

EXTRACTION

Datasheet provides the memory mappings

Debugger/JLink to read to a file

Load into IDA

Draw the rest of the owl



Resource	Start Address	End Address	Size (kB)	PD	AMBA	Comments
Remapped Devices	0	800000	8192	PD_SYS	AHB	Remap IVT into SYSRAM
SYSRAM (code)	800000	880000	512	PD_MEM	AHB	Remapped at 0x0. DA14691 end address: 0x860000
Reserved						
ROM	900000	920000	128	PD_SYS	AHB	Remapped at 0x0

```
J-Link>savebin c:\users\chris\bootrom.bin, 0x900000, 0x20000  
Opening binary file for writing... [c:\users\chris\bootrom.bin]  
Reading 131072 bytes from addr 0x00900000 into file...O.K.
```

Library function Regular function Instruction Data Unexplored External symbol Lumina

Functions	IDB View-A
Function name	^ Start
Booter_Flow	000015E8
CLK_Enable_RC32M	0000155C
CLK_Set_source_xtal32m	0000159C
CLK_Switch_to_RC32M	00000344
CLK_Switch_to_XTAL32M	00000384
CRC_get_crc16_ccitt	0000146C
CRYPTO_is_processing_data	000014FC
CRYPTO_setup_data_and_st...	00005D98
CRYPTO_setup_data_locatio...	000014B8
CRYPTO_waiting_for_input	00001520
CRYPT_process_last_block	00005E34
Cache_setup_qspi_cache	00001ADC
ConfigurationScript_Read	00000228
ConfigurationScript_Read_O...	000001E0
ConfigurationScript_Read_Q...	00002026
Crypto_Validate_EdDSA	00005E68
Crypto_hash_sha512_setup	00005D5C
Crypto_setup_sha512_start...	00005F94
DeviceAdministration_KeyTy...	00000C94
ImageHeader_version_check	000065F4
NVIC_ICER0_clear_b16	0000250C

```
19 CLK_Enable_RC32M();
20 CRG_TOP_CLK_AMBA_REG[0] = 0; // reset
21 SYS_WDOG_WATCHDOG_CTRL_REG = 6; // reset
22 v0 = CRG_TOP_PMU_CTRL_REG;
23 CRG_TOP_PMU_CTRL_REG = v0 & 0xFFFFFFFF7; // clear
24 do
25     ptr_sys_stat_reg = CRG_TOP_SYS_STAT_REG;
26     while ( (ptr_sys_stat_reg & 0x800) == 0 ); // spin
27     GPIO_P0_08_MODE_REG = 0x200; // open
28     GPIO_P0_08_MODE_REG = 0x100; // push-
29     GPIO_P0_08_MODE_REG = 0x200; // // open
30     WDOG_feed_ff();
31     ptr_pmu_ctrl_reg = CRG_TOP_PMU_CTRL_REG;
32     CRG_TOP_PMU_CTRL_REG = ptr_pmu_ctrl_reg & 0xFFFFFFFFFB; // clear
33 do
34     ptr_sys_stat_reg_ = CRG_TOP_SYS_STAT_REG;
35     while ( (ptr_sys_stat_reg_ & 0x200) == 0 ); // spin
36     ptr_power_ctrl_reg = CRG_TOP_POWER_CTRL_REG;
37     CRG_TOP_POWER_CTRL_REG = ptr_power_ctrl_reg & 0xFF8FFF;
38     CRG_TOP_POWER_CTRL_REG = ptr_power_ctrl_reg & 0xFF8FFF;
39     ptr_pmu_ctrl_reg_ = CRG_TOP_PMU_CTRL_REG;
40     CRG_TOP_PMU_CTRL_REG = ptr_pmu_ctrl_reg_ & 0xFFFFFFF;
41 do
42     ptr_sys_stat_reg__ = CRG_TOP_SYS_STAT_REG;
43     while ( (ptr_sys_stat_reg__ & 8) == 0 );
44     OTPC_enable_clock_and_reset(1);
45     OTPC_set_read_mode();
46     QSPIC_set_manual_mode();
47     QSPIC_Software_Reset_peripheral(); // reset
48     LORNTF(configuration_script);
```

The screenshot shows the Immunity Debugger interface with the assembly pseudocode for a function named `Start`. The assembly window is titled "Pseudocode-A". The code implements a loop that repeatedly calls `sub_1544` until the memory at `0x50000028` has a value of 0x200. It then performs bit manipulation on memory at `0x500000F0` and `0x50000020`, and calls `sub_1CAC` with argument 1, followed by `sub_1DAC`, `sub_2052`, and `sub_20B6`. Finally, it sets memory at `0x2003C954` to 1, `0x2003C958` to 0, and `0x50020904` to 1u, and calls `sub_264C` with argument `0x1106`.

Function name	Start
<code>sub_15E8</code>	<code>000015E8</code>
<code>sub_18AC</code>	<code>000018AC</code>
<code>sub_195C</code>	<code>0000195C</code>
<code>sub_1ADC</code>	<code>00001ADC</code>
<code>sub_1C00</code>	<code>00001C00</code>
<code>sub_1C34</code>	<code>00001C34</code>
<code>sub_1C6C</code>	<code>00001C6C</code>
<code>sub_1CAC</code>	<code>00001CAC</code>
<code>sub_1DAC</code>	<code>00001DAC</code>
<code>sub_1DE8</code>	<code>00001DE8</code>
<code>sub_1E20</code>	<code>00001E20</code>
<code>sub_1EC4</code>	<code>00001EC4</code>
<code>sub_1EE0</code>	<code>00001EE0</code>
<code>sub_1EF4</code>	<code>00001EF4</code>
<code>sub_1F08</code>	<code>00001F08</code>
<code>sub_1F46</code>	<code>00001F46</code>
<code>sub_1FA2</code>	<code>00001FA2</code>
<code>sub_2026</code>	<code>00002026</code>

```
v0 = sub_155C();
MEMORY[0x50000000] = 0;
MEMORY[0x50000704] = 6;
MEMORY[0x50000020] &= ~8u;
while ( (MEMORY[0x50000028] & 0x800) == 0 )
;
MEMORY[0x50020A38] = 0x200;
sub_1544(v0);
MEMORY[0x50000020] &= ~4u;
while ( (MEMORY[0x50000028] & 0x200) == 0 )
;
MEMORY[0x500000F0] = MEMORY[0x500000F0] & 0xFF8FFFFF | 0x400000;
MEMORY[0x500000F0] |= 0x80u;
MEMORY[0x50000020] &= ~1u;
while ( (MEMORY[0x50000028] & 8) == 0 )
;
sub_1CAC(1);
sub_1DAC();
sub_2052();
sub_20B6(v1);
MEMORY[0x2003C954] = 1;
MEMORY[0x2003C958] = 0;
MEMORY[0x50020904] |= 1u;
v2 = sub_264C(0x1106);
MEMORY[0x50020A3C] = 2;
MEMORY[0x50020A38] = 1;
```

ANALYSIS CHALLENGES

Dense code

- No strings
- No symbols
- No debug statements/printf-like functions
- No external libraries

Strings			
Address	Length	Type	String
ROM:00002...	00000006	C	\n \vJ\vK
ROM:00007...	00000020	C	SigEd25519 no Ed25519 collisions
Line 1 of 2			



Function name	Start
f sub_100	00000100
f sub_1E0	000001E0
f sub_228	00000228
f sub_32C	0000032C
f sub_344	00000344
f sub_384	00000384
f sub_3C4	000003C4
f sub_3E0	000003E0
f sub_3F4	000003F4
f sub_418	00000418
f sub_4D0	000004D0