

Keil调试时设置断点的高级用法

爱上电路设计 已于 2024-05-28 13:01:21 修改



单片机(汇编与C代码... 专栏收录该内容

95 订阅 88 篇文章

订阅专栏

在线调试程序时，打断点是非常有效的一种方式，配合单步调试，可以快速定位问题。但有的时候，手动打断点用起来不是那么方便。比如，想要在一个循环的第N次停下来，如果手动打断点，那就要不停的点击单步运行，直到循环运行到第N次。再比如，程序运行过程中变量不知何时被改变了，又不好定位，手动打断点就不知打在哪里。这时就要用到一些断点的高级用法。

目录：

- 1、设置断点的运行次数
- 2、变量匹配断点
- 3、变量条件匹配断点
- 4、打印信息
- 5、条件匹配打印信息

1、设置断点的运行次数

以下面程序为例，在循环的第十次将程序停止。首先在循环中手动打一个断点：

内容来源：[csdn.net](https://blog.csdn.net/liht1634)
作者昵称：爱上电路设计
原文链接：<https://liht1634.blog.csdn.net/article/details/125120980>
作者主页：<https://liht1634.blog.csdn.net>

```
112     /* Initialize all configured peripherals */
113     MX_GPIO_Init();
114     MX_DMA_Init();
115     MX_SDIO_SD_Init();
116     MX_USART1_UART_Init();
117     MX_FATFS_Init();
118     /* USER CODE BEGIN 2 */
119     for(uint8_t i = 0;i<100;i++)
120     {
121         AD++;
122         printf("AD = %d\r\n",AD);
123     }
124     /* USER CODE END 2 */
125
126
127
128
```

