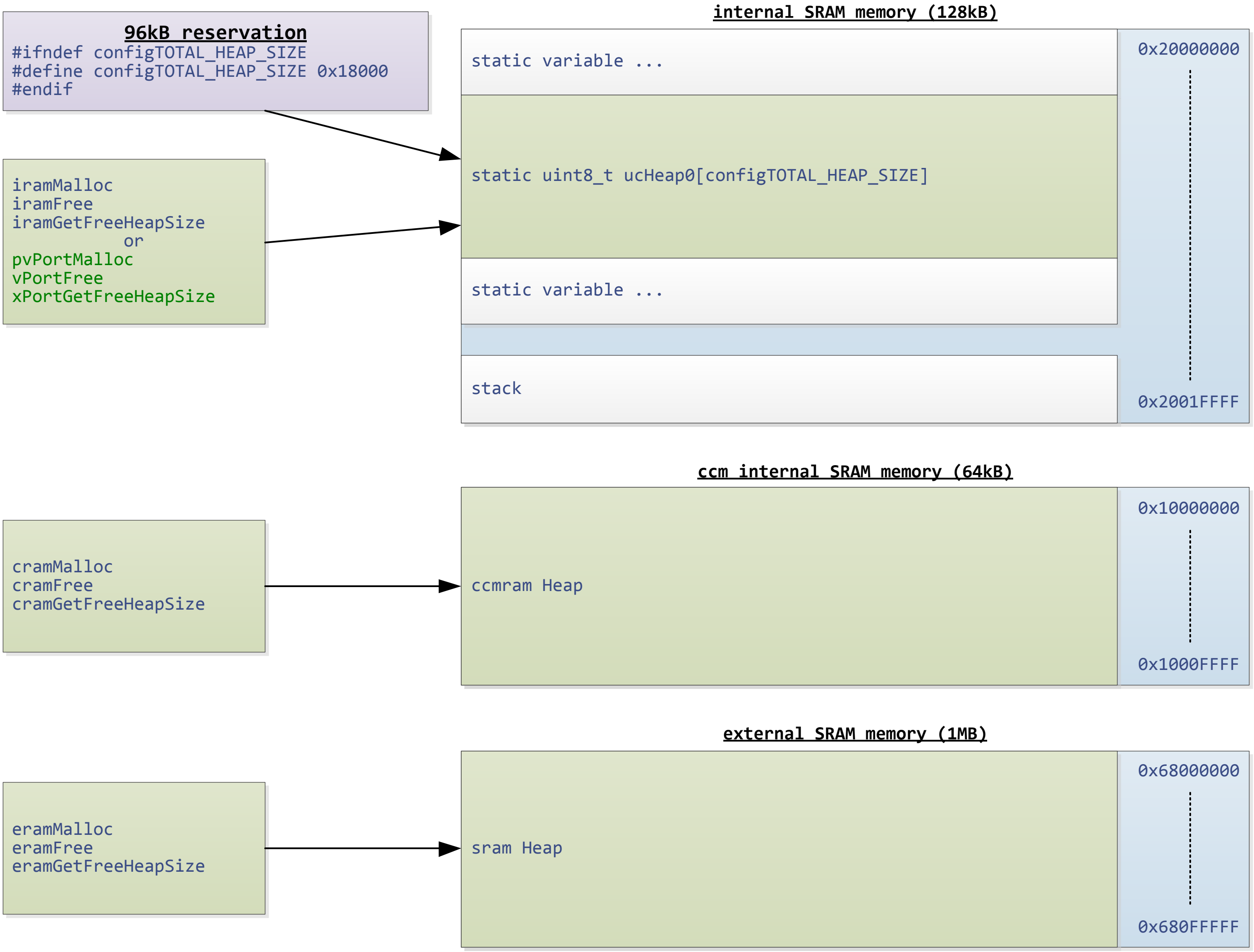
# Multi region heap memory divide example
# (stm32f407 black board with 1MB external sram)



**96kB reservation**

```
#ifndef configTOTAL_HEAP_SIZE
#define configTOTAL_HEAP_SIZE 0x18000
#endif
```

iramMalloc
iramFree
iramGetFreeHeapSize
or
pvPortMalloc
vPortFree
xPortGetFreeHeapSize

**internal SRAM memory (128kB)**

static variable ...
static uint8_t ucHeap0[configTOTAL_HEAP_SIZE]
static variable ...
stack

