

《嵌入式系统原理与实践》作业

10213903403 岳锦鹏
10225001410 朱宇笑

2024 年 11 月 3 日

目录

第三次实验

准备

端口修改

代码修改

思考题

实验结果

第三次实验

准备

端口修改

代码修改

思考题

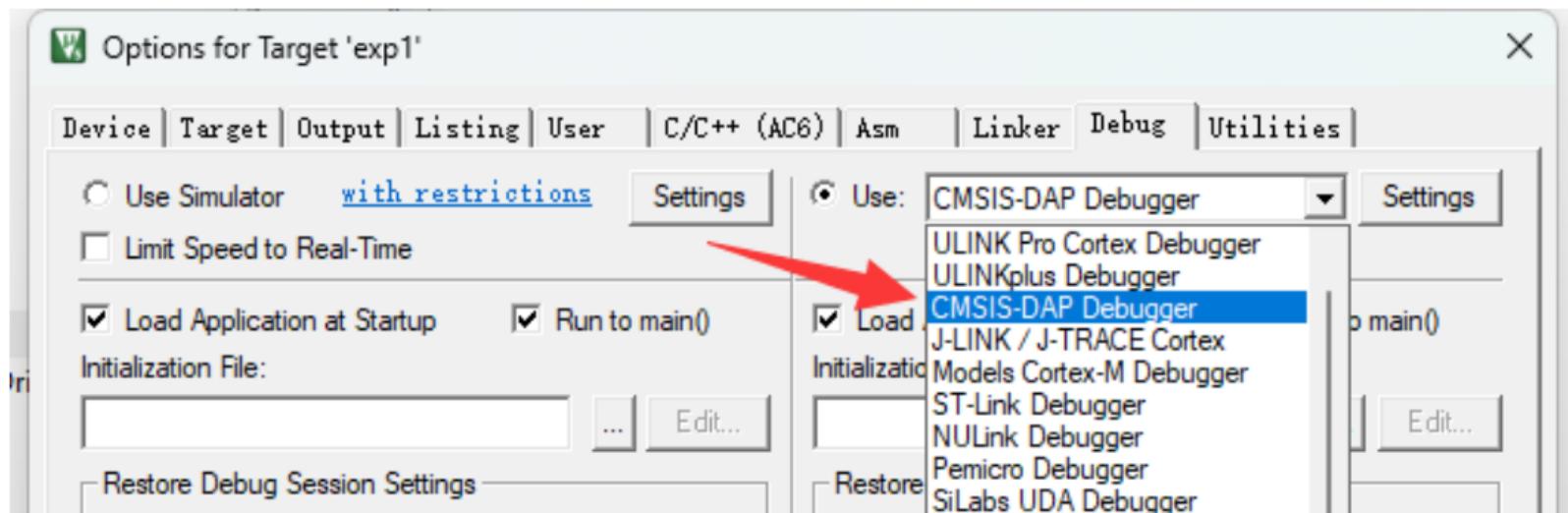
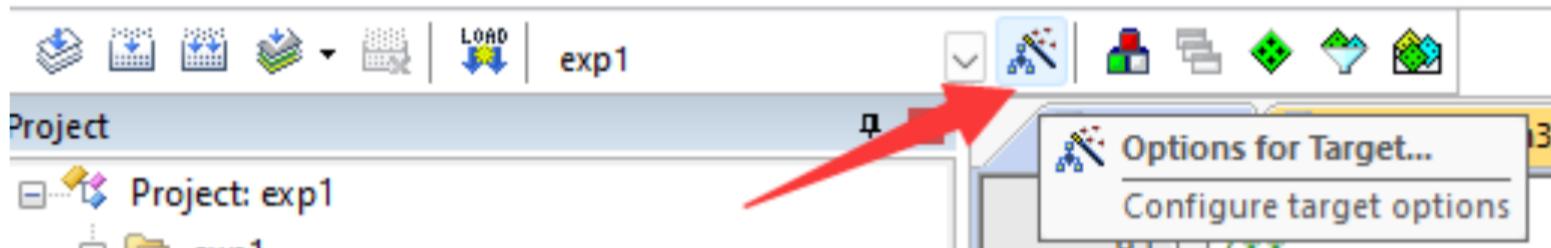
实验结果

将实验二的代码复制一份，并将 exp2 改名为 exp3:

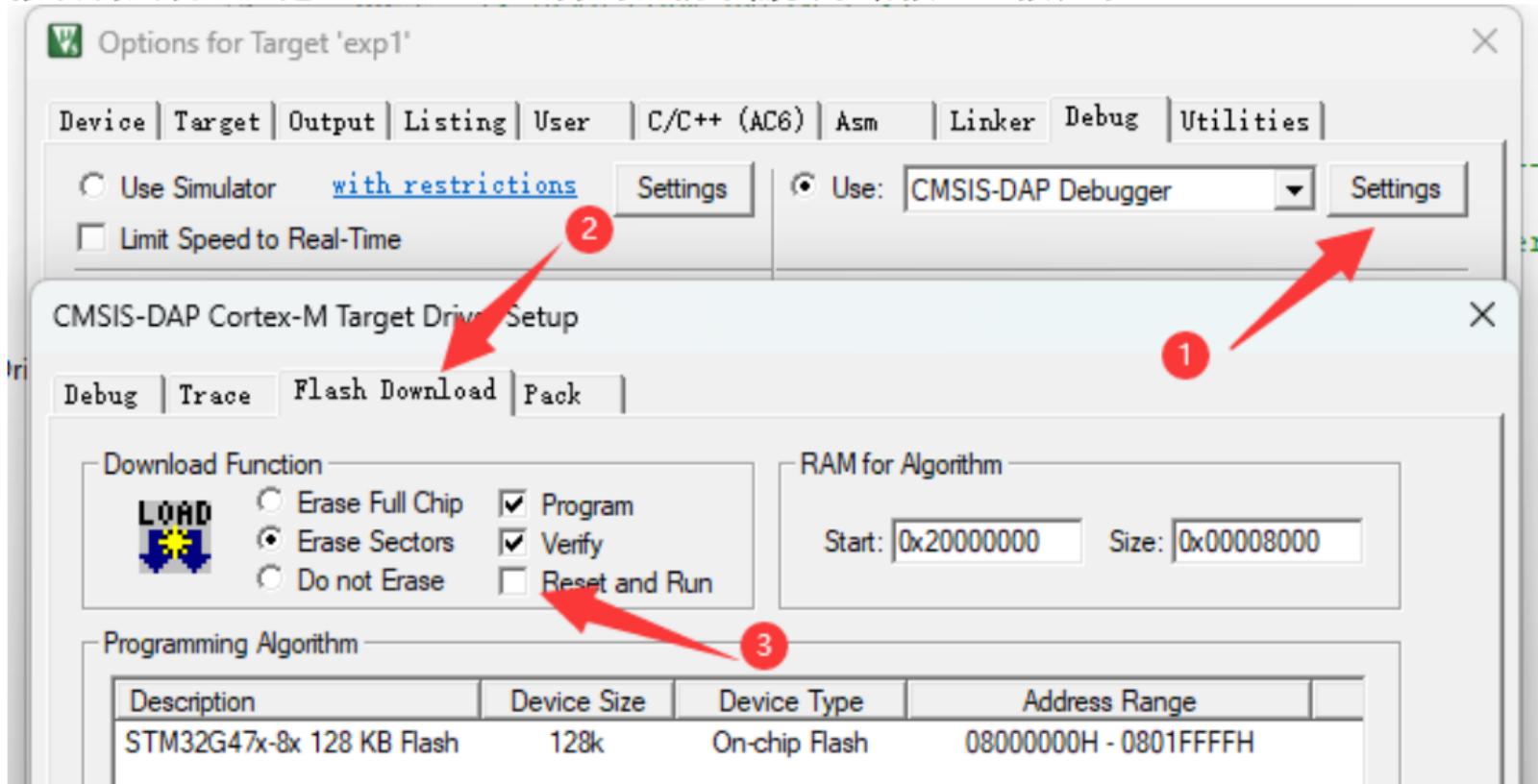
exp2		exp3
.mxproject		.mxproject
exp2.ioc		exp3.ioc
Core	→	Core
Drivers		Drivers
MDK-ARM		MDK-ARM
exp2.uvprojx		exp3.uvprojx
startup_stm32g473xx.s		startup_stm32g473xx.s

需要改文件夹名、.ioc、.uvprojx 这三个地方。

打开.uvprojx 文件，修改调试器：



修改调试设置，把 reset and run 打勾，就不需要手动按重置按钮了。



第三次实验

准备
端口修改
代码修改
思考题
实验结果

这样设置端口，点击生成代码。

Software Packs Pinout

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO RCC SYS

Search Signals

Search (Ctrl+F) Show only Modified Pins

Pin	Signal	GPIO o.	GPIO m.	GPIO P.	Maximu.	Fast Mo.	User La.	Modified
PA0	n/a	High	Output ...	Pull-up	Low	n/a	SegLed...	<input checked="" type="checkbox"/>
PA1	n/a	High	Output ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA2	n/a	High	Output ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA3	n/a	High	Output ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA4	n/a	High	Output ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA5	n/a	High	Output ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA6	n/a	High	Output ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA7	n/a	High	Output ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PB0	n/a	Low	Output ...	Pull-up	Low	n/a	AddrA	<input checked="" type="checkbox"/>
PB1	n/a	Low	Output ...	Pull-up	Low	n/a	AddrB	<input checked="" type="checkbox"/>
PB2	n/a	Low	Output ...	Pull-up	Low	n/a	AddrC	<input checked="" type="checkbox"/>
PB3	n/a	High	Output ...	Pull-up	Low	n/a	OE	<input checked="" type="checkbox"/>
PC13	n/a	Low	Output ...	Pull-up	Low	n/a	LED	<input checked="" type="checkbox"/>

Select Pins from table to configure them. Multiple selection is Allowed.

Pinout view System view

第三次实验

准备
端口修改
代码修改
思考题
实验结果

main.c:

```
45  /* USER CODE BEGIN PV */
46  stSysTickTimer sSysTickTimer = {
47      0, 0, 0, 0
48  };
49  uint16_t display_tab[] = {
50      0x3f,
51      0x06,
52      0x5b,
53      0x4f,
54      0x66,
55      0x6d,
56      0x7d,
57      0x07,
58      0x7f,
59      0x6f,
60      0x77,
61      0x7c,
62      0x39,
63      0x5e,
64      0x79,
65      0x71
66  };
67  uint16_t PosSel = 0;
68  /* USER CODE END PV */
```

main.c:

```
117  /* USER CODE BEGIN WHILE */
118  while (1)
119  {
120      /* USER CODE END WHILE */
121
122      /* USER CODE BEGIN 3 */
123      if (sSysTickTimer.bTimeOk) {
124          sSysTickTimer.bTimeOk = 0;
125          HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
126      }
127      // if (sSysTickTimer.bTenMilSecOk) {
128
129          SegLedData_GPIO_Port->ODR = 0x00;
130          AddrA_GPIO_Port->ODR = PosSel;
131          SegLedData_GPIO_Port->ODR = display_tab[PosSel %
132          ↵ 8];
133          if (++PosSel >= 8) {
134              PosSel = 0;
135          }
136          // }
137      }
138      /* USER CODE END 3 */
```

第三次实验

准备
端口修改
代码修改
思考题
实验结果

我们的需求是将数字一位一位向前推进，那么考虑现在的位选信号（控制哪一位上显示数字）与段选信号（控制显示哪个数字）之间的关系，根据 `SegLedData_GPIO_Port->ODR = display_tab[PosSel % 8];` 可知

$$\text{段选信号}_0 = f(\text{位选信号}_0)$$

而修改后就变成了（时间用 t 表示）

$$\text{段选信号}_t = g(\text{位选信号}_t, t)$$

因此，只需要加一个随时间变化的变量，例如 $bias = bias(t)$ ，再把之前的 f 看做是时间为 0 时的静止状态，则可以得到

$$\begin{cases} \text{段选信号}_0 = g(\text{位选信号}_0, 0) = h(\text{位选信号}_0, bias(0)) = f(\text{位选信号}_0) \\ \text{段选信号}_t = g(\text{位选信号}_t, t) = h(\text{位选信号}_t, bias(t)) \end{cases}$$

其中的 h 和 $bias$ 就是我们需要构造的函数和变量。

确定 $bias(0)$ 的值:

```
68  uint8_t bias = 0;
```

在 bTimeOk 时改变 $bias$ 的值, 使 $bias$ 随时间变化:

```
128     if (++bias >= 8) {  
129         bias = 0;  
130     }
```

构造 $h(\text{段选信号}, bias)$ 使得 $h(\text{段选信号}, 0) = f(\text{段选信号})$

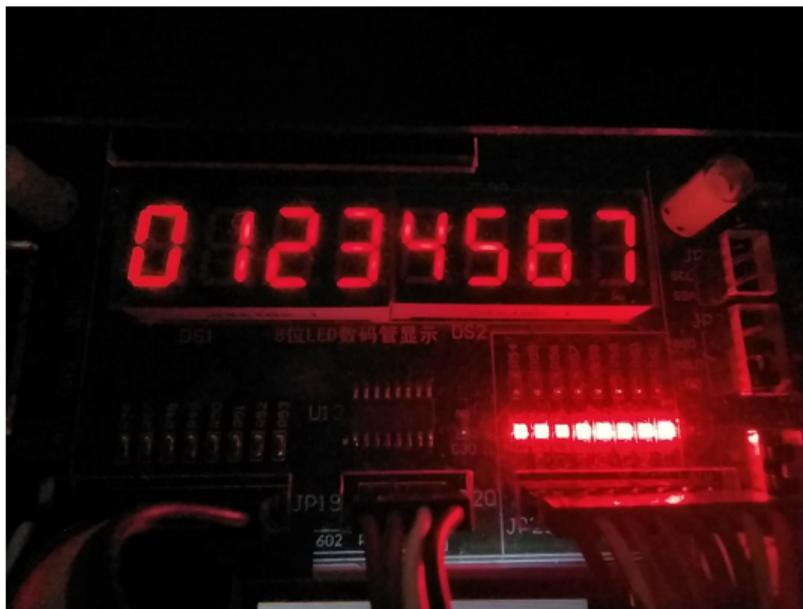
```
136     SegLedData_GPIO_Port->ODR = display_tab[(PosSel + bias) % 8];
```

第三次实验

准备
端口修改
代码修改
思考题
实验结果

初始显示 01234567，每过一秒，数字向左循环移位一位，即

01234567 → 12345670 → 23456701 → ... → 70123456 → 01234567



完整视频可以查看：

https://gitea.librastalker.top/423A35C7/STM32CubeMX-Keil_uVision5