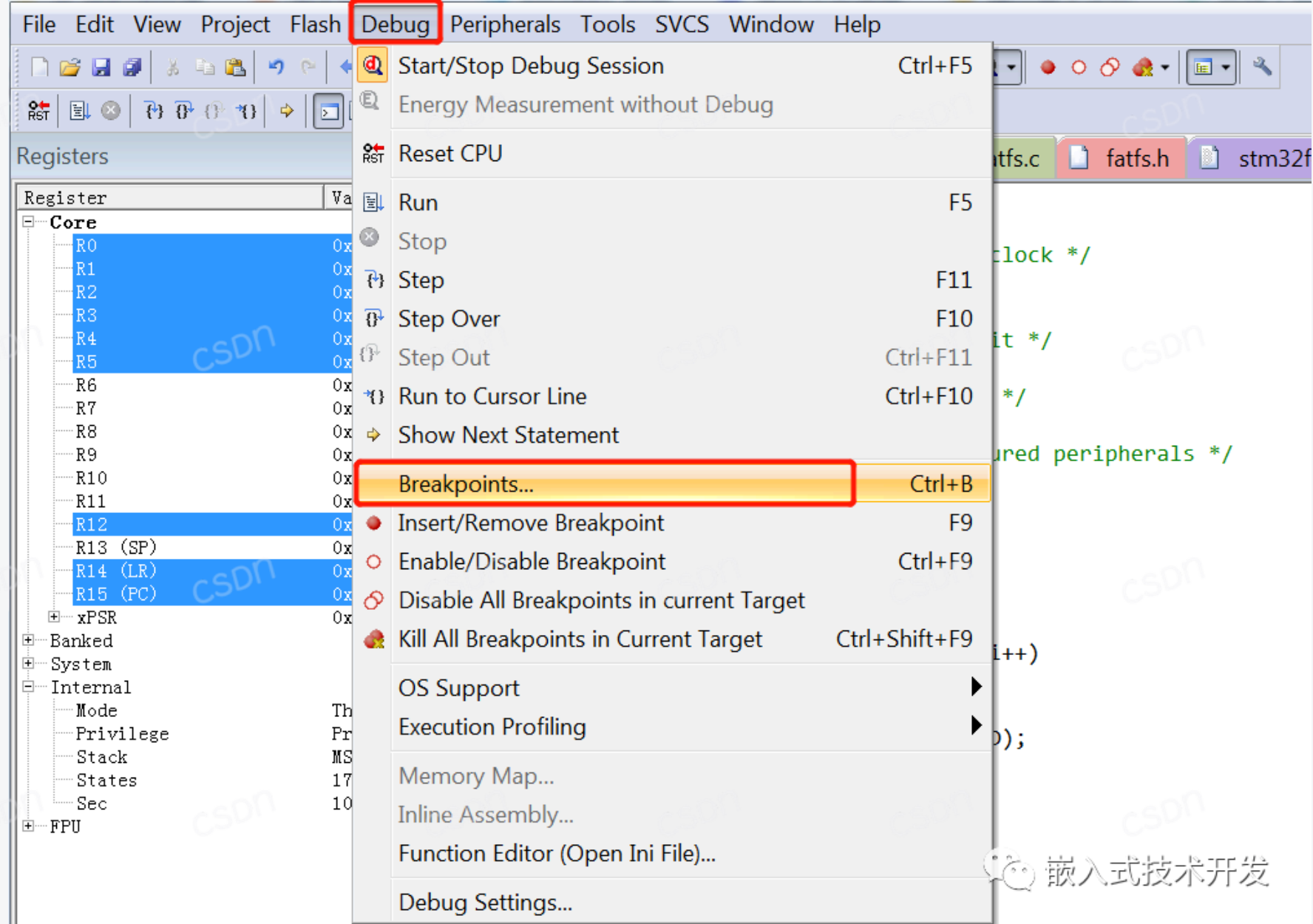
点击Debug菜单，选择BreakPoints：

内容来源：csdn.net

作者昵称：爱上电路设计

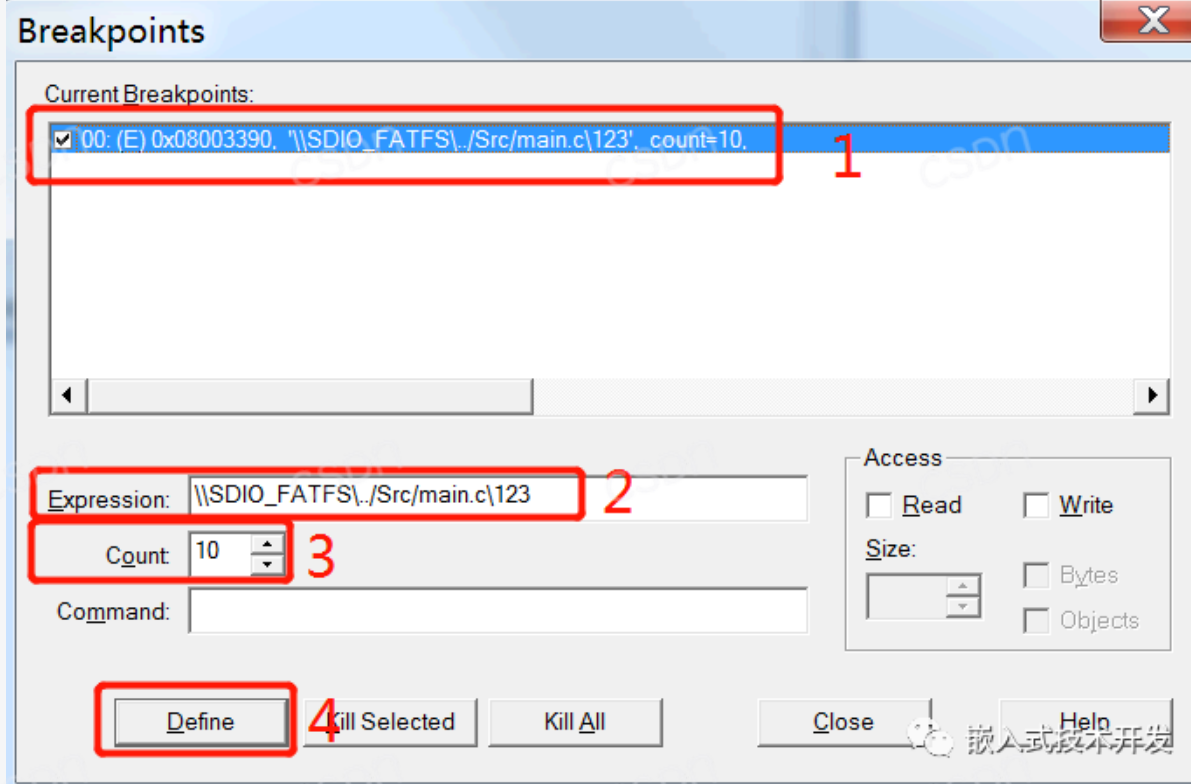
原文链接：https://liht1634.blog.csdn.net/article/details/125120980

