

UM800Y 用户设计要点指南

版本：V1.0



UNICMICRO

广芯微电子

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	2022.03.07	初始版

目录

1	硬件设计相关	1
1.1	IO 管脚相关	1
1.2	NRST（外部复位管脚）相关	1
2	编译环境设置	2
3	软件设计相关	4
3.1	时钟设置	4
3.2	PWM 信号相关	4
3.3	Flash 操作相关	6
3.4	I/O 口操作相关	6

1 硬件设计相关

1.1 IO 管脚相关

1. VDDCORE 管脚必须并联 1 个电容到地，推荐 1uF（设计范围 0.47~2.2 uF）。
2. 避免 IO 脚的上电速度快于供电电压 VDD。
3. 避免 IO 脚的电压大于供电电压 VDD+0.3V。
4. 为提高抗干扰能力，未使用的 IO 脚不要悬空。可外部加上/下拉电阻，或者打开内部的上/下拉电阻。
5. 全部 IO 都可以配置为外部中断。外部中断可配置为上升沿中断、下降沿中断或者双边沿中断，全部 IO 的外部中断共用 INT0 中断入口。

1.2 NRST（外部复位管脚）相关

1. NRST 外部复位管脚，当被用作外部复位口时，此端口不能悬空，否则容易受到干扰，导致芯片复位。

常用的复位电路如图 1-1，复位性能 A>B>C>D。

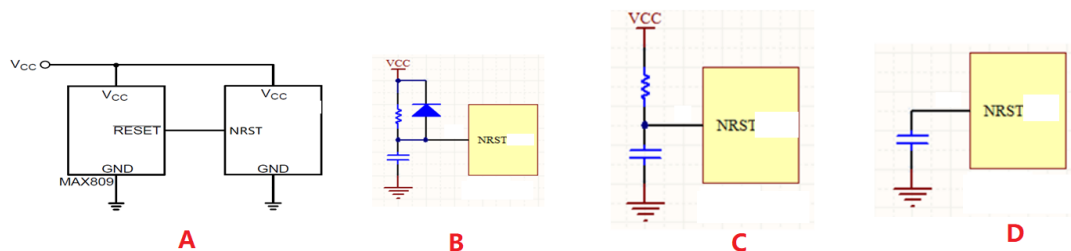


图 1-1：复位电路图

2. NRST 脚可被当作一般 IO 使用。上电默认 NRST 为外部复位口，直到程序配置为 IO 脚。UM800Y 中该管脚既可作为输入，也可作为输出。

UM800Y 的寄存器 ESTCR 中的第 3 位 ERSTEN 设置为 0 时，即把外部复位功能禁止，只作为 IO 脚使用。

2 编译环境设置

如图 2-1，在“Target”界面中，“Use multiple DPTR registers”不要勾选，因为双 DPTR 或者多 DPTR 各个厂家的实现可能不同，Keil c51 编译的方式不一定适配，导致运行异常。下面做个详细的解释：

