传输完成标志 (MTF)。
6. 等到读取 RXNE 为 1 时，处理器需读取 I2C_DR 寄存器中接收到的字节，同时硬件会清除 RXNE 位。等待下一个字节的接收结束。
7. 等待 MTF 标志，并软件清除（不清软件流程也能继续）。
8. 重复 6-7 操作，接收到 stop 信号后停止。
9. I2C 也可以在一个字节接收完成，并发送完成 (即 MTF==1)。当写 I2C_CR 寄存器的 TACK 为 1 时，则会在下一个字节接收完成后发送 NACK 信号给主机。软件等待 MBB 为 0 后退出。

注意：I2C 默认的 IO 为 P04、P10，若要复用其他 IO 为 I2C 功能时，需要先将 P04、P10 配为 0，再复用其他 IO 才有效。

10 ADC

10.1 ADC 采样不准，导致该通道采样数值/电压抖动

现象：ADC 采样不准，导致该通道采样数值/电压抖动。

问题原因及分析：ADC 输入阻抗差异导致，高速采样时需要转移电荷实现采样，如果内部阻抗小时，等同于需要消耗转换的电荷实现稳定采样，这时需要外部可提供电荷/电容远大于内部充电电容，或采样速度降低时，才能实现高速有效 ADC 采样转换。

解决方法：建议外围电路采样电压不要等同于 VDD 电压，这样有外围电压抖动时，可能会损坏芯片且造成不可逆现象。使用外围分压再经过 ADC 采样时，建议并一个 1uF 电容在改 ADC 通道上，增加驱动能力使得 ADC 在高速采样时转换的电荷能够满足高精度 ADC 采样。

11 芯片使用注意事项

11.1 LVD 低压检测功能

LVD 低压检测功能，需要配置对应的档位开关，可以使能 LVD 中断/LVD 复位功能。LVD 检测点电压设置如下图所示：

| | | | |
|-----|-----------|--------------|-----------|
| 3-1 | LVDS[3:0] | LVD 检测点电压设置: | |
| | | LVDS | LVD point |
| | | 000 | 4.12V |
| | | 001 | 3.69V |
| | | 010 | 3.38V |
| | | 011 | 3.09V |

版本 V1.5

Copyright © 2022 广芯微电子（广州）股份有限公司

159

UM800Y 用户手册

LVD

| | | | |
|--|--|-----|-------|
| | | 100 | 2.85V |
| | | 101 | 2.65V |
| | | 110 | 2.48V |
| | | 111 | 2.32V |

11.2 SPI 注意事项

- SPI时钟源来自系统时钟，作为SPI_MISO信号的管脚输入必须使能，由Px_IE配置。
- 用作或者不用作SPI信号的管脚，由Pxx_CFG配置。
- 建议SCK_EN、MOSI_EN、MISO_EN、CS_EN设置为0，由Pxx_CFG配置为SPI管脚。
- PxOD对SPI管脚配置有效。

11.3 UART2/UART3 使用单向半双工模式时，无法实现此功能

在 layout 时要注意 TX 端口才能实现单向半双工模式，RX 管脚无法实现。

下图是寄存器配置使能 UART 单向半双工模式。

| CD02H/CE02H | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|-------------|--------|--|--------|---------|---------|-------|------|--------|
| UARTCR | RFU | TX_EN | TX_OEN | UART_LB | UART_P0 | FLUSH | TRS | ODD_EN |
| 读/写 | 读 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读/写 | 读 |
| 复位值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-5 | RFU | 保留 | | | | | | |
| 6 | TX_EN | Uart 单线模式使能控制： TX_EN =0, 不使能； TX_EN =1, 使能。 | | | | | | |
| 5 | TX_OEN | Uart 单线模式 TX 管脚数据传输方向控制： TX_OEN =0, TX 管脚作为数据输出脚； TX_OEN =1, TX 管脚作为数据输入脚。 | | | | | | |

11.4 芯片工作电压

芯片电源供电范围 VDD(2.5~5.5V)，且 IO 电压不能超过 VDD 的电压。IO 最大容忍电压值是 VDD+0.3V，且尽量避免这种情况。