

Vorlesungsskript zu „Vertiefung Programmieren“ Boot Prozess



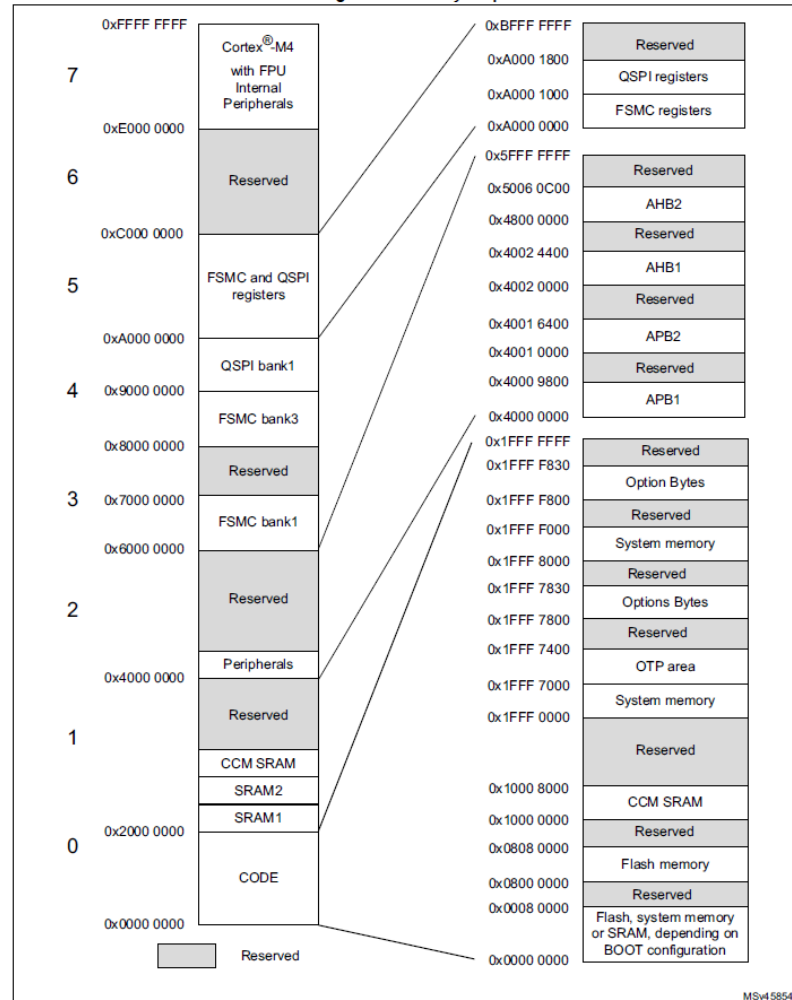
Dozent: Dipl.-Inf. (FH) Andreas Schmidt

Bootprozess

Bootprozess STM32G474

Memory Map Cortex-M4

Figure 2. Memory map



Quelle: RM0440 Reference Manual

Memory Map STM32G474

Table 6. Memory mapping versus boot mode/physical remap⁽¹⁾

Addresses	Boot/remap in main Flash memory	Boot/remap in embedded SRAM 1	Boot/remap in system memory	Remap in FSMC	Remap in QUADSPI
0x2000 0000 - 0x2002 3FFF	SRAM1	SRAM1	SRAM1	SRAM1	SRAM1
0x1FFF 7000 - 0x1FFF FFFF	System memory/OTP/Options bytes	System memory/OTP/Options bytes	System memory/OTP/Options bytes	System memory/OTP/Options bytes	System memory/OTP/Options bytes
0x1000 8000 - 0x1FFE FFFF	Reserved	Reserved	Reserved	Reserved	Reserved
0x1000 0000 - 0x1000 7FFF	CCM SRAM	CCM SRAM	CCM SRAM	CCM SRAM	CCM SRAM
0x0808 0000 - 0x0FFF FFFF	Reserved	Reserved	Reserved	Reserved	Reserved
0x0800 0000 - 0x0807 FFFF	Flash memory	Flash memory	Flash memory	Flash memory	Flash memory
0x0400 0000 - 0x07FF FFFF	Reserved	Reserved	Reserved	FSMC bank 1 NOR/ PSRAM 2 (128 MB) Aliased	QUADSPI bank (128 MB) Aliased
0x0010 0000 - 0x03FF FFFF	Reserved	Reserved	Reserved	FSMC bank 1 NOR/ PSRAM 1 (128 MB) Aliased	QUADSPI bank (128 MB) Aliased
0x0000 0000 - 0x0007 FFFF (2) (3)	Flash Aliased	SRAM1 Aliased	System memory (28 KB) Aliased	FSMC bank 1 NOR/ PSRAM 1 (128 MB) Aliased	QUADSPI Aliased

