

《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

第四次实验

准备

端口修改

代码修改

实验结果

第四次实验

准备

端口修改

代码修改

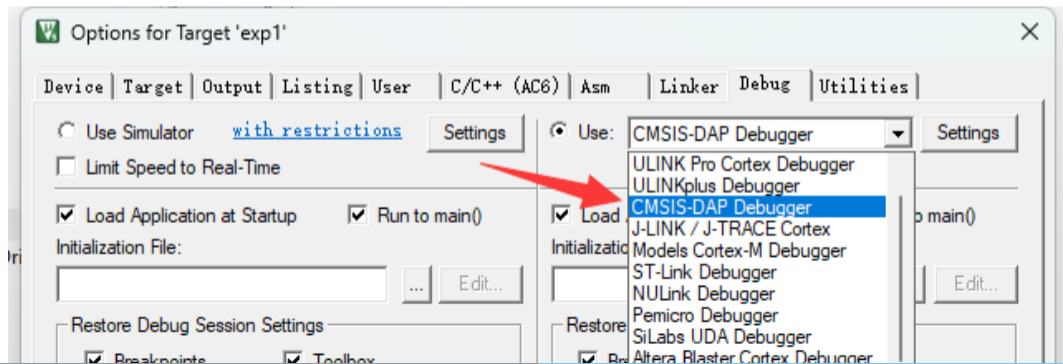
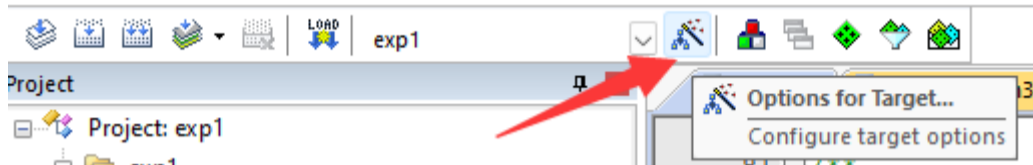
实验结果

将实验三的代码复制一份，并将 exp3 改名为 exp4:

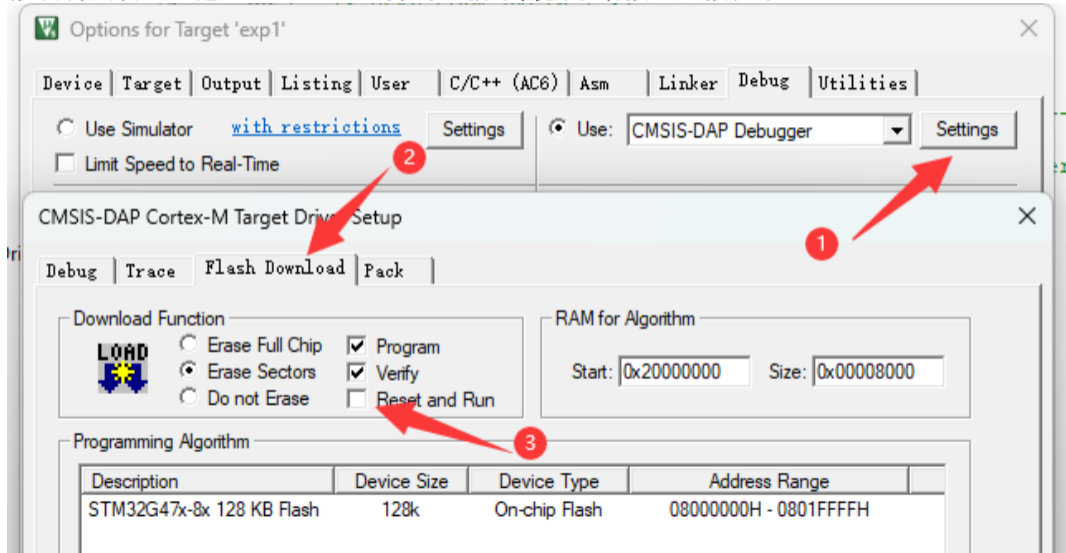
exp3		exp4
.mxproject		.mxproject
exp3.ioc		exp4.ioc
Core	→	Core
Drivers		Drivers
MDK-ARM		MDK-ARM
exp3.uvprojx		exp4.uvprojx
startup_stm32g473xx.s		startup_stm32g473xx.s

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。



第四次实验

准备
端口修改
代码修改
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

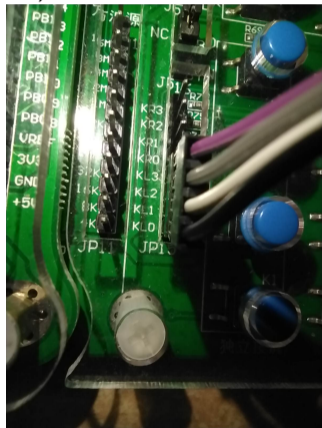
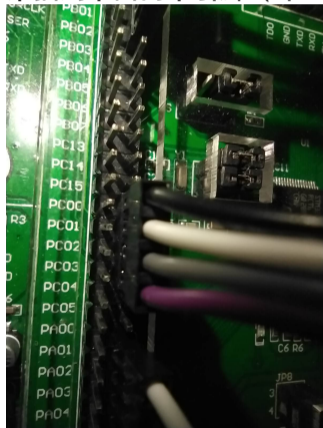
Search (Ctrl+F) Show only Modified Pins

Pin Name	Signal on Pin	GPIO output I	GPIO mode	GPIO Pull-up/	Maximum out	Fast Mode	User Label	Modified
PA0	n/a	High	Output Push ...	Pull-up	Low	n/a	SegLedData	<input checked="" type="checkbox"/>
PA1	n/a	High	Output Push ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA2	n/a	High	Output Push ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA3	n/a	High	Output Push ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA4	n/a	High	Output Push ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA5	n/a	High	Output Push ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA6	n/a	High	Output Push ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA7	n/a	High	Output Push ...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PB0	n/a	Low	Output Push ...	Pull-up	Low	n/a	AddrA	<input checked="" type="checkbox"/>
PB1	n/a	Low	Output Push ...	Pull-up	Low	n/a	AddrB	<input checked="" type="checkbox"/>
PB2	n/a	Low	Output Push ...	Pull-up	Low	n/a	AddrC	<input checked="" type="checkbox"/>
PB3	n/a	High	Output Push ...	Pull-up	Low	n/a	OE	<input checked="" type="checkbox"/>
PC0	n/a	n/a	Input mode	Pull-up	n/a	n/a	Key1	<input checked="" type="checkbox"/>
PC1	n/a	n/a	Input mode	Pull-up	n/a	n/a	Key5	<input checked="" type="checkbox"/>
PC2	n/a	n/a	Input mode	Pull-up	n/a	n/a	Key9	<input checked="" type="checkbox"/>
PC3	n/a	n/a	Input mode	Pull-up	n/a	n/a	Key13	<input checked="" type="checkbox"/>

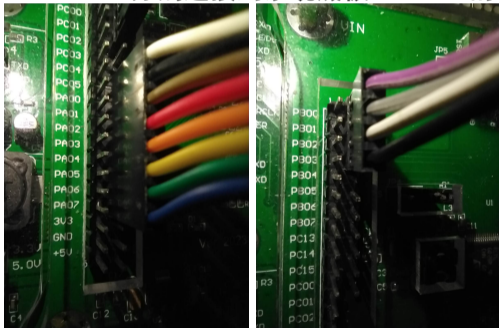
Select Pins from table to configure them. Multiple selection is Allowed.

Pinout view System view

按键：(Key1、Key5、Key9 和 Key13 的四个按键的一个端子与 JP15 的 KL0~KL3 相连。) 核心板 JP1 的 PB0~PB3 接底板 JP15 的 KL0~KL3。将 JP16 上面两个引脚的跳线帽拔下来插到下面两个引脚 (即 BTN)。



数码管：将 STM32 核心板 JP1 的 PA0~PA7 与系统的七段数码管电路的 JP19 相连，PA0~PA7 分别对应于 JP19 的 D0~D7，用 1x4 的杜邦线将 STM32 核心板的 JP2 的 PB0~PB3 分别连接到系统底板上 JP20 的 A、B、C 和 OE 接口。



第四次实验

准备
端口修改
代码修改
实验结果

添加 DirectKey.h:

```
1  #ifndef __DIRECTKEY_H
2  #define __DIRECTKEY_H
3
4  #include<stdint.h>
5
6  enum {State0=0, State1, State2};
7
8  #define NoKey 0x0F
9  #define KeyPort GPIOC
10
11
```

```
12 #ifdef __cplusplus
13 extern "C" {
14 #endif
15
16 uint8_t ReadDirectKey(void);
17
18 #ifdef __cplusplus
19 }
20 #endif
21
22 #endif
```

添加 DirectKey.c:

```
1  #include "directkey.h"
2  #include "main.h"
3
4
5  uint8_t ReadDirectKey(void) {
6      static uint8_t KeyState = State0;
7      static uint8_t KeyOld;
8      uint8_t KeyPress;
9      uint8_t KeyValue = NoKey;
10
11     KeyPress = KeyPort->IDR & 0x000F; // 读按键
12
13     switch (KeyState) {
14     case State0:
15         if (KeyPress != NoKey) {
16             KeyOld = KeyPress; // 保存原来的键
17             KeyState = State1; // 切换状态
18         }
19         break;
20
21     case State1:
22
23         if (KeyPress == KeyOld) {
24             KeyState = State2;
25             // KeyValue = KeyOld; // 按下有效
26         } else {
27             KeyState = State0;
28         }
29         break;
30
31     case State2:
32         if (KeyPress == NoKey) {
33             KeyState = State0;
34             KeyValue = KeyOld; // 抬起有效
35         } else {
36             break;
37         }
38
39     default:
40         break;
41     }
42     return KeyValue;
43 }
```

stm32g4xx_it.c:

```
182  /**
183     * @brief This function handles System tick timer.
184     */
185  void SysTick_Handler(void)
186  {
187     /* USER CODE BEGIN SysTick_IRQn 0 */
188
189     /* USER CODE END SysTick_IRQn 0 */
190     HAL_IncTick();
191     /* USER CODE BEGIN SysTick_IRQn 1 */
192     Display();
193
194     if (++sSysTickTimer.mMilSecCount >= 10) {
195         sSysTickTimer.mMilSecCount = 0;
196         