作者主页：https://liht1634.blog.csdn.net



弹出下面菜单:

内容来源: csdn.net
作者昵称: 爱上电路设计
原文链接: <https://liht1634.blog.csdn.net/article/details/125120980>
作者主页: <https://liht1634.blog.csdn.net>



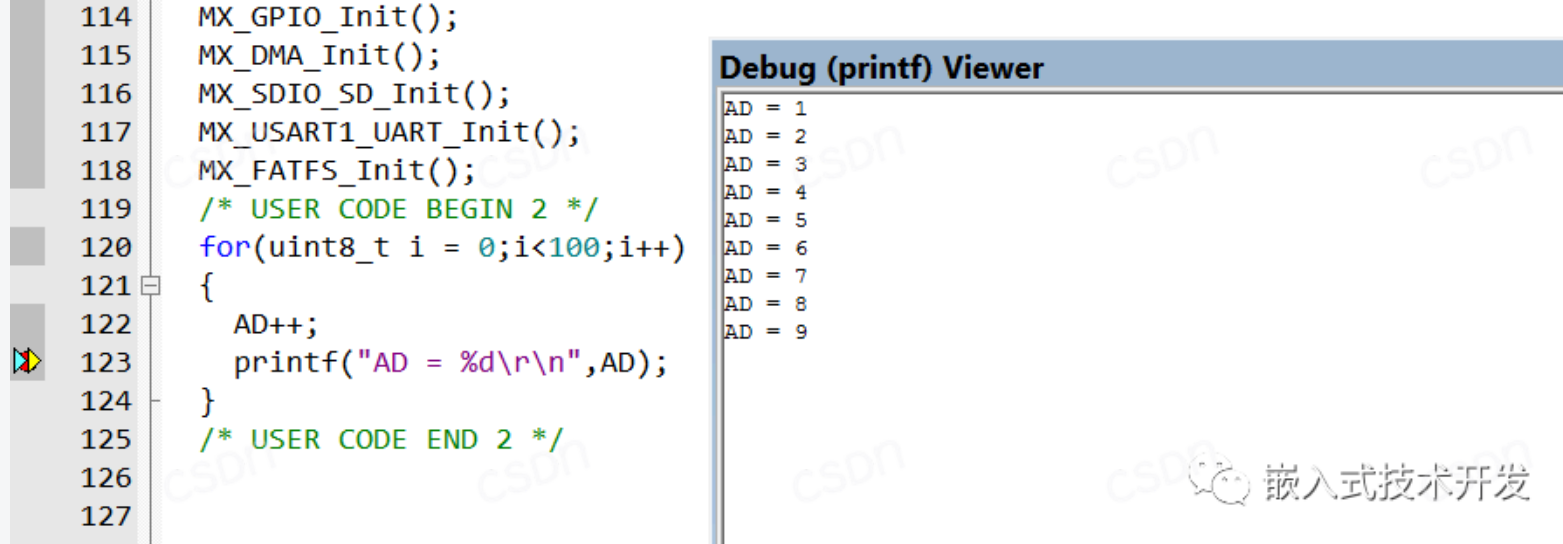
最上面的断点是手动打的，双击该断点，会看到Expression会显示该断点的信息，修改Count的值为10，点击Define，然后关闭该窗口。这里说明一下：

Expression为表达式，即断点的条件，可以看到，手动设置的断点结尾为\\123,表示在main.c文件的123行。这里支持基本的>、<、==、!=等操作符。

