

In My Codes

- In my main.c:

```
/* USER CODE BEGIN Includes */
#include <string.h>
/* USER CODE END Includes */

...

/* USER CODE BEGIN PV */
__attribute__((section(".sdram"))) unsigned char myMemoryPool[4096];
/* USER CODE END PV */

...

int main(void)
{
    /* USER CODE BEGIN 1 */
    memset(myMemoryPool, 0xFF, sizeof(myMemoryPool));
    /* USER CODE END 1 */

    ...
}

...
```

- In my STM32H750XBHX_FLASH.ld:

```
MEMORY
{
    FLASH    (rx)    : ORIGIN = 0x8000000,   LENGTH = 128K
    DTCMRAM  (rwx)   : ORIGIN = 0x20000000,  LENGTH = 128K
    RAM_D1   (rwx)   : ORIGIN = 0x24000000,  LENGTH = 512K
    ITCMRAM  (rwx)   : ORIGIN = 0x00000000,  LENGTH = 64K
    RAM_D3   (rwx)   : ORIGIN = 0x38000000,  LENGTH = 64K
    RAM_D2   (rwx)   : ORIGIN = 0x30000000,  LENGTH = 288K
    SDRAM    (rwx)   : ORIGIN = 0xC0000000,  LENGTH = 16384K
}

SECTIONS
{
    ...

    .bss :
    {
        ...
    } >RAM_D1

    .sdram_section :
    {
        . = ALIGN(4);
        __SDARAM_START__ = .;
    }
```

```

*(.sdrum)
*(.sdrum*)
. = ALIGN(4);
__SDRAM_END__ = .;
} >SDRAM

._user_heap_stack :
{
...
} >RAM_D1
...
}

```

- In my STM32H750XBHX_FLASH_NOLOAD.ld:

```

MEMORY
{
    FLASH (rx) : ORIGIN = 0x8000000, LENGTH = 128K
    DTCMRAM (rwx) : ORIGIN = 0x20000000, LENGTH = 128K
    RAM_D1 (rwx) : ORIGIN = 0x24000000, LENGTH = 512K
    ITCMRAM (rwx) : ORIGIN = 0x00000000, LENGTH = 64K
    RAM_D3 (rwx) : ORIGIN = 0x38000000, LENGTH = 64K
    RAM_D2 (rwx) : ORIGIN = 0x30000000, LENGTH = 288K
    SDRAM (rwx) : ORIGIN = 0xC0000000, LENGTH = 16384K
}

SECTIONS
{
...

.bss :
{
...
} >RAM_D1

.sdram_section (NOLOAD) :
{
. = ALIGN(4);
__SDARAM_START__ = .;
*(.sdrum)
*(.sdrum*)
. = ALIGN(4);
__SDRAM_END__ = .;
} >SDRAM

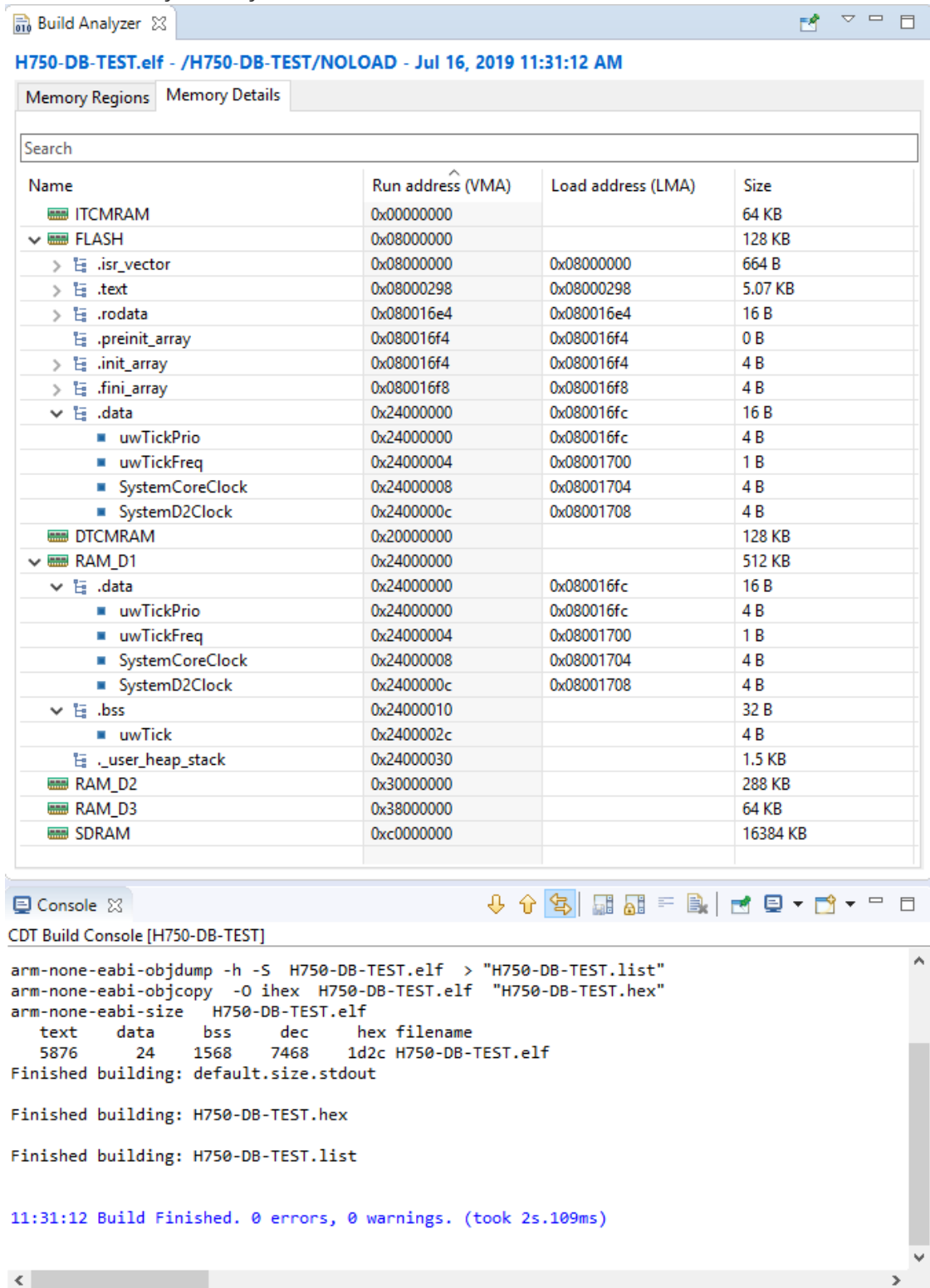
._user_heap_stack :
{
...
} >RAM_D1
...
}