0x20000000
0x2001FFFF

cramMalloc
cramFree
cramGetFreeHeapSize

**ccm internal SRAM memory (64kB)**

ccmram Heap

0x10000000
0x1000FFFF

eramMalloc
eramFree
eramGetFreeHeapSize

**external SRAM memory (1MB)**

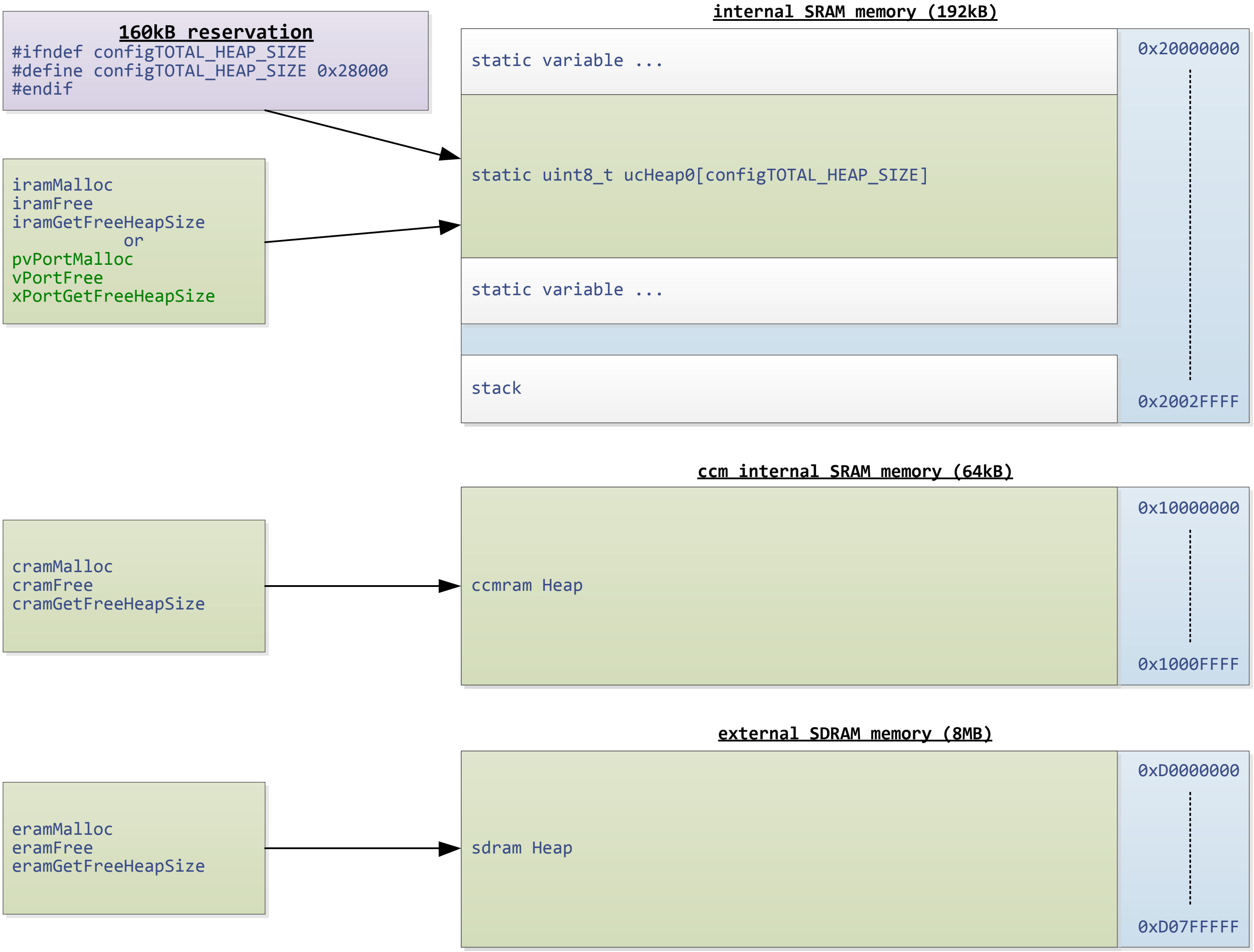
sram Heap

0x68000000
0x680FFFFF

## multi_heap_4.h

```
/* Heap region number (1..6) */
#define HEAP_NUM      3

/* region 0 heap static reservation (if not used freertos -> check the free RAM size for setting) */
#ifndef configTOTAL_HEAP_SIZE
#define configTOTAL_HEAP_SIZE    0x18000
#endif

#define HEAP_0        ucHeap0[configTOTAL_HEAP_SIZE] // internal sram heap reservation if freertos used

/* regions table: adress and size (internal sram region (0), ccmram region (1), external ram region (2) ) */
#define HEAP_REGIONS  {{ (uint8_t *) &ucHeap0, sizeof(ucHeap0) },\
                       { (uint8_t *) 0x10000000, 0x10000       },\
                       { (uint8_t *) 0x68000000, 0x100000      }};

/* internal sram memory (region 0) procedures (*iramMalloc, iramFree, iramGetFreeHeapSize) */
#define iramMalloc(a)        multiPortMalloc(0, a)
#define iramFree(a)          multiPortFree(0, a)
#define iramGetFreeHeapSize(a)  multiPortGetFreeHeapSize(0)
#define iramGetMinimumEverFreeHeapSize multiPortGetMinimumEverFreeHeapSize(0)

/* ccm ram memory (region 1) procedures (*cramMalloc, cramFree, cramGetFreeHeapSize) */
#define cramMalloc(a)        multiPortMalloc(1, a)
#define cramFree(a)          multiPortFree(1, a)
#define cramGetFreeHeapSize(a)  multiPortGetFreeHeapSize(1)
#define cramGetMinimumEverFreeHeapSize multiPortGetMinimumEverFreeHeapSize(1)

/* external ram memory (region 2) procedures (*eramMalloc, eramFree, eramGetFreeHeapSize) */
#define eramMalloc(a)      multiPortMalloc(2, a)
#define eramFree(a)        multiPortFree(2, a)
#define eramGetFreeHeapSize(a)  multiPortGetFreeHeapSize(2)
#define eramGetMinimumEverFreeHeapSize multiPortGetMinimumEverFreeHeapSize(2)
```

# Multi region heap memory divide example
# (stm32f429 discovery board with 8MB external sdram)



**160kB reservation**

```
#ifndef configTOTAL_HEAP_SIZE
#define configTOTAL_HEAP_SIZE 0x28000
#endif
```

```