Count为次数，表示运行多少次中断一次，手动设置的断点Count都是1。

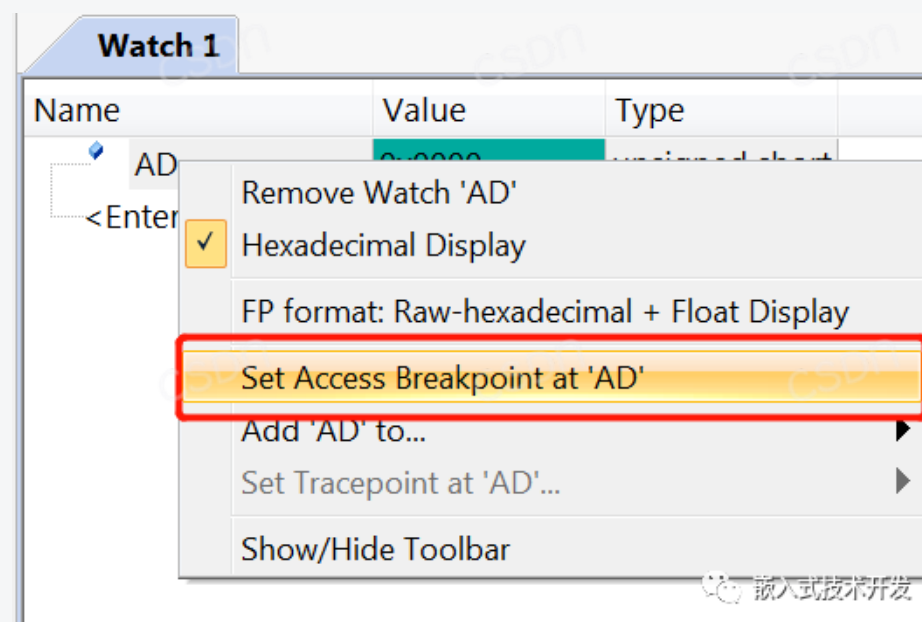
Command为命令，表示到达该断点时执行的命令，默认为空。

按照上面的设置，运行程序。可以看到，程序运行到断点时没有马上停下，而是在第10次才停下来：

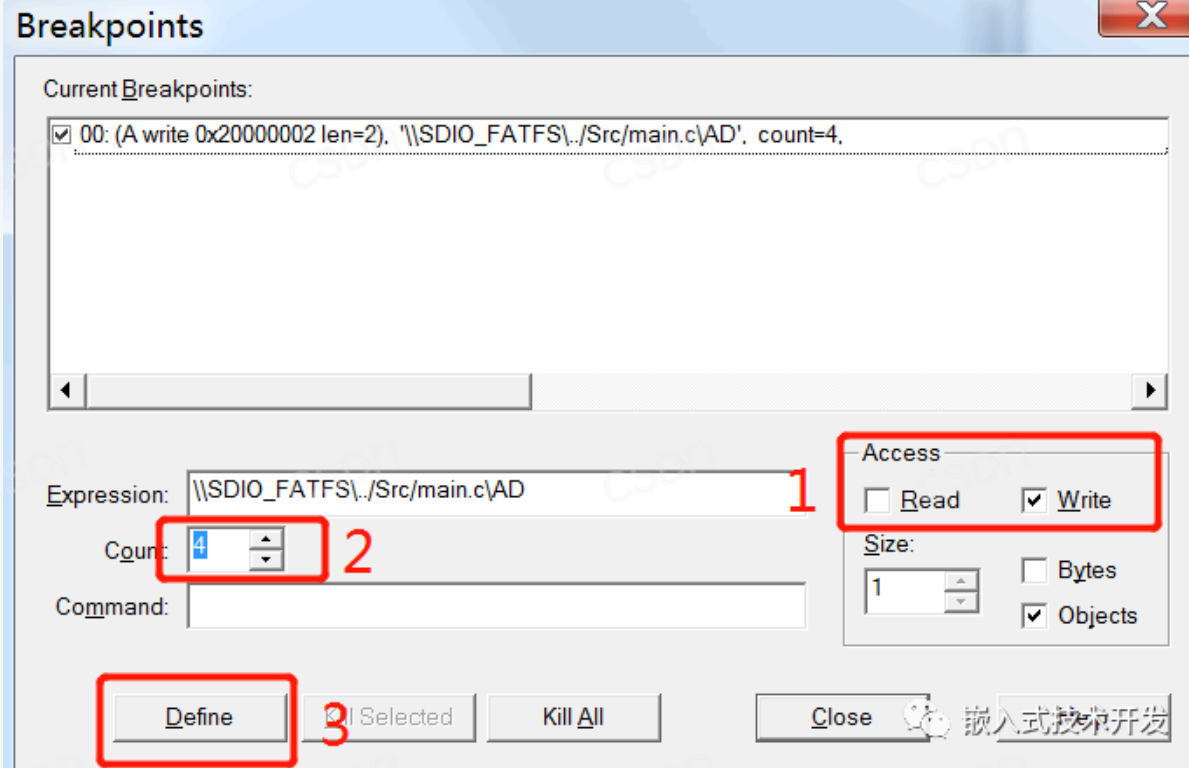


2、变量匹配断点

