

STM32CubeIDE 实用技巧之配置 Heap 空间

关键字: STM32CubeIDE, Heap, 堆, __sbrk

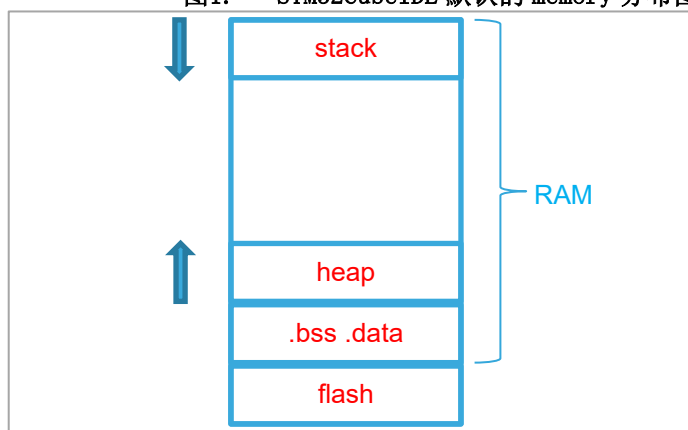
1. 问题描述

近日接到一客户需求, 其欲在 STM32CubeIDE 中指定堆(Heap)的地址到某一专用的 RAM 中。

2. 问题分析

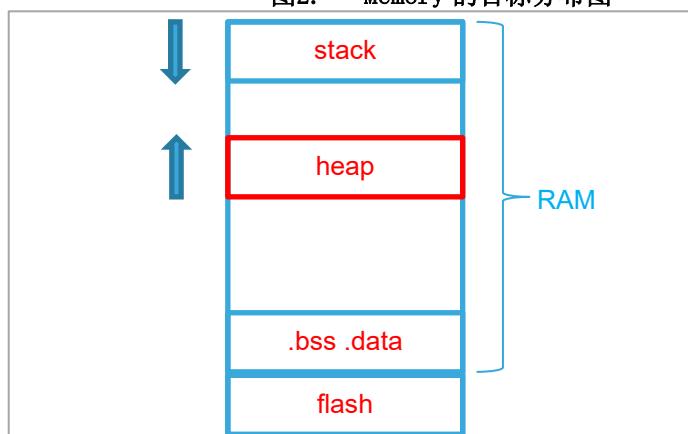
STM32CubeIDE 生成的工程中, 默认的堆栈配置示意图如图 1 所示:

图1. STM32CubeIDE 默认的 memory 分布图



客户的需求就是想配置堆栈为图 2 的分布状态, 这里堆的地址要能够自定义:

图2. Memory 的目标分布图



3. 解决方法

通过 STM32CubeIDE 生成的工程中，会默认生成 `system.c`。这个文件中的 `__sbrk_heap_end` 指针默认赋值为 `NULL`，我们只需要更改这个指针为目标地址即可达到期望。

例如：默认配置是 “`static uint8_t * __sbrk_heap_end = NULL;`” 我们希望将堆的地址分配到 `0x20001000` 为起始的空间上，可以这样做：

```
“static uint8_t * __sbrk_heap_end = (uint8_t*)0x20001000;”
```

注意：由于堆（`heap`）向上增长，栈（`stack`）向下增长，所以在手动更改堆的起始地址时，如果定义后的堆和栈内存存在重叠区域，STM32CubeIDE 无法识别这种情形，用户应当避免这样的设置。

4. 验证

在 STM32CubeIDE 生成的串口打印的工程，然后按照上面示例修改 `system.c` 中的 `__sbrk_heap_end` 参数。

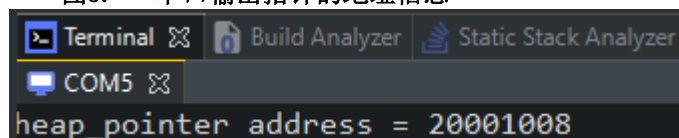
```
int main(void)
{
    /* USER CODE BEGIN 1 */
    uint8_t *heap_pointer = (uint8_t *)malloc(10);

    /* USER CODE END 1 */
    . . .

    /* USER CODE BEGIN 2 */
    if(heap_pointer != NULL)
    {
        printf("heap_pointer address = %x\r\n", heap_pointer);
    }else
    {
        printf("malloc memory failed\r\n");
    }

    /* USER CODE END 2 */
    . . .
}
```

图3. 串口输出指针的地址信息



可以看到 `heap_pointer` 指针通过 `malloc` 申请到的地址，已经指向了目标 RAM 范围内了。

注意，这里我们在 `0x20001000` 开始的位置申请了 10 个字节的空间，理论上 `heap_pointer` 应当从 `0x20001000` 开始计算，但是由于 `malloc` 在申请地址时，会占用 8 个字节用来存储状态等信息，所以 `heap_pointer` 指针从 `0x20001008` 开始。这是 STM32CubeIDE 使用了 `newlib` 库来实现 `malloc` 的机制所致，不影响用户的使用。

文档中所用到的工具及版本

STM32CubeIDE Version 1.7.0

版本历史

日期	版本	变更
2021年12月10日	1.0	首版发布

重要通知 - 请仔细阅读

意法半导体公司及其子公司 (“ST”) 保留随时对 ST 产品和 / 或本文档进行变更的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章，旨在交流与分享，并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致，请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档（包括代码，电路图等信息），我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。