```

Test Result

Here is the result of three test cases by using "Build Analyzer" in STM32CubeIDE:

1. Build without myMemoryPool:



The screenshot displays the Build Analyzer window for the project H750-DB-TEST.elf. The 'Memory Regions' tab is active, showing a table of memory regions with their names, run addresses (VMA), load addresses (LMA), and sizes. Below the table is the 'Console' window showing the build process output.

Name	Run address (VMA)	Load address (LMA)	Size
ITCMRAM	0x00000000		64 KB
FLASH	0x08000000		128 KB
.isr_vector	0x08000000	0x08000000	664 B
.text	0x08000298	0x08000298	5.07 KB
.rodata	0x080016e4	0x080016e4	16 B
.preinit_array	0x080016f4	0x080016f4	0 B
.init_array	0x080016f4	0x080016f4	4 B
.fini_array	0x080016f8	0x080016f8	4 B
.data	0x24000000	0x080016fc	16 B
uwTickPrio	0x24000000	0x080016fc	4 B
uwTickFreq	0x24000004	0x08001700	1 B
SystemCoreClock	0x24000008	0x08001704	4 B
SystemD2Clock	0x2400000c	0x08001708	4 B
DTCMRAM	0x20000000		128 KB
RAM_D1	0x24000000		512 KB
.data	0x24000000	0x080016fc	16 B
uwTickPrio	0x24000000	0x080016fc	4 B
uwTickFreq	0x24000004	0x08001700	1 B
SystemCoreClock	0x24000008	0x08001704	4 B
SystemD2Clock	0x2400000c	0x08001708	4 B
.bss	0x24000010		32 B
uwTick	0x2400002c		4 B
.user_heap_stack	0x24000030		1.5 KB
RAM_D2	0x30000000		288 KB
RAM_D3	0x38000000		64 KB
SDRAM	0xc0000000		16384 KB

```
CDT Build Console [H750-DB-TEST]
arm-none-eabi-objdump -h -S H750-DB-TEST.elf > "H750-DB-TEST.list"
arm-none-eabi-objcopy -O ihex H750-DB-TEST.elf "H750-DB-TEST.hex"
arm-none-eabi-size H750-DB-TEST.elf
  text  data  bss   dec   hex filename
 5876   24  1568  7468  1d2c H750-DB-TEST.elf
Finished building: default.size.stdout

Finished building: H750-DB-TEST.hex

Finished building: H750-DB-TEST.list

11:31:12 Build Finished. 0 errors, 0 warnings. (took 2s.109ms)
```

2. Build with myMemoryPool but without NOLOAD used in **.sdram section** inside linker script (using STM32H750XBHX_FLASH.ld):

- o Build Analyzer and CDT Build Console (showing result of arm-none-eabi-size):

The image shows two windows from an IDE. The top window is 'Build Analyzer' displaying memory regions for the file 'H750-DB-TEST.elf'. The bottom window is 'Console' showing the output of the 'arm-none-eabi-size' command.