将变量添加到Watch窗口，右击选择Set Access BreakPoint at xxx。还是弹出刚才的菜单：



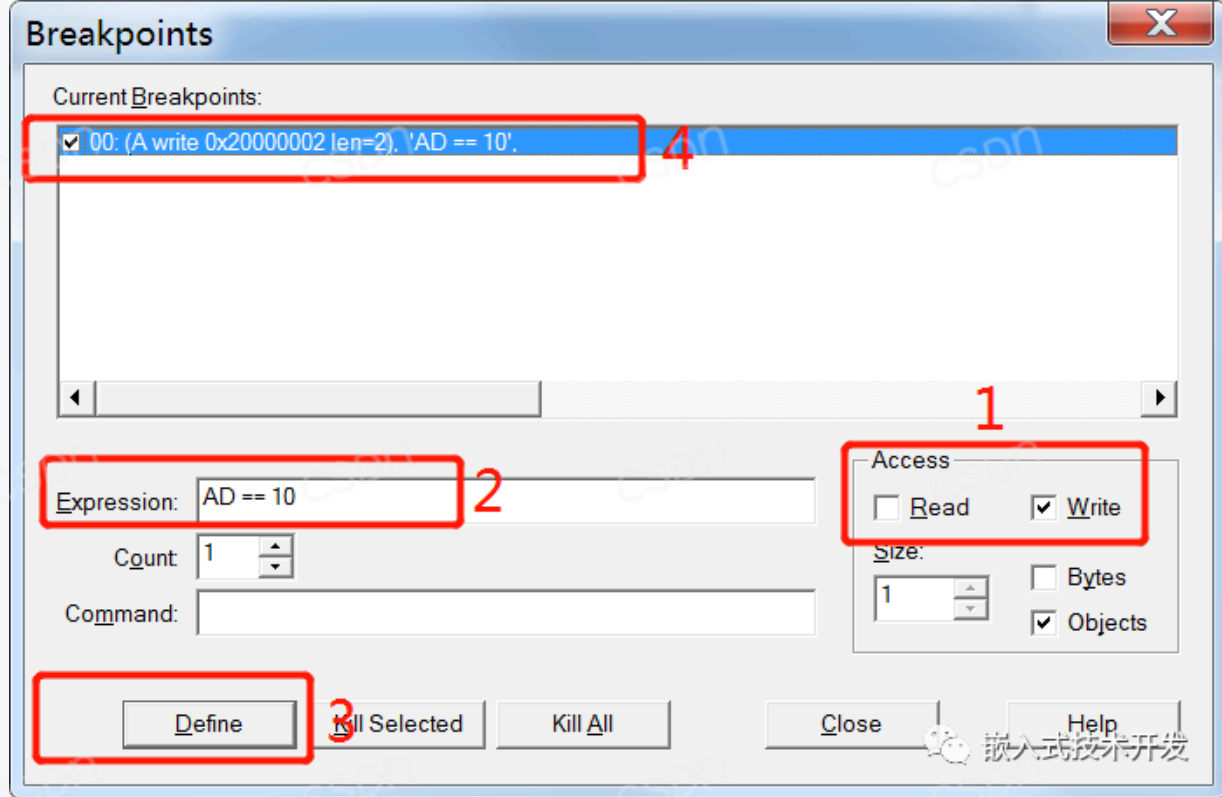
内容来源: csdn.net
作者昵称: 爱上电路设计
原文链接: <https://liht1634.blog.csdn.net/article/details/125120980>
作者主页: <https://liht1634.blog.csdn.net>



