

## 免责声明

---



### Firmware Disclaimer Information

1. The customer hereby acknowledges and agrees that the program technical documentation, including the code, which is supplied by BEST HEALTH ELECTRONIC Inc., (hereinafter referred to as BestHealth) is the proprietary and confidential intellectual property of BestHealth, and is protected by copyright law and other intellectual property laws.
2. The customer hereby acknowledges and agrees that the program technical documentation, including the code, is confidential information belonging to BestHealth, and must not be disclosed to any third parties other than BestHealth and the customer.
3. The program technical documentation, including the code, is provided and for customer reference only. After delivery by BestHealth, the customer shall use the program technical documentation, including the code, at their own risk. BestHealth disclaims any expressed, implied or statutory warranties, including the warranties of merchantability, satisfactory quality and fitness for a particular purpose.

**Copyright (C) BEST HEALTH ELECTRONIC Inc.**  
**All rights reserved**

---

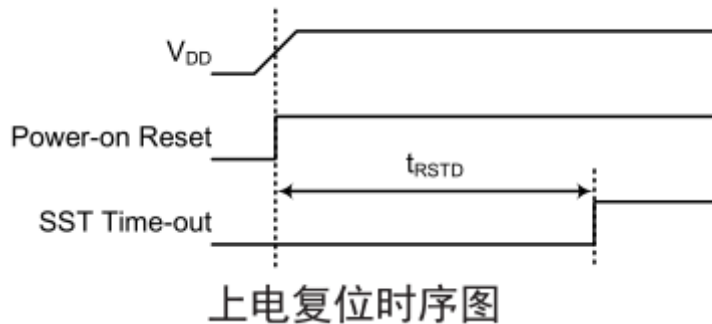
# 复位和初始化

单片机的几种内部复位方式将在此处做具体介绍。

相关复位时序中的 $t_{sst}$ 、 $t_{rstd}$ 等请查看 datasheet 中上电复位电气特性部分规格。

## 上电复位

这是最基本且不可避免的复位，发生在单片机上电后。除了保证程序存储器从开始地址执行，上电复位也使得其它寄存器被设定在预设条件。所有的输入、输出端口控制寄存器在上电复位时会保持高电平，以确保上电后所有引脚被设定为输入状态。

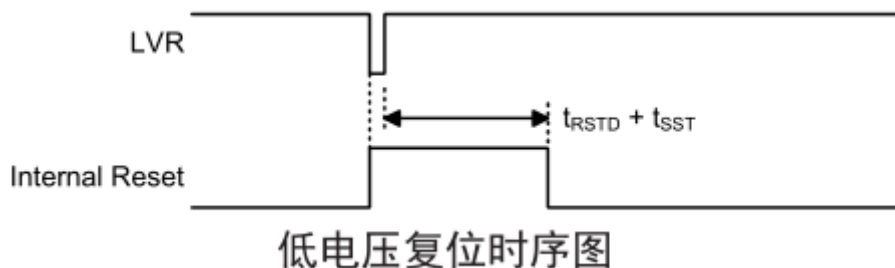


## 内部复位控制

内部复位控制寄存器 RSTC 用于为单片机在受到环境噪声干扰而异常工作或其他需求时提供复位。(本范例方式)

## 低电压复位

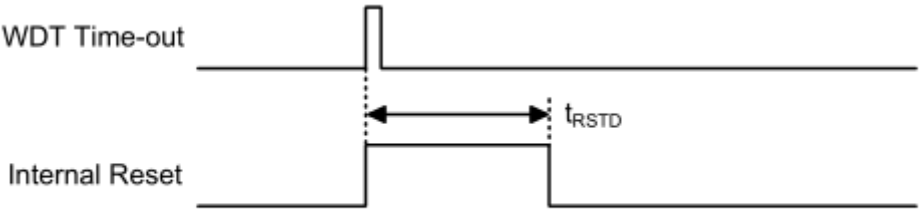
1. 单片机具有低电压复位电路，用来监测它的电源电压。LVR 始终使能，并会设定一个电源复位低电压，V<sub>LVR</sub>。例如在更换电池的情况下，单片机供应的电压可能会在 0.9V~V<sub>LVR</sub> 之间，这时 LVR 将会自动复位单片机且 RSTFC 寄存器中的 LVRF 标志位置位。
2. 另外通过对 LVRC 写入特定值也会是 MCU 产生复位操作。



## 看门狗溢出复位

1. 正常运行时看门狗溢出复位

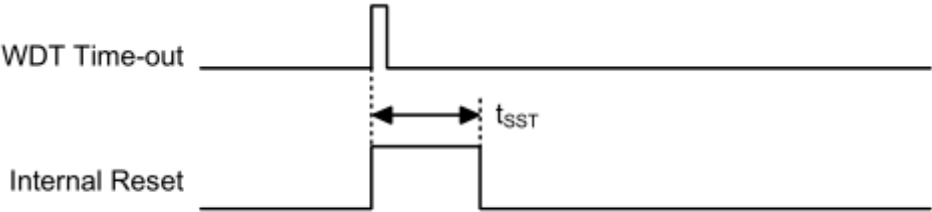
除了看门狗溢出标志位 TO 将被设为“1”之外，正常运行时看门狗溢出复位和硬件 LVR 复位相同。



正常运行时看门狗溢出复位时序图

2. 休眠或空闲时看门狗溢出复位

休眠或空闲时看门狗溢出复位和其它种类的复位有些不同。除了程序计数器与堆栈指针将被清“0”及TO 被设为“1”外，绝大部分的条件保持不变。图中  $t_{SST}$  的详细说明请参考系统上电时间电气特性。



休眠或空闲时看门狗溢出复位时序图

IAP复位

若 MCU有IAP功能，对IAP的FC1暂存器写0x55，也会产生复位信号使MCU复位。

# 功能說明

此范例演示了软件通过 RSTF 暂存器复位MCU的使用，调用 ResetMcu\_RstReg 后MCU将复位到main函数重新执行

1. 连接elink与开发板
2. 编译并下载程序
3. 执行 ResetMcu\_RstReg() 后，debug执行重新回到main

```

20 #include <stdint.h>
21 #include "..\..\driver\wdt.h"
22 #include "..\..\driver\Oscillators.h"
23 #include "..\..\driver\Reset.h"
24
25 void main()
26 {
27     if (IsDetectedFlag_RstRegResetMcu){
28         ClearFlag_RstRegResetMcu(); ① 此处上电不会执行，执行ResetMcu_RstReg后会执行
29     }
30
31     // 判断是否为上电复位或者非正常情况下的复位，如果是上电复位，执行上电复位初始化，反之执行WDT溢出初
32     if (_to == 0 || _pdf == 0) {
33         // config sys clock
34         Oscillators_Cfg();
35     } else {
36         // WDT溢出复位初始化
37         GCC_CLRWDT();
38     }
39     while (1) {
40         GCC_NOP();
41         GCC_NOP();
42         ResetMcu_RstReg(); ② 调用此函数复位Mcu
43     }
44 }

```