STM32F030 在低温下无法启动

关键字：低温，启动

1 问题描述

客户反馈 STM32F030 作为他们产品的控制芯片，在常温下工作是正常的，但是稍微冷冻下就会启动失败，重现率 100%，再次加热或者恢复到常温又能正常工作。此问题已经困扰了客户四五年，一直没有头绪，每次都更换一块芯片就好了，因为客户自己也知道，换芯片时会将其吹下来，必定会加热芯片，这样 MCU 也就能恢复正常了。但这种办法终究不是解决方法，客户急切想找到原因并解决问题。

2 分析问题

从客户描述上来看，猜测很大可能是硬件问题，因此带了一块 STM32F030-NUCLEO 板过去，想着做个芯片交换测试看下结果。

到达客户现场，了解到客户只是使用了内部高速晶振 HSI。先使用示波器抓下 VDD 和 NRST 的启动波形，在常温下发现并没有明显异常。于是做低温测试，为了对比，基于 STM32F030-NUCLEO 板了写了一个只使用内部高速晶振 HSI , 翻转一个 LED 灯的程序。结果显示，STM32F030-NUCLEO 板能正常启动，而客户的板子问题重现，再次测量其 VDD 和 NRST 的启动波形，发现 VDD 上电过程中有稍微不规则波形，但感觉不至于导致 MCU 无法启动。考虑到当前客户板子上的 MCU 跑的是客户自己的程序，为了统一对比，将客户板子上的 MCU 烧录成 NUCELO 板上一样的程序，再次做低温测试，结果显示客户的板子也能正常启动！

于是可以初步断定，此问题与客户自己的软件有关，而与外围电路无关。

接下来对比测试代码与客户自己的代码，并再次做低温测试验证结果，最终发现客户的时钟树配置有个参数有问题：

```
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_NONE;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
  RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL12;
  RCC_OscInitStruct.PLL.PREDIV = RCC_PREDIV_DIV1;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
```

如上红色代码所示，其 RCC_OSCILLATORTYPE_NONE 改成 RCC_OSCILLATORTYPE_HSI 后，问题现象明显改善，但经过测试，发现偶尔还会启动不正常的时候。但相对于之前 100%可以重现的现象，至少说明之前软件的失误至少是一个因素。

现在问题变成偶尔重现，已经向前迈进一大步。接下来怀疑与硬件有关了，理由是同样的测试软件跑在用户的板子上和跑在 NUCELO 软件上的结果不一致。因此接下来首先对于用户的板子的外围电路与 NUCELO-STM32F030 板子的外围电路，发现客户 MCU 的 BOOT0 引脚是悬空的，于是加上一个外部 10K 下拉电路，再次测试问题不再重现。

至此，问题解决！

3 后话

回过头来看这个问题，发现客户知道 MCU 使用的是 HSI，可偏偏在代码中配置时钟树时使用的晶振类型却是 NONE！这种问题现在看来是非常低级的问题，但在代码量大，或者代码迭代的过程中，之前写代码的人离职，后续接手的工程师又不能全盘了解所有代码的情况下时就会变成非常束手无策，当碰到此类莫名其妙的问题，特别是无法判断到底是硬件问题还是软件问题的时候，保持清晰的思路是非常重要的。这里我需要强调的是，最有效的解决方法就是快速找到一个 “参照物”，而 ST 的 DEMO 板和示例代码就是在硬件上和软件上扮演这样一个参照物的角色。可以通过 MCU 交换测试来判断是不是芯片外围电路的问题或者芯片问题，可以使用 Cube 库下的示例代码，对比其运行结果来判断是否与软件有关。先从大方向明确问题到底是与硬件有关还是与软件有关，然后再做下一步分析，这种方法希望读者能有效掌握。

版本历史

日期	版本	变更
2021 年 10 月 28 日	1.0	首版发布