勾选Access方式Read或Write，设置Count值，点击Define。这里选择Write，Count值为4，表示该变量第四次被写入时程序会停止。

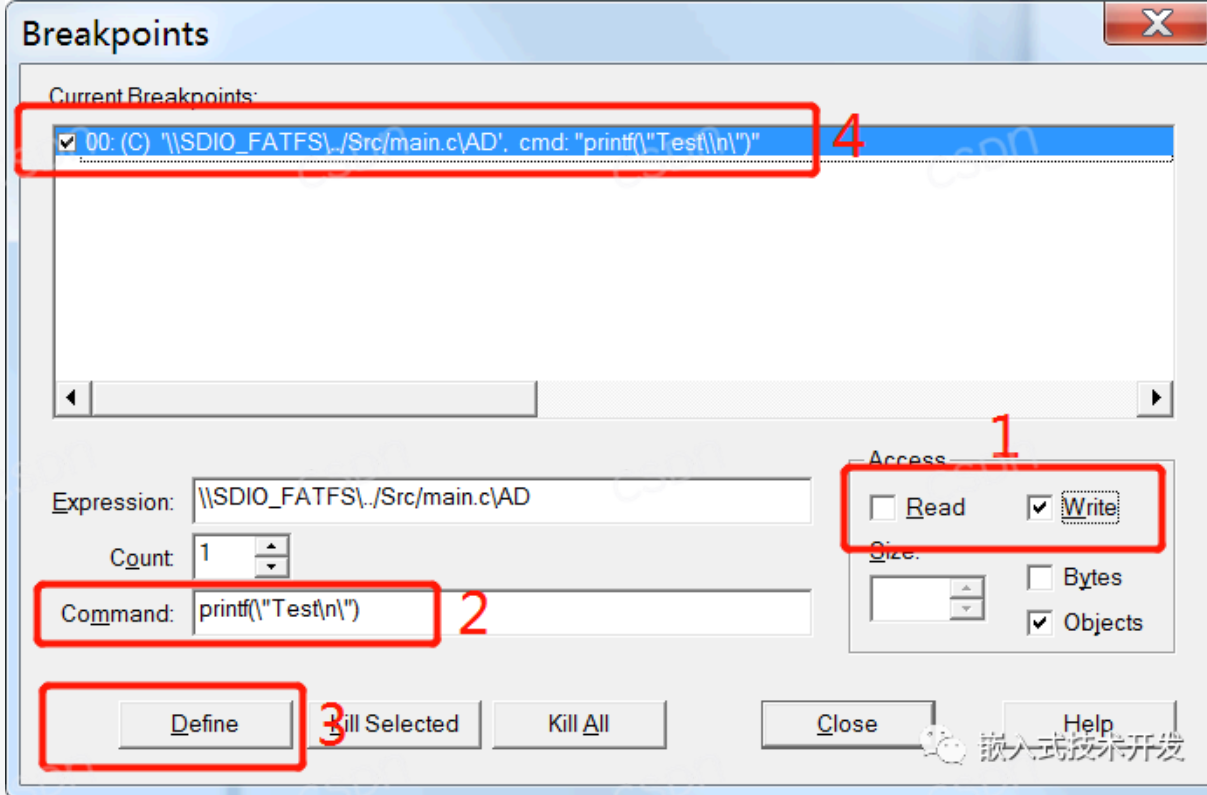
3、变量条件匹配断点

跟上一步一样，在Watch窗口，右击变量选择Set Access BreakPoint at xxx。勾选Access方式Read或Write，删除Expression下原来的内容，填写表达式“AD== 10”。点击Define。这样当AD==10时程序会停止。



