

BootROM load miniloader , and miniloader load u-boot and threadx.

in miniloader/source/main/code_images.c

```
1.  error_type_t load_code_images( rom_types_e rom_type, uint32_t *AP_loaded_code_lau
2.  {
3.      error_type_t ret_val = FAIL;
4.      void      *image_info;
5.      uint32_t flash_addr;
6.      uint32_t dest_addr;
7.      uint32_t load_size;
8.
9.      if (STATUS_OK == find_code_image("OSLO", &image_info, &flash_addr, &dest_addr
10. , &load_size)) ①
11.      {
12.          dbg_printf("\nLoading 'OSLO' @ flash_addr: %08x dest_addr: %08x load_size
13. : %08x\n", flash_addr, dest_addr, load_size);
14.          *AP_loaded_code_launch_addr = dest_addr;
15.
16.          ret_val = read_flash(rom_type, flash_addr, dest_addr, load_size,
17. hwGetSDMMCBusSpeed()); ②
18.          dbg_hex_dump((void*)dest_addr, 64);
19.
20.          if (STATUS_OK == ret_val)
21.          {
22.              if (signed_code_mode() && (false == validate_code_signatures(rom_type
23. , image_info)))
24.              {
25.                  load_wfi_loop(dest_addr, load_size);
26.              }
27.
28.              ret_val = handle_R4_mp_load(rom_type, R4_loaded_code_launch_addr);
29.              ③
30.          }
31.      }
32.
33.      return ret_val;
34.  }
```

rom_type = eROM_Ge2_RevA

①

```
error_type_t find_code_image(char *image_name, void **image_info, uint32_t *flash_addr, uint32_t *dest_addr, uint32_t *load_size)
```

find_code_image()用于查找image_name所指示的binary

```
1.  uint32_t *TIMH = *((uint32_t **) (((eROM_RevA == rom_type) || (eROM_RevA_bSPI == rom_type)) ? TIMH_BASE_ADDR_FIELD-4 : TIMH_BASE_ADDR_FIELD));
```

由于这里rom_type = eROM_Ge2_RevA,所以TIMH = TIMH_BASE_ADDR_FIELD = 0xD100003C

in miniloader/source/main/code_images.c

```
#define TIMH_BASE_ADDR_FIELD (0xD100003C)
```

而0xD100,0000 - 0xD101,7FFF是BootROM的data + stack所在。很显然BootROM把NTIM.txt中相关信息 (是否是NTIM_GR2.bin) 载入到了该地址。

然后就是根据NTIM中的信息取出对应bin的在SD card partition上的offset (NTIM.txt中的Flash Entry Address) , load address, bin size。

②

有了①中的所有信息就可以从sd card上读出bin并载入到指定的physical address。

③

载入R4运行的threadx bin。由于threadx.bin打包了所有支持的SoC(Granite2 and Gemstone2)及不同revision的bin , 所有还有一个读取当前SoC即revision来提取对应的threadx bin的过程。

