# 目录

[目录 1](#_Toc855)

[1.计数器 1](#_Toc16834)

[2.设置断点 1](#_Toc15014)

[2.1程序断点： 2](#_Toc3260)

# 1.计数器

设置一个变量（或直接使用地址），通过对变量(地址)的内容取值加1，再存入变量(地址)中达到计数的目的。

fetch 1,0x4ff0

pincrease 1

store 1,0x4ff0

# 2.设置断点

断点本身是一种异常的中断，依靠修改中断寄存器达到让程序停止运行的效果。芯片断点分两种，一种是程序断点，一种是e命令断点。

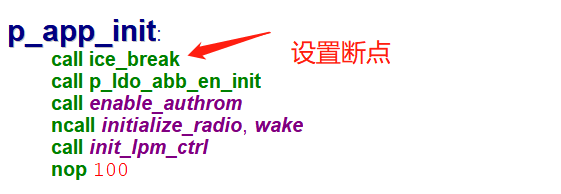
## 2.1程序断点：

1. 、在需要设置断点的地方调用 ice\_break 达到触发异常中断的效果。

格式：

*call ice\_break*

软件调用断点：



e命令查看断点：



2）、变量异常改写断点：**当某个指定变量被改写时**，程序会暂停在被改写位置。

程序运行中，可能存在变量被异常改写的情况，使用该断点，可以快速定位到，变量被异常改写的地方。

格式：

*arg mem\_ice\_break\_test //指定变量*

*call ice\_set\_write\_bp*

软件调用断点：



e命令查看断点：



**2.2 e命令断点**

e命令断点为程序运行时断点，即程序运行过程中，可以通过e命令设置断点。

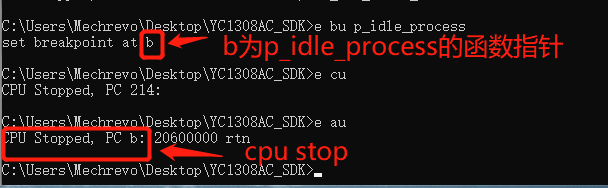
1）、e bu <address>

格式：

e bu <address>

*其中：*<address>为函数地址，当程序运行到该地址时，会暂停。

e命令操作断点：



1. 、e uu <address>

格式：

e uu <address>

*其中：*<address>为函数地址，当程序运行到该地址时，会暂停。

e uu 和e bu的区别：

使用e uu 时，e命令会在当前命名行等待，直到程序暂停为止。e bu不会让e命令等待。