4、打印信息

选择一个手动打好的断点，然后在Command下填写printf()，则在运行到断点处时，程序不会停止，而是在Command窗口打印设置的信息。

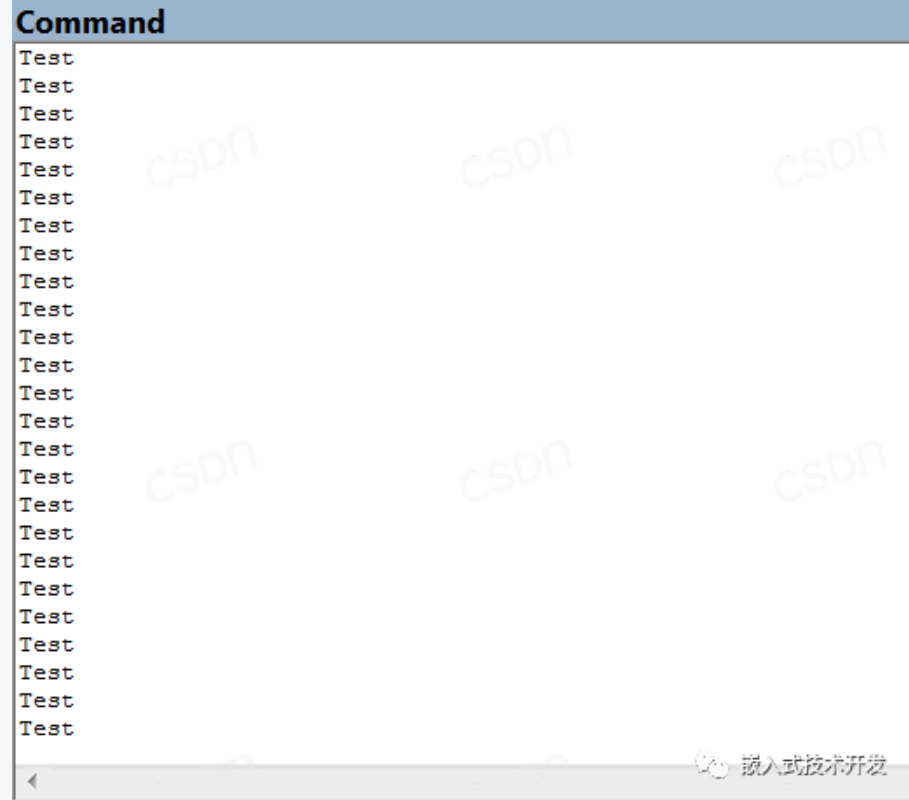


内容来源: csdn.net

作者昵称: 爱上电路设计

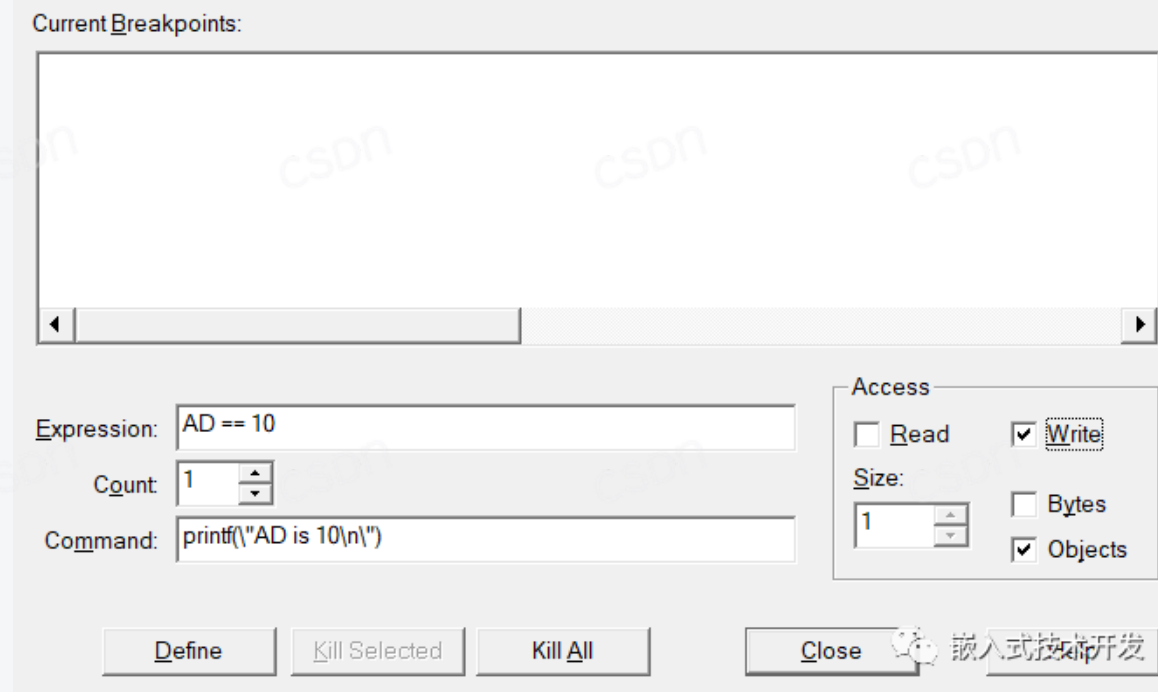
原文链接: <https://liht1634.blog.csdn.net/article/details/125120980>