iramMalloc
iramFree
iramGetFreeHeapSize
          or
pvPortMalloc
vPortFree
xPortGetFreeHeapSize
```

**internal SRAM memory (192kB)**

```
static variable ...
static uint8_t ucHeap0[configTOTAL_HEAP_SIZE]
static variable ...
stack
```

0x20000000 – 0x2002FFFF

```
cramMalloc
cramFree
cramGetFreeHeapSize
```

**ccm internal SRAM memory (64kB)**

ccmram Heap

0x10000000 – 0x1000FFFF

```
eramMalloc
eramFree
eramGetFreeHeapSize
```

**external SDRAM memory (8MB)**

sdram Heap

0xD0000000 – 0xD07FFFFF

**multi_heap_4.h**

```
/* Heap region number (1..6) */
#define HEAP_NUM      3

/* region 0 heap static reservation (if not used freertos -> check the free RAM size for setting) */
#ifndef configTOTAL_HEAP_SIZE
#define configTOTAL_HEAP_SIZE    0x28000
#endif

#define HEAP_0        ucHeap0[configTOTAL_HEAP_SIZE] // internal sram heap reservation if freertos used

/* regions table: adress and size (internal sram region (0), ccmram region (1), external ram region (2) ) */
#define HEAP_REGIONS  {{ (uint8_t *) &ucHeap0, sizeof(ucHeap0) },\
                       { (uint8_t *) 0x10000000, 0x10000       },\
                       { (uint8_t *) 0xD0000000, 0x800000      }};

/* internal sram memory (region 0) procedures (*iramMalloc, iramFree, iramGetFreeHeapSize) */
#define iramMalloc(a)        multiPortMalloc(0, a)
#define iramFree(a)          multiPortFree(0, a)
#define iramGetFreeHeapSize(a)  multiPortGetFreeHeapSize(0)
#define iramGetMinimumEverFreeHeapSize multiPortGetMinimumEverFreeHeapSize(0)

/* ccm ram memory (region 1) procedures (*cramMalloc, cramFree, cramGetFreeHeapSize) */
#define cramMalloc(a)        multiPortMalloc(1, a)
#define cramFree(a)          multiPortFree(1, a)
#define cramGetFreeHeapSize(a)  multiPortGetFreeHeapSize(1)
#define cramGetMinimumEverFreeHeapSize multiPortGetMinimumEverFreeHeapSize(1)

/* external ram memory (region 2) procedures (*eramMalloc, eramFree, eramGetFreeHeapSize) */
#define eramMalloc(a)      multiPortMalloc(2, a)
#define eramFree(a)        multiPortFree(2, a)
#define eramGetFreeHeapSize(a)  multiPortGetFreeHeapSize(2)
#define eramGetMinimumEverFreeHeapSize multiPortGetMinimumEverFreeHeapSize(2)
```