

领芯 32 位芯片如何固定从 MainFlash 启动

1，确认芯片 BOOT 模式

1.1，启动模式

2.5 启动模式

启动时，通过 boot0pin 和 boot selector option bit 选择三种启动模式：

- 从 Main Memory 启动
- 从 System Memory 启动
- 从 SRAM 启动

Boot Loader 在 System memory 里，可以通过 UART/I2C/SSP 通信接口对 Flash 进行在应用编程和升级。

三种启动模式的配置方式如下：

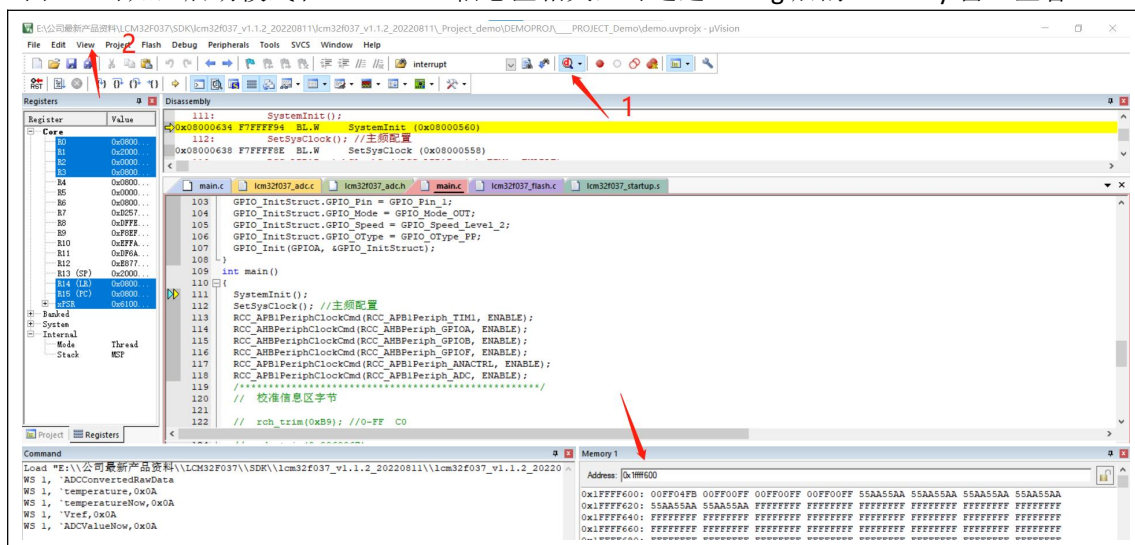
表 2-2 启动模式选择配置

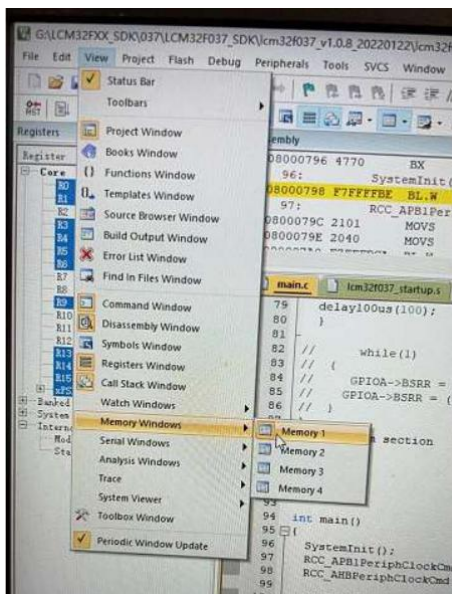
Boot 模式配置				模式
nBOOT1 bit	BOOT0 pin	BOOT_SEL bit	nBOOT0 bit	
x	0	1	x	Main Flash memory 启动
1	1	1	x	System memory 启动
0	1	1	x	SRAM 启动
x	x	0	1	Main Flash memory 启动
1	x	0	0	System memory 启动
0	x	0	0	SRAM 启动

其中 nBOOT1 bit、BOOT_SEL bit、nBOOT0 bit 存放于选项配置字节（Flash 空间 0x1FFF_F600），在复位时载入到寄存器 FLASH_OBR 中，用户可以根据需求通过烧录设备预先设置好相关的 BOOT 模式。