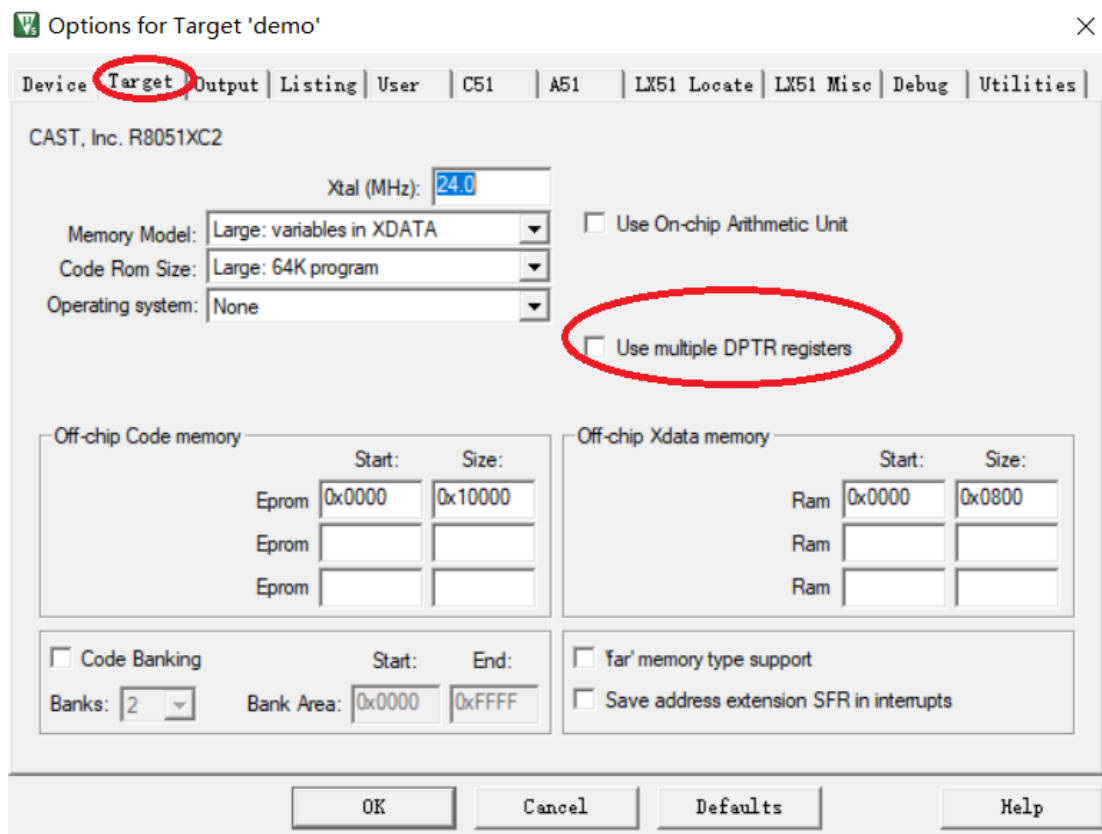


图 2-1: Target 配置界面

UM800Y 有双数据指针 DPTR0、DPTR1。

如果是汇编工程，从其它品牌转换到 UM800Y，并且有切换使用双数据指针的用法，逻辑算法可能需要修改，因为 UM800Y 的 DPTR0、DPTR1 物理上地址是不同的，而其它品牌的有可能物理地址一样。

- 假如 DPTR0、DPTR1 初始值都是 0000H，执行如下程序：

```
ANL DPS, #(~(1<<0)) ;数据指针选择 DPTR0
MOV DPL, #01H
MOV DPH, #00H
ORL DPS, #(1<<0) ;数据指针选择 DPTR1
MOV DPL, #02H
MOV DPH, #00H
```

此段程序原本目标为 DPTR0=0001H, DPTR1=0002H。

但实际如下：

如果物理地址一样，那么 DPTR0=0001H, DPTR1=0002H。

如果物理地址不一样，那么 DPTR0=0002H, DPTR1 保持初始值 0000H。

- 假如 DPTR0、DPTR1 初始值都是 0000H，执行如下程序：

```
ANL DPS, #(~(1<<0)) ;数据指针选择 DPTR0
```

```
INC DPTR
```

```
ORL DPS, #(1<<0) ;数据指针选择 DPTR1
```

```
INC DPTR
```

不管物理地址是否一样，那么 DPTR0、DPTR1 都是 0001H。

不管物理地址是否一样，把 DPL、DPH 组合起来当 DPTR 的使用（如下指令），如若切换到数据指针 0，那操作的就是 DPTR0，如若切换到数据指针 1，那操作的就是 DPTR1。

```
INC DPTR
```

```
MOV DPTR,#data16
```

```
MOVC A,@A+DPTR
```

```
MOVX A,@DPTR
```

```
MOVX @DPTR,A
```

```
JMP @A+DPTR
```