Build Analyzer - Memory Regions

Name	Run address (VMA)	Load address (LMA)	Size
ITCMRAM	0x00000000		64 KB
FLASH	0x08000000		128 KB
> .isr_vector	0x08000000	0x08000000	664 B
> .text	0x08000298	0x08000298	5.09 KB
> .rodata	0x080016f4	0x080016f4	16 B
> .preinit_array	0x08001704	0x08001704	0 B
> .init_array	0x08001704	0x08001704	4 B
> .fini_array	0x08001708	0x08001708	4 B
> .data	0x24000000	0x0800170c	16 B
uwTickPrio	0x24000000	0x0800170c	4 B
uwTickFreq	0x24000004	0x08001710	1 B
SystemCoreClock	0x24000008	0x08001714	4 B
SystemD2Clock	0x2400000c	0x08001718	4 B
DTCMRAM	0x20000000		128 KB
RAM_D1	0x24000000		512 KB
> .data	0x24000000	0x0800170c	16 B
uwTickPrio	0x24000000	0x0800170c	4 B
uwTickFreq	0x24000004	0x08001710	1 B
SystemCoreClock	0x24000008	0x08001714	4 B
SystemD2Clock	0x2400000c	0x08001718	4 B
> .bss	0x24000010		32 B
uwTick	0x2400002c		4 B
> .user_heap_stack	0x24000030		1.5 KB
RAM_D2	0x30000000		288 KB
RAM_D3	0x38000000		64 KB
SDRAM	0xc0000000		16384 KB
> .sdram_section	0xc0000000	0xc0000000	4 KB
myMemoryPool	0xc0000000	0xc0000000	4 KB

Console - CDT Build Console [H750-DB-TEST]

```

arm-none-eabi-size H750-DB-TEST.elf
arm-none-eabi-objdump -h -S H750-DB-TEST.elf > "H750-DB-TEST.list"
text data bss dec hex filename
5892 4120 1568 11580 2d3c H750-DB-TEST.elf
arm-none-eabi-objcopy -O ihex H750-DB-TEST.elf "H750-DB-TEST.hex"
Finished building: default.size.stdout

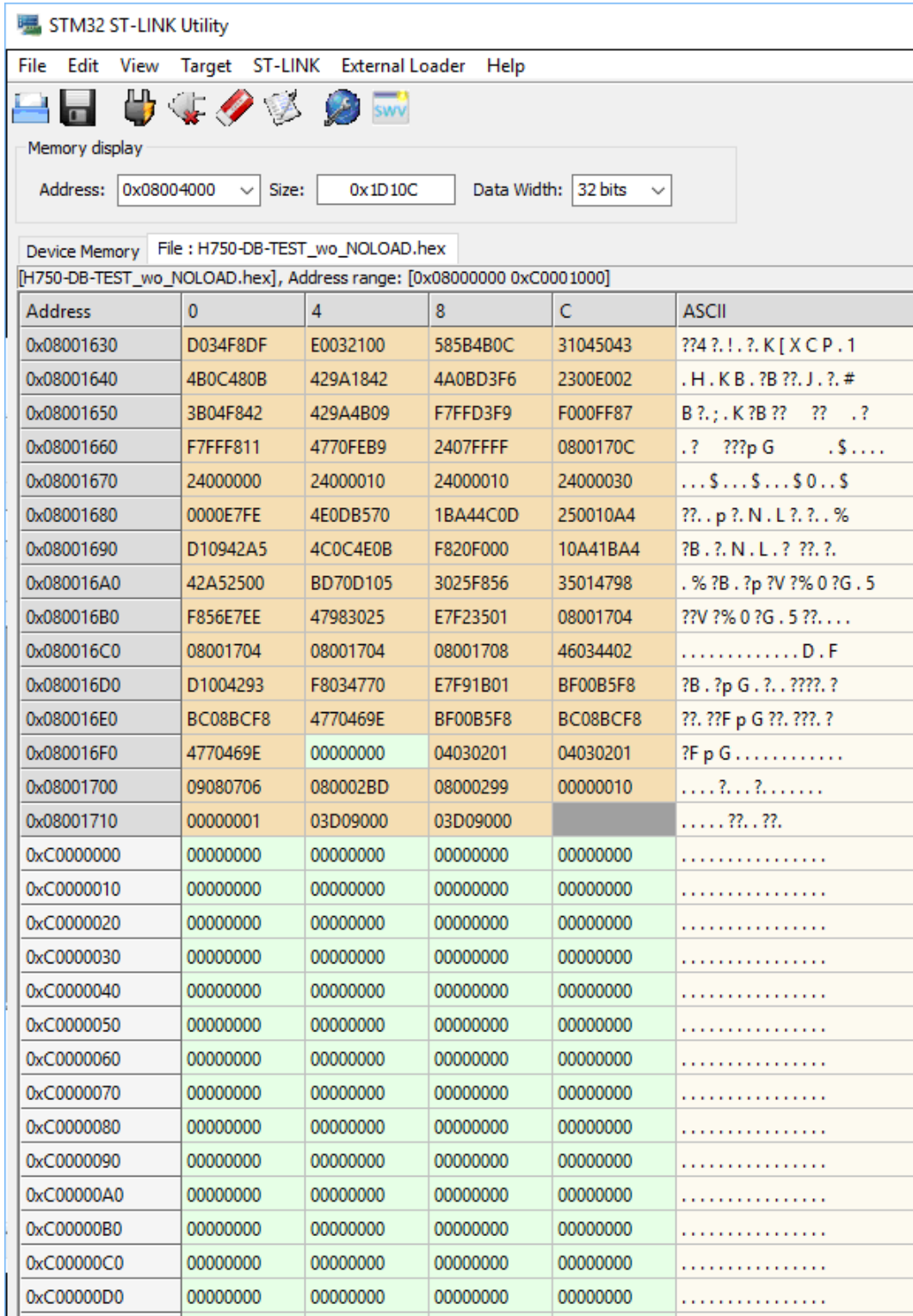
Finished building: H750-DB-TEST.list
Finished building: H750-DB-TEST.hex

|

13:23:30 Build Finished. 0 errors, 0 warnings. (took 5s.200ms)