1.2，查看启动模式

由 1.1 可知，启动模式和 0x1ffff600 信息区相关，可通过 debug 后的 memory 窗口查看。

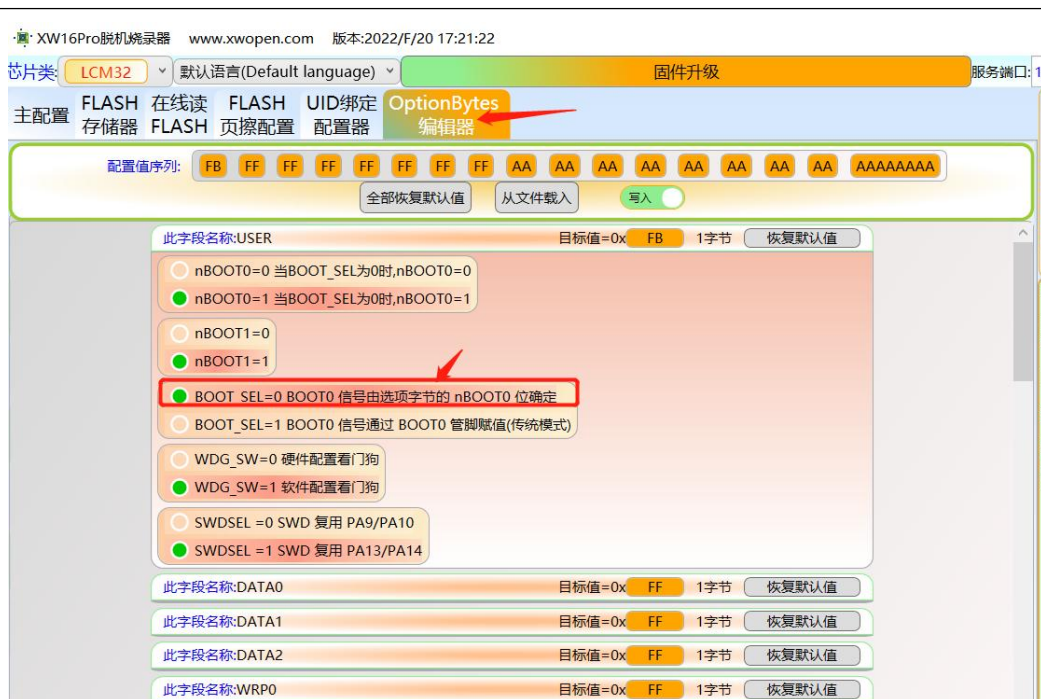




启动类型	0X1FFFF600(如果是 038 或者 03x 地址 0x08010000)
启动由外部 BOOT 脚 PF3 决定	0x00FF00FB 或者 0xFFFFFFFF
固定从 MAINflash 区启动	0x00FF04FB

1.3 如何改写此地址信息区

1.3.1 通过第三方烧录器(轩微)，通过 SWD 直接烧录



1.3.2 通过芯片程序 IAP 方式写入：参考 SDK 例程

名称	修改日期	类型	大小
Project_demo	2022/1/19 14:17	文件夹	
ACMP	2022/1/19 17:38	文件夹	
Acmp1_Opa0_Timbk_OPA_ADC_DMA	2022/1/21 14:01	文件夹	
ADC	2021/8/31 10:58	文件夹	
complex	2022/1/21 13:57	文件夹	
DAC	2021/8/31 10:58	文件夹	
DIV	2021/8/31 12:05	文件夹	
EEPROM_2Page	2022/4/25 17:11	文件夹	
GPIO	2022/1/20 10:38	文件夹	
IIC	2022/1/17 16:13	文件夹	
LVD_LVR	2022/1/21 14:01	文件夹	
OPA	2022/1/21 14:01	文件夹	
TIM1	2022/1/21 13:56	文件夹	
TIM3	2021/8/31 10:58	文件夹	
TIM6	2021/8/31 10:58	文件夹	
TIM14	2021/8/31 10:58	文件夹	
TIM15	2021/11/15 14:16	文件夹	
UART	2021/8/31 10:58	文件夹	
Write_Optionbyte	2022/4/25 11:04	文件夹	
使用说明.docx	2020/10/22 12:37	Microsoft Word ...	14 KB

```
128 // opa1_trim1(0x10);
129 // opa2_trim1(0x10);
130
131 clkpout_gpio_cfg();
132 mco_set(3, 3); //系统时钟8分频
133
134 chipctrl_access();
135 CHIPCTRL->PWR_CFG = 0x000C0001; // vrh=2.5v PA0 2.5
136 __dekey();
137
138 // ERASE_SYSTEM_MERMORY(); //擦除信息区 等校准字节
139 // TRIM_INIT(); //固化信息区 各个校准字节
140 /*****
141
142 //烧死 boot0 ,固定从mainflash启动 PT3下拉无效
143 ERASE_OPT(); //擦除选项字节 会恢复度保护
144 write_opt_flash_n(0x1ffff600,0x00ff04fb);//14eb写PA9 PA10 SWD; 0X00FF04FB 是 pal3PA14 且烧死boot
145 write_opt_flash_n(0x1ffff610,0X55AA55AA);
146 write_opt_flash_n(0x1ffff614,0X55AA55AA);
147 write_opt_flash_n(0x1ffff618,0X55AA55AA);
148 write_opt_flash_n(0x1ffff61c,0X55AA55AA);
149 write_opt_flash_n(0x1ffff620,0X55AA55AA);
150
151 while (1)
152 {
153     GPIO_SetBits(GPIOA, GPIO_Pin_1);
154     delay(2000);
155     GPIO_ResetBits(GPIOA, GPIO_Pin_1);
156     delay(2000);
157 }
158
159
```

此方式是程序自写，下载此程序，**PF3** 拉低，确保此程序正常运行。运行结束后使用 **DEBUG** 确认数据。