

# 如何固定一个全局变量的VMA 地址或者一个函数的VMA 地址？

之前和一个客户讨论一个技术问题，这个客户之前用ARM 的Core 和用KEIL IDE，他说KEIL 所带的编译器支持一个\_at\_ 扩展语法，可以很简单的就把一个全局变量的VMA 指定到一个地址上，问如果切换到Nuclei Core 和Nuclei Studio IDE 这边，要如何做到？

首先要说明的是：CARM's \_\_at\_\_ 以及 armcc's \_\_attribute\_\_((at)) GNU GCC [c - What is the equivalent of Keil \\_\\_at\\_\\_ in gcc - Stack Overflow](#)ARM: Locating Variables at Absolute Memory Addresses

VMA

```
/* General Purpose Input/Output (GPIO) */
#define IOPIN0      (*((volatile unsigned long *) 0xE0028000))
.
.
.
IOPIN0 = 0x4;
```

section

A C \_\_attribute\_\_ section

```
int __attribute__((section(".myfuncsect"))) test_self_section_func (int a, int b)
{
    int c;
    c = a+b;
    printf(" my section function!\r\n");
    return c;
}

int test_self_section_data [4] __attribute__((section(".mydatasect")));
```

B section VMA

```
.myfuncsect      :
{

    *(.myfuncsect)
    *(.myfuncsect.*)
} >ilm1 AT>ilm1
```

```
.mydatasect      :
{

    *(.mydatasect)
    *(.mydatasect.*)
} >ram1 AT>ram1
```



TestSelfSection.zip

Nuclei IDE

```
237 80000730 g F .text 00000040 __write
238 80002518 g F .text 0000004c __swrite
239 80001e14 g F .text 00000024 fprintf
240 8000222e g F .text 0000028c vfprintf_r
241 80001e38 g F .text 0000006a _fwalk_reent
242 800002e2 w F .text 00000024 _kill
243 900003dc g O .rodata 00000020 __sf_fake_stdout
244 80000c42 g F .text 0000048c print_misa
245 80000af6 g F .text 0000005a SystemBannerPrint
246 80000344 w F .text 00000084 read
247 90100000 g O .mydatasect 00000010 test self section data
248 80000260 w F .text 0000001a _exit
249 80001efe g F .text 0000009e _smakebuf_r
250 80100000 g F .myfuncsect 00000038 test self section func
251 800002ce g F .text 0000026a _printf_1
252 90000510 g O .bss 00000004 __malloc_sbrk_start
253 9000050c g O .bss 00000004 __malloc_free_list
254 90000490 g .rodata 00000000 __fsymtab_start
255 8000222e g F .text 0000028c vfprintf_r
256 800005f2 g F .text 000000be gpio_iof_config
257 80002acc g F .text 00000008 raise
258 80000240 w F .text 00000020 _close
259
260
261
262 Disassembly of section .init:
```