Quelle: RM0440 Reference Manual

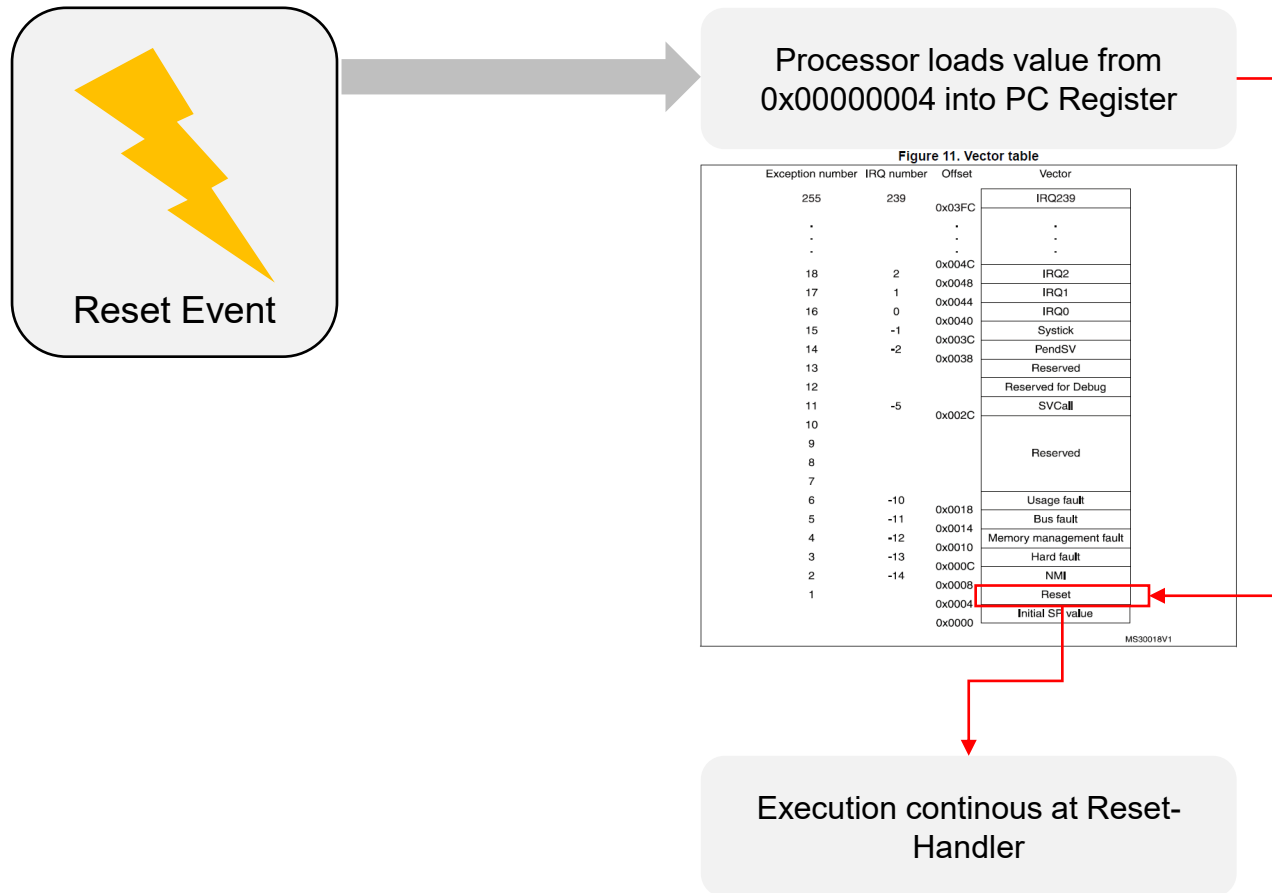
Vector Table STM32G474

Figure 11. Vector table

Exception number	IRQ number	Offset	Vector
255	239	0x03FC	IRQ239
.		.	.
.		.	.
.		.	.
18	2	0x004C	IRQ2
17	1	0x0048	IRQ1
16	0	0x0044	IRQ0
15	-1	0x0040	Systick
14	-2	0x003C	PendSV
13		0x0038	Reserved
12			Reserved for Debug
11	-5	0x002C	SVCaII
10			Reserved
9			
8			
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value