3 软件设计相关

3.1 时钟设置

1. 系统时钟可跑高速时钟或者内部低速时钟，由 CLKCON 的 BIT7: SYSCLKSEL 选择，高速时钟可选择内部高速时钟或者外部高速时钟，由 CLKCON 的 BIT0: HSCLKSEL 选择。
2. 系统时钟最高可跑 24M，当系统时钟 $\leq 16M$ 时，等待周期设置为 0；当系统时钟 $> 16M$ 时，等待周期设置为 1，等待周期由 OPSET 的 BIT5~3 的 RDWAIT 设置。
3. RCHDIV 只对内部高速时钟分频有效。
SYSDIV 对外部高速时钟、内部高速时钟、内部低速时钟 作为系统时钟时，分频都有效。

3.2 PWM 信号相关

1. LPTIMER（超低功耗定时器）产生 PWM 信号注意事项：
 - CMP=TARGET 时，占空比为 0%
 - CMP>TARGET 时，占空比为 100%
 - 其它占空比 $= (CMP+1)/(TARGET+1)$
2. GTIMER（通用定时器）产生 PWM 信号注意事项：
 - CH、CHN 周期设置为一样，同向位，占空比各自设置，当作 2 路 PWM 使用。
 - 如果使能了互补 PWM 输出，原 PWM 或者互补 PWM 的任意一路占空比设置为 0%的话，那么原 PWM 和互补 PWM 的占空比都会为 0%。
 - 如果需要一个 PWM 占空比为 0%，另一个 PWM 的占空比不为 0%时，软件可以这样处理：需要占空比为 0%的那路 PWM，占空比不要设置为 0%，通过 Pxx_CFG 配置为 IO 功能，再输出需要占空比 0%的电平。
3. 关于互补 PWM 的说明。
假如 CH、CHN 反向位。
 - CH、CHN 的周期按周期大的 PWM，如图 3-1：
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。
 $t1 = \text{原 PWM 设置周期} - \text{互补 PWM 设置周期} = 1000 - 900 = 100 \text{ 次计数时间}$
 $t2 = \text{互补 PWM 设置占空比} - \text{原 PWM 设置占空比} = 200 - 100 = 100 \text{ 次计数时间}$

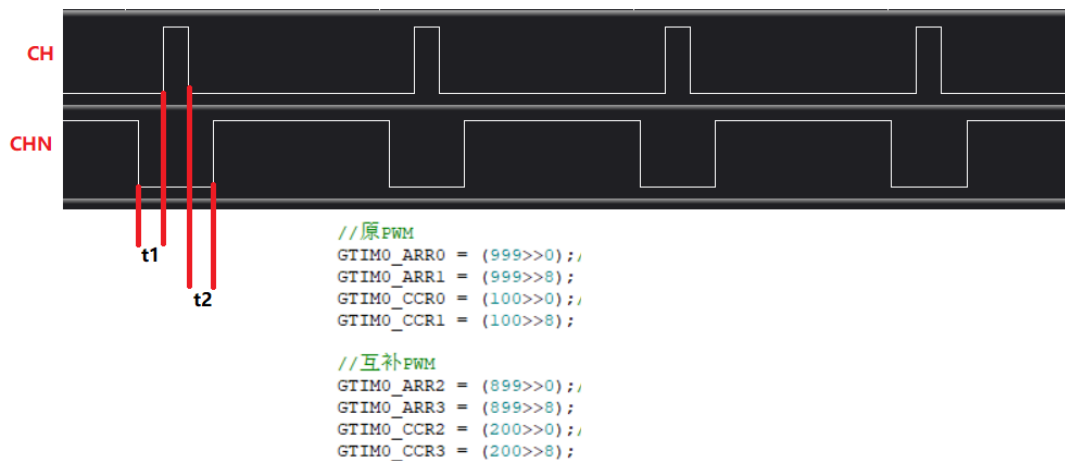


图 3-1: PWM 示意图

- CH、CHN 的周期按周期大的 PWM，如图 3-2:

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

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

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

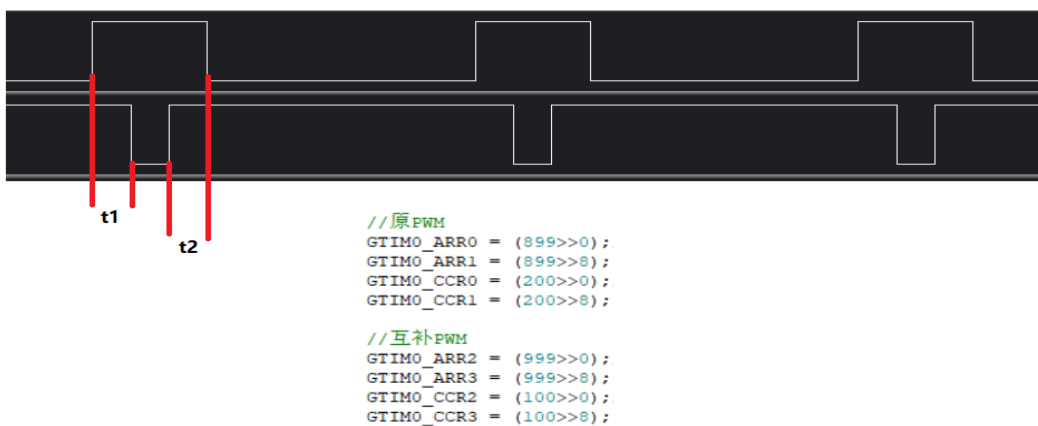


图 3-2: PWM 示意图

由图 3-1 及图 3-2 可看出:

- 谁的周期小，谁往左拉，最后拉至与周期大的一样，差值就是左边的死区时间。
- 谁的占空比大，谁往右，差值就是右边的死区时间。

4. PWM 设置生效时间。

如果停机前把占空比设置为 0，但停机后测试到 PWM 口有时为高电平，有时为低电平。

原因：修改 PWM 周期或者占空比，要下一个周期才生效。

解决办法：改了占空比为 0 到 STOP 指令前，确保时间 ≥ 1 个 PWM 周期的时间可解决；或者在 STOP 指令前，把端口配置成不作为 PWM 输出口，而是配置为一般 IO，再输出需要的电平也可解决。

3.3 Flash 操作相关

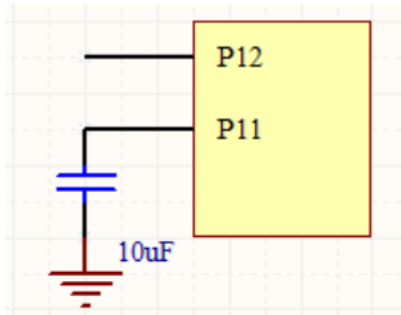
擦写 flash 时，先关闭总中断，操作完成后再打开。

3.4 I/O 口操作相关

1. 同组端口“读-改-写”问题。

操作同组端口，可能输出的高低电平并非如程序预期那样。

比如：



```
main()
{
    //P11、P12 配置为推挽输出

    P11=0;
    Delay();
    P11=1;
    P12=1;
    while(1);
}
```

最后 P12 输出高电平，P11 输出低电平。

原因：导致此问题的原因为操作同组端口时，有读改写的问题，由于电容的存在，执行 P11=1，实际 P11 不能立即为高电平，给电容充电到高电平需要一定时间，而此时立即操作同组端口，相当于读回来的 P11 还是低电平，所以执行 P12=1 时，把 0 又写到 P11 了。

解决办法：把同组端口的值先写到一个变量缓存，然后直接写到端口寄存器。

比如：

```
P1Value = 0x06;
P1= P1Value;
```

- 即使端口配置为输出，对应的端口输入使能寄存器 P0_IE/P1_IE/P2_IE 也配置为开启输入，此现象其实也是上面“读-改-写”的问题。