```

- o Checking HEX file by ST-Link-Utility:



- According to ST-LINK-Utility, Hex file contains content from 0xC0000000. This also leads to large binary file.
- According to "Build Analyzer", myMemoryPool is put in **.sdram section** according to "Memory Detail" in Build Analyzer.
- According to "Build Analyzer", **.sdram section** has VMA.
- According to "CDT Build Console", "data" shown by arm-none-eabi-size increases to 4120 from 24.

3. Build with myMemoryPool and with NOLOAD used in **.sdram section** inside linker script (using STM32H750XBHX_FLASH_NOLOAD.Id):

- o Build Analyzer and CDT Build Console (showing result of arm-none-eabi-size):

The screenshot displays the Build Analyzer interface for the file `H750-DB-TEST.elf` at `/H750-DB-TEST/NOLOAD` on July 16, 2019, at 1:25:09 PM. The interface is split into two main sections: Memory Regions and Console.

Memory Regions: This section shows a table of memory regions with columns for Name, Run address (VMA), Load address (LMA), and Size. The regions are categorized into ITCMRAM, FLASH, DTCMRAM, RAM_D1, RAM_D2, RAM_D3, and SDRAM. The FLASH region is expanded to show sub-sections like `.isr_vector`, `.text`, `.rodata`, `.preinit_array`, `.init_array`, `.fini_array`, and `.data`. The `.data` section is further expanded to show variables like `uwTickPrio`, `uwTickFreq`, `SystemCoreClock`, and `SystemD2Clock`. The RAM_D1 region is also expanded to show similar variables. The SDRAM region is expanded to show `.sdrum_section` and `myMemoryPool`.

