

1. 上电复位后，从地址 **0x0000_0004** 处获得 **Reset_Handler** 的地址 **0x0000_01f0** 并跳转到该处执行。

2. **Reset_Handler**

- ① 向 **0x5000_0100** 方连续写入 0x59,0x16,0x88，实现特殊寄存器解锁。
- ② 向 **0x5000_0024** 写入 0x5AA5，禁止 POR 电路。
- ③ 向 **0x5000_0100** 方连续写入 0x00，实现特殊寄存器上锁。
- ④ 调用 **SystemInit** 子程序，地址位于 **0x000001d8** 本程序实际为空。
- ⑤ 跳转到 **__main**，地址位于 **0x000000c0**。

3. **__main**

- ① 调用 **__scatterload** 子程序，地址位于 **0x000000c8**。

__scatterload_rt2				
__scatterload_rt2_thumb_only				
0x000000c8:	a00c	..	ADR	r0,{pc}+0x34;0xfc
0x000000ca:	c830	0.	LDM	r0!,{r4,r5}
0x000000cc:	3808	.8	SUBS	r0,r0,#8
0x000000ce:	1824	\$.	ADDS	r4,r4,r0
0x000000d0:	182d	-.	ADDS	r5,r5,r0
0x000000d2:	46a2	.F	MOV	r10,r4
0x000000d4:	1e67	g.	SUBS	r7,r4,#1
0x000000d6:	46ab	.F	MOV	r11,r5
__scatterload_null				
0x000000d8:	4654	TF	MOV	r4,r10
0x000000da:	465d]F	MOV	r5,r11
0x000000dc:	42ac	.B	CMP	r4,r5
0x000000de:	d101	..	BNE	0xe4;__scatterload_null+12
0x000000e0:	f00f832	..2.	BL	__rt_entry;0x148
0x000000e4:	467e	~F	MOV	r6,pc
0x000000e6:	3e0f	.>	SUBS	r6,r6,#0xf
0x000000e8:	cc0f	..	LDM	r4!,{r0-r3}
0x000000ea:	46b6	.F	MOV	lr,r6
0x000000ec:	2601	.&	MOVS	r6,#1
0x000000ee:	4233	3B	TST	r3,r6
0x000000f0:	d000	..	BEQ	0xf4;__scatterload_null+28
0x000000f2:	1afb	..	SUBS	r3,r7,r3
0x000000f4:	46a2	.F	MOV	r10,r4
0x000000f6:	46ab	.F	MOV	r11,r5
0x000000f8:	4333	3C	ORRS	r3,r3,r6
0x000000fa:	4718	.G	BX	r3
\$d				
0x000000fc:	00000320	...	DCD	800
0x00000100:	00000340	@...	DCD	832
\$t				
!!handler_copy				
__scatterload_copy				
0x00000104:	3a10	..	SUBS	r2,r2,#0x10

0x00000106:	d302	..	BCC	0x10e ; __scatterload_copy + 10
0x00000108:	c878	x.	LDM	r0!,{r3-r6}
0x0000010a:	c178	x.	STM	r1!,{r3-r6}
0x0000010c:	d8fa	..	BHI	__scatterload_copy ; 0x104
0x0000010e:	0752	R.	LSLS	r2,r2,#29
0x00000110:	d301	..	BCC	0x116 ; __scatterload_copy + 18
0x00000112:	c830	0.	LDM	r0!,{r4,r5}
0x00000114:	c130	0.	STM	r1!,{r4,r5}
0x00000116:	d501	..	BPL	0x11c ; __scatterload_copy + 24
0x00000118:	6804	.h	LDR	r4,[r0,#0]
0x0000011a:	600c	.'	STR	r4,[r1,#0]
0x0000011c:	4770	pG	BX	lr
0x0000011e:	0000	..	MOVS	r0,r0
!!handler_zi				
__scatterload_zeroinit				
0x00000120:	2300	.#	MOVS	r3,#0
0x00000122:	2400	.\$	MOVS	r4,#0
0x00000124:	2500	.%	MOVS	r5,#0
0x00000126:	2600	.&	MOVS	r6,#0
0x00000128:	3a10	.:	SUBS	r2,r2,#0x10
0x0000012a:	d301	..	BCC	0x130 ; __scatterload_zeroinit + 16
0x0000012c:	c178	x.	STM	r1!,{r3-r6}
0x0000012e:	d8fb	..	BHI	0x128 ; __scatterload_zeroinit + 8
0x00000130:	0752	R.	LSLS	r2,r2,#29
0x00000132:	d300	..	BCC	0x136 ; __scatterload_zeroinit + 22
0x00000134:	c130	0.	STM	r1!,{r4,r5}
0x00000136:	d500	..	BPL	0x13a ; __scatterload_zeroinit + 26
0x00000138:	600b	.'	STR	r3,[r1,#0]
0x0000013a:	4770	pG	BX	lr

- ② 跳转到 **__rt_entry**，地址位于 **0x00000148**。
- ③ 调用 **__user_setup_stackheap** 子程序，地址位于 **0x000003b8**。实现用户堆栈初始化。
- ④ 调用 **__user_setup_stackheap** 子程序，地址位于 **0x000003b8**。实现用户堆栈初始化。
- ⑤ 调用 **__rt_lib_init** 子程序，地址位于 **0x0000013c**。实现驱动库初始化。本程序实际为空。
- ⑥ 在 **__rt_entry_main**，地址位于 **0x00000152** 处跳转到 **main**，地址位于 **0x00000254**。
- ⑦ 在 **main** 处开始执行真正用户自己的程序。本程序为空循环。