作者主页: <https://liht1634.blog.csdn.net>



5、条件匹配打印信息

还可以将3和4组合使用，如下：



则AD==10时，程序不中断，在Command窗口打印信息。

断点的高级用法还有很多种，不同的表达式，不同的指令等，具体可查阅Keil的使用帮助文档。

ARM Development Tools

隐藏

查找

上一步

前进

打印

选项(Q)

目录(C)索引(N)搜索(S)

键入关键字进行查找(W):

BreakSet

ASM
ASSIGN
_atan uv4 Library Routine
BA
batch build
BD
BE
BK
BL
bookmarks
BreakAccess
BreakDisable
BreakEnable
BreakKill
BreakList
Breakpoint
BreakSet
BS
build multiple projects
build project
CID
Code Coverage
configuration
configure targets

显示(D)

BreakSet

The **BreakSet** command sets a breakpoint for the specified expression (*exp*). Breakpoints are program addresses or expressions that, when true, halt execution of your target program or execute the specified command ("*cmd*").

Syntax	Description
BreakSet <i>exp</i> , <i>cnt</i> , " <i>cmd</i> "	Set an execution or conditional breakpoint.
BreakSet READ <i>exp</i> , <i>cnt</i> , " <i>cmd</i> "	Set a read access breakpoint.
BreakSet WRITE <i>exp</i> , <i>cnt</i> , " <i>cmd</i> "	Set a write access breakpoint.
BreakSet READWRITE <i>exp</i> , <i>cnt</i> , " <i>cmd</i> "	Set a read/write access breakpoint.

Breakpoint parameters:

where:	
<i>exp</i>	is an address specification or an expression that is evaluated during execution. Refer to Expressions for details.
<i>cnt</i>	is an expression that determines the number of times a breakpoint condition is met before the target program halts or the specified command is executed. The default count value is 1. After the first breakpoint trigger, count is disregarded.
<i>cmd</i>	is a µVision command string that is executed when the breakpoint triggers. Program execution is not halted. User-defined and signal functions can be used in the command expression. In functions, you can set the system variable _break to 1 to halt program execution. When no command is specified, program execution is halted.

Breakpoint types are categorized through the following rules:

- Access Breakpoint:** one of the memory access modes (READ, WRITE, or READWRITE) was specified.

Memory Access Breakpoints halt program execution or execute a command when a specific address is read, written, or accessed. Program execution speed is not affected dramatically, because expressions are

当时年少春衫薄。骑马倚斜桥，满楼红袖招。觉得不错，动动发财的小手点个赞哦！

内容来源: csdn.net
作者昵称: 爱上电路设计
原文链接: <https://liht1634.blog.csdn.net/article/details/125120980>
作者主页: <https://liht1634.blog.csdn.net>



问题咨询、技术交流、项目合作



微信名片



内容来源: [csdn.net](https://blog.csdn.net/liht_1634)

作者昵称: 爱上电路设计

原文链接: <https://liht1634.blog.csdn.net/article/details/125120980>

作者主页: <https://liht1634.blog.csdn.net>