Name	Run address (VMA)	Load address (LMA)	Size
ITCMRAM	0x00000000		64 KB
FLASH	0x08000000		128 KB
<code>.isr_vector</code>	0x08000000	0x08000000	664 B
<code>.text</code>	0x08000298	0x08000298	5.09 KB
<code>.rodata</code>	0x080016f4	0x080016f4	16 B
<code>.preinit_array</code>	0x08001704	0x08001704	0 B
<code>.init_array</code>	0x08001704	0x08001704	4 B
<code>.fini_array</code>	0x08001708	0x08001708	4 B
<code>.data</code>	0x24000000	0x0800170c	16 B
<code>uwTickPrio</code>	0x24000000	0x0800170c	4 B
<code>uwTickFreq</code>	0x24000004	0x08001710	1 B
<code>SystemCoreClock</code>	0x24000008	0x08001714	4 B
<code>SystemD2Clock</code>	0x2400000c	0x08001718	4 B
DTCMRAM	0x20000000		128 KB
RAM_D1	0x24000000		512 KB
<code>.data</code>	0x24000000	0x0800170c	16 B
<code>uwTickPrio</code>	0x24000000	0x0800170c	4 B
<code>uwTickFreq</code>	0x24000004	0x08001710	1 B
<code>SystemCoreClock</code>	0x24000008	0x08001714	4 B
<code>SystemD2Clock</code>	0x2400000c	0x08001718	4 B
<code>.bss</code>	0x24000010		32 B
<code>uwTick</code>	0x2400002c		4 B
<code>._user_heap_stack</code>	0x24000030		1.5 KB
RAM_D2	0x30000000		288 KB
RAM_D3	0x38000000		64 KB
SDRAM	0xc0000000		16384 KB
<code>.sdrum_section</code>	0xc0000000		4 KB
<code>myMemoryPool</code>	0xc0000000		4 KB

Console: The console shows the output of the `arm-none-eabi-objcopy` command. It displays the memory layout for the `H750-DB-TEST.elf` file, showing the text, data, bss, and dec sections. The output indicates that the build finished successfully with 0 errors and 0 warnings, taking 5s.352ms.

```

CDT Build Console [H750-DB-TEST]
arm-none-eabi-objcopy -O ihex H750-DB-TEST.elf "H750-DB-TEST.hex"
  text  data  bss  dec  hex filename
 5892   24  5664 11580 2d3c H750-DB-TEST.elf
Finished building: default.size.stdout

Finished building: H750-DB-TEST.hex

Finished building: H750-DB-TEST.list

13:25:09 Build Finished. 0 errors, 0 warnings. (took 5s.352ms)

```

- o Checking HEX file by ST-Link-Utility:

STM32 ST-LINK Utility

File Edit View Target ST-LINK External Loader Help

Memory display

Address: 0x08004000 Size: 0x1D10C Data Width: 32 bits

Device Memory File : H750-DB-TEST_wi_NOLOAD.hex

[H750-DB-TEST_wi_NOLOAD.hex], Address range: [0x08000000 0x0800171C]

Address	0	4	8	C	ASCII
0x08001550	B4804770	BF00AF00	F85D46BD	47707B04	p G ? . ? . ??F] ? . { p G
0x08001560	AF00B580	FF26F7FE	BD80BF00	AF00B480	? . ???& ? . ? ? . ?
0x08001570	F8D34B29	4A283088	0370F443	3088F8C2) K ???0 (J C ? p . ???0
0x08001580	681B4B26	F0434A25	60130301	22004B23	& K . h % J C ? . . ` # K . "
0x08001590	4B22611A	4921681A	40134B21	4B1F600B	. a " K . h ! ! ! K . @ . ` . K
0x080015A0	619A2200	22004B1D	4B1C61DA	621A2200	. " ? a . K . " ? a . K . " . b
0x080015B0	22004B1A	4B19629A	62DA2200	22004B17	. K . " ? b . K . " ? b . K . "
0x080015C0	4B16631A	635A2200	22004B14	4B13639A	. c . K . " Z c . K . " ? c . K
0x080015D0	63DA2200	22004B11	4B10641A	645A2200	. " ? c . K . " . d . K . " Z d
0x080015E0	681B4B0E	F4234A0D	60132380	22004B0B	. K . h . J # ? # . ` . K . "
0x080015F0	4B0C661A	4B0C681A	F1B34013	D2025F00	. f . K . h . K . @ ? ? . _ . ?
0x08001600	22014B0A	4B04601A	6200F04F	BF00609A	. K . " . ` . K O ? . b ? . ?
0x08001610	F85D46BD	47707B04	E000ED00	58024400	?F] ? . { p G . ? . ? . D . X
0x08001620	EAF6ED7F	5C001000	FFFF0000	51008108	???. . . \ . . . ? . Q
0x08001630	D034F8DF	E0032100	585B4B0C	31045043	??4 ? . ! . ? . K [X C P . 1
0x08001640	4B0C480B	429A1842	4A0BD3F6	2300E002	. H . K B . ? B ? ? . J . ? . #
0x08001650	3B04F842	429A4B09	F7FFD3F9	F000FF87	B ? ; . K ? B ? ? ? ? . ?
0x08001660	F7FFF811	4770FEB9	2407FFFF	0800170C	. ? ? ? ? p G . \$
0x08001670	24000000	24000010	24000010	24000030	. . . \$. . . \$. . . \$. . . \$
0x08001680	0000E7FE	4E0DB570	1BA44C0D	250010A4	?? . . p ? . N . L ? . ? . %
0x08001690	D10942A5	4C0C4E0B	F820F000	10A41BA4	? B . ? . N . L . ? ? ? . ?
0x080016A0	42A52500	BD70D105	3025F856	35014798	. % ? B . ? p ? V ? % 0 ? G . 5
0x080016B0	F856E7EE	47983025	E7F23501	08001704	??V ? % 0 ? G . 5 ? ? . . .
0x080016C0	08001704	08001704	08001708	46034402 D . F
0x080016D0	D1004293	F8034770	E7F91B01	BF00B5F8	? B . ? p G . ? . ? ? ? . ?
0x080016E0	BC08BCF8	4770469E	BF00B5F8	BC08BCF8	?? . ??F p G ?? . ??? . ?
0x080016F0	4770469E	00000000	04030201	04030201	?F p G
0x08001700	09080706	080002BD	08000299	00000010 ? . . ?
0x08001710	00000001	03D09000	03D09000	 ? . . ? ? .