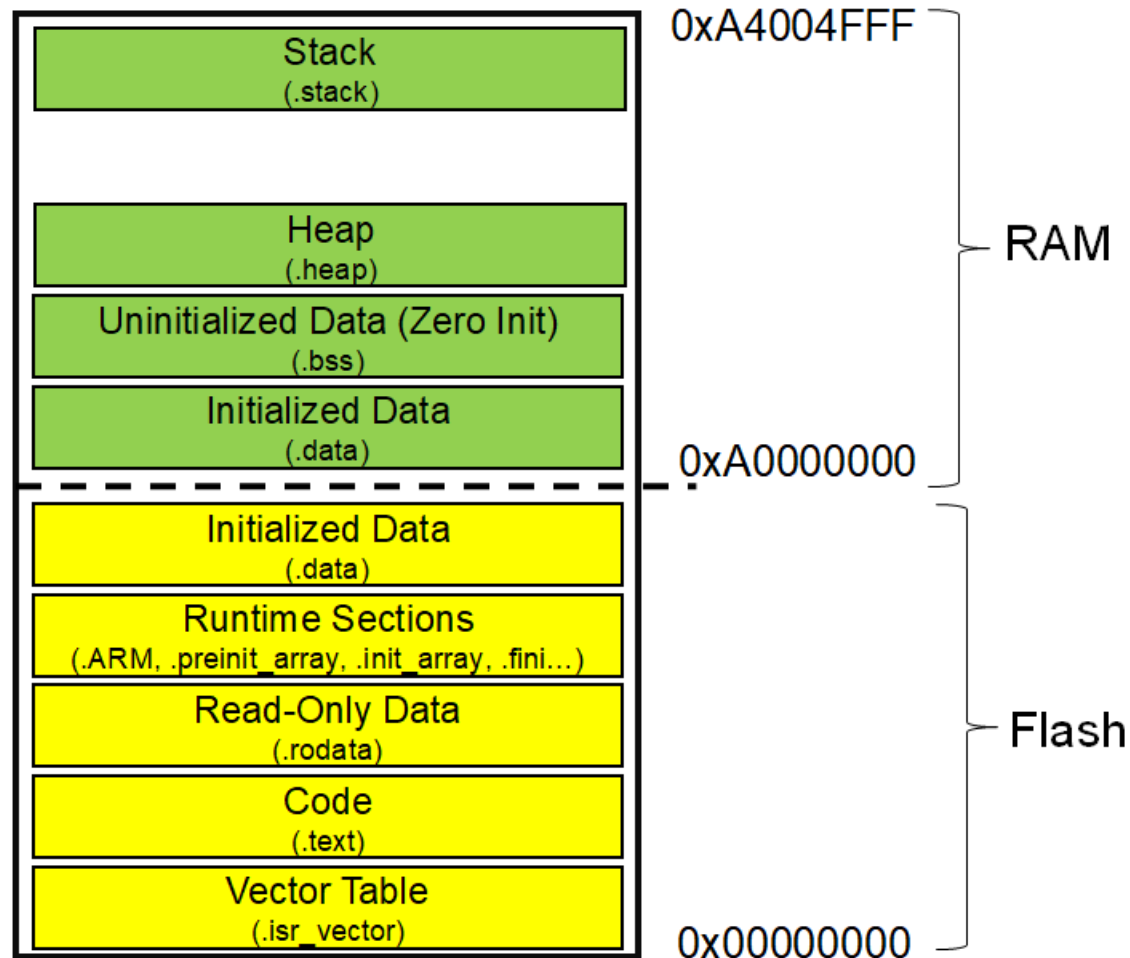
MS30018V1

Reset (Power On Reset (POR) / Signaled Reset



Memory Layout

Beispiel eines Linker-Files und dem dazugehörigen Memory Layout



Reset Handler Code

```
29 .section .text.Reset_Handler
30 .weak Reset_Handler
31 .type Reset_Handler, %function
32 Reset_Handler:
33     /* Copy the data segment initializers from flash to SRAM */
34     ldr r0, =_sdata
35     ldr r1, =_edata
36     ldr r2, =_sidata
37     movs r3, #0
38     b LoopCopyDataInit
39
40 CopyDataInit:
41     ldr r4, [r2, r3]
42     str r4, [r0, r3]
43     adds r3, r3, #4
44
45 LoopCopyDataInit:
46     adds r4, r0, r3
47     cmp r4, r1
48     bcc CopyDataInit
49
50     /* Zero fill the bss segment. */
51     ldr r2, =_sbss
52     ldr r4, =_ebss
53     movs r3, #0
54     b LoopFillZerobss
55
56 FillZerobss:
57     str r3, [r2]
58     adds r2, r2, #4
59
60 LoopFillZerobss:
61     cmp r2, r4
62     bcc FillZerobss
63
64     ldr r0, =_initial_stack_pointer /* load address of initial_stack_pointer
65     mov sp, r0                    /* set stack pointer */
66
67     /* Call the clock system initialization function.*/
68     bl SystemInit
69
70     /* Call the application's entry point.*/
71     bl main
72     bx lr
```

Initialize Data Segment

Initialize BSS Segment

Setup Stack Pointer

Call main()