13:29:19 : [H750-DB-TEST_wo_NOLOAD.hex] opened successfully.
 Address Ranges [0x08000000 0x0800171C] [0xC0000000 0xC0001000]
 13:29:19 : [H750-DB-TEST_wo_NOLOAD.hex] checksum : 0x00084FFE
 13:30:21 : [H750-DB-TEST_wi_NOLOAD.hex] opened successfully.
 13:30:21 : [H750-DB-TEST_wi_NOLOAD.hex] checksum : 0x00084FFE

Disconnected Device ID : -----

- According to ST-LINK-Utility, Hex file DOES NOT contain content from 0xC0000000.
- According to "Build Analyzer", myMemoryPool is put in **.sdram section** according to "Memory Detail" in Build Analyzer.

- According to "Build Analyzer", **.sdram section** DOES NOT have VMA (like what .bss does).
- According to "CDT Build Console", "bss" shown by arm-none-eabi-size increases to 4120 from 24.

Some additional observations:

1. Comparing arm-none-eabi-size:

(Left: w/o NOLOAD; Right: w/i NOLOAD)

```

arm-none-eabi-size H750-DB-TEST-w/o-NOLOAD.elf
text  data  bss  dec  hex  filename
5892  4120  1568  11580  2d3c  H750-DB-TEST-w/o-NOLOAD.elf

arm-none-eabi-size --format=Berkeley H750-DB-TEST-w/o-NOLOAD.elf
text  data  bss  dec  hex  filename
5892  4120  1568  11580  2d3c  H750-DB-TEST-w/o-NOLOAD.elf

arm-none-eabi-size --format=SysV H750-DB-TEST-w/o-NOLOAD.elf
H750-DB-TEST-w/o-NOLOAD.elf:
section      size      addr
.isr_vector  664      134217728
.text        5212     134223952
.rodata      16       134223604
.ARM_extab   0         134223620
.ARM         0         134223620
.prelimit_array  4       134223620
.init_array  4         134223620
.fini_array  4         134223624
.data        16       683979776
.bss         24       683979792
.sdram_section 4096     321225472
.user_heap_stack 1568     683979824
.ARM_attributes 46       0
.debug_info  41189    0
.debug_abbrev 5224     0
.debug_ranges 1440     0
.debug_macro 208743  0
.debug_line  19716    0
.debug_str   1335244  0
.comment    123      0
.debug_frame 5412     0
Total       1638808

arm-none-eabi-size H750-DB-TEST-w/i-NOLOAD.elf
text  data  bss  dec  hex  filename
5892  24  5664  11580  2d3c  H750-DB-TEST-w/i-NOLOAD.elf

arm-none-eabi-size --format=Berkeley H750-DB-TEST-w/i-NOLOAD.elf
text  data  bss  dec  hex  filename
5892  24  5664  11580  2d3c  H750-DB-TEST-w/i-NOLOAD.elf

arm-none-eabi-size --format=SysV H750-DB-TEST-w/i-NOLOAD.elf
H750-DB-TEST-w/i-NOLOAD.elf:
section      size      addr
.isr_vector  664      134217728
.text        5212     134223952
.rodata      16       134223604
.ARM_extab   0         134223620
.ARM         0         134223620
.prelimit_array  4       134223620
.init_array  4         134223620
.fini_array  4         134223624
.data        16       683979776
.bss         24       683979792
.sdram_section 4096     321225472
.user_heap_stack 1568     683979824
.ARM_attributes 46       0
.debug_info  41189    0
.debug_abbrev 5224     0
.debug_ranges 1440     0
.debug_macro 208743  0
.debug_line  19716    0
.debug_str   1335244  0
.comment    123      0
.debug_frame 5412     0
Total       1638801
  
```

- By using `--format=SysV`, `.data` and `.bss` sections have the same size in both cases.
- **.sdram section** occupies 4096 which is the size of myMemoryPool.

2. Comparing objdump -x:

(Left: w/o NOLOAD; Right: w/i NOLOAD)

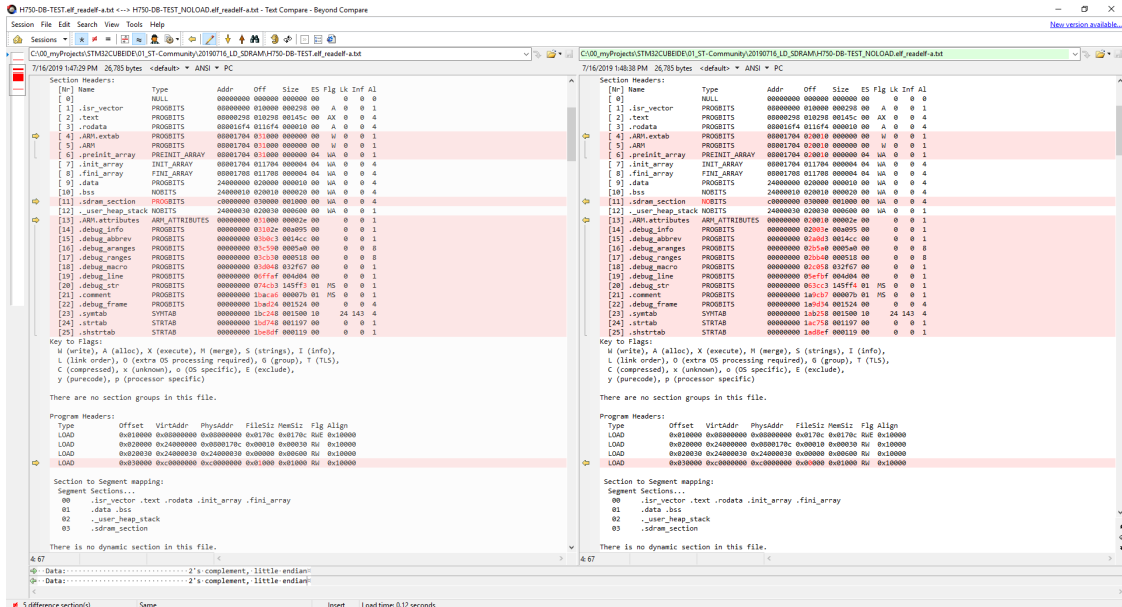
```

Sections:
Idx Name          Size      VMA               LMA               File off  Algn
0 .isr_vector     0x00000000 0x00000000         0x00000000        0x00000000 2**0
1 .text          0x00000298 0x00000000         0x00000000        0x00000298 2**2
2 .rodata        0x00000010 0x00000000         0x00000000        0x00000010 2**2
3 .ARM_extab     0x00000000 0x00017840         0x00017840        0x00000018 2**0
4 .ARM           0x00000000 0x00017840         0x00017840        0x00000018 2**0
5 .preinit_array 0x00000004 0x00017840         0x00017840        0x00000018 2**0
6 .init_array    0x00000004 0x00017840         0x00017840        0x00000018 2**2
7 .fini_array    0x00000004 0x00017840         0x00017840        0x00000018 2**2
8 .data          0x00000010 0x00020000         0x00020000        0x00020000 2**2
9 .bss           0x00000024 0x00020010         0x00020010        0x00020010 2**2
10 .sdram_section 0x00000010 0xc0000000         0xc0000000        0x00030000 2**2
11 .user_heap_stack 0x00000500 0x24000030         0x24000030        0x00020030 2**0
12 .ARM_attributes 0x0000002e 0x00000000         0x00000000        0x00010010 2**0
13 .debug_info    0x00000405 0x00000000         0x00000000        0x0001001c 2**0
14 .debug_abbrev 0x0000012c 0x00000000         0x00000000        0x0001001c 2**0
15 .debug_ranges 0x000005ab 0x00000000         0x00000000        0x0001001c 2**3
16 .debug_macro  0x00000018 0x00000000         0x00000000        0x0001001c 2**3
17 .debug_line    0x00000040 0x00000000         0x00000000        0x0001001f 2**0
18 .debug_str     0x00000404 0x00000000         0x00000000        0x0001001f 2**0
19 .debug_str     0x00000404 0x00000000         0x00000000        0x0001001f 2**0
20 .comment       0x00000075 0x00000000         0x00000000        0x0001001f 2**0
21 .debug_frame   0x00000124 0x00000000         0x00000000        0x0001001f 2**2
SYMBOL TABLE:
0x00000000 l d .isr_vector 0x00000000 .isr_vector
0x00000298 l d .text 0x00000000 .text
0x00000010 d .rodata 0x00000000 .rodata
0x00017840 l d .ARM_extab 0x00000000 .ARM_extab

```

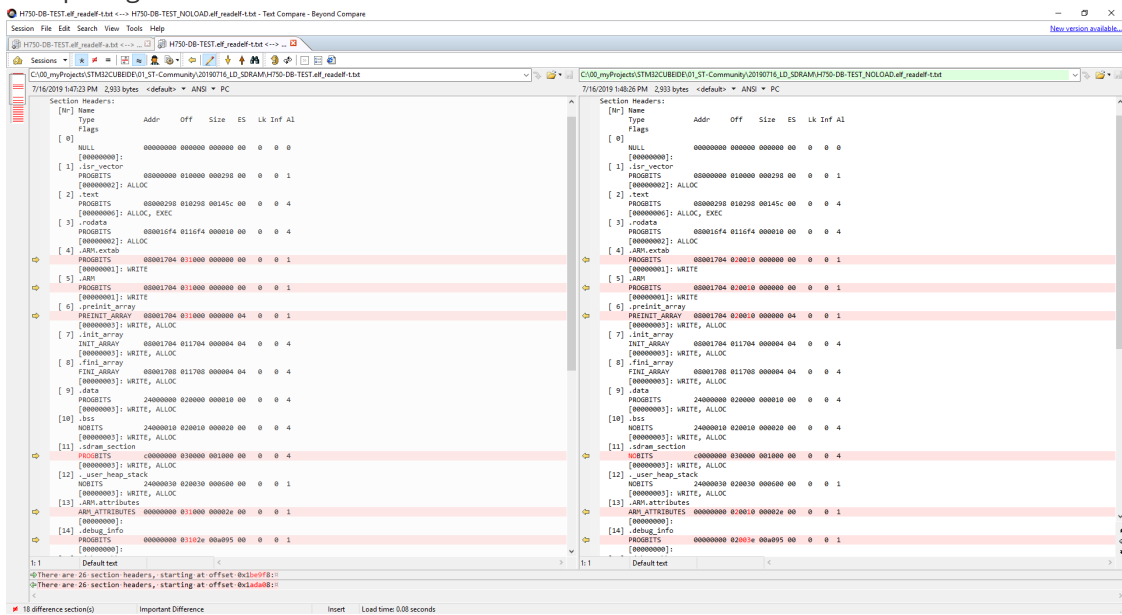
- **.sdram section** is "CONTENTS" when NOLOAD is not used.

3. Comparing readelf -a: (Left: w/o NOLOAD; Right: w/ NOLOAD)



- o **.sdrum section** is "PROGBITS" type when NOLOAD is not used and is "NOBITS" type when NOLOAD is used.

4. Comparing readelf -t:



- o **.sdrum section** contains "PROGBITS" flag when NOLOAD is not used and contains "NOBITS" flags when NOLOAD is used.

Questions from candyLife91

1. Where linker put my self-defined section to? for example i defined a sdrum_section output section in the linker script, but i didn't see it in the build output information.

[My answer]:

According to "Build Analyzer" in STM32CubeIDE, customized output section **.sdrum_section** which contains myMemoryPool is indeed generated. Customized section is indeed NOT shown in the output of arm-none-eabi-size, it only shows "standard ELF sections (?)" text/data/bss.

2. why my uninitialized huge array was put to the data section, from what i understand ,this data section is for the initialized data use.

[My answer]:

According to "Build Analyzer" in STM32CubeIDE as well as result of using some other tools (such as objdump and readelf shown above), myMemoryPool is actually put in **.sdram section** and not in either .data or .bss section.

I think the reason you feel myMemoryPool is put in data section is that you refer to the result generated by arm-none-eabi-size which only shows "standard ELF sections" text/data/bss as mentioned in my answer #1. I do not know how arm-none-eabi-size handles the size of something put in customized section. My assumption, however, is that:

- arm-none-eabi-size shows the size of data by summing up the size of all symbols put in .data section along with that in the other sections with flags "CONTENTS, ALLOC, LOAD, DATA" shown in objdump (such as .preinit_array, init_array, fini_array and **.sdram section** when NOLOAD is NOT used in this case).
- Similarly, arm-none-eabi-size shows the size of bss by summing up the size of all symbols put in .bss section along with that in the other sections with flags "ALLOC" only (**.sdram section** when NOLOAD is used in this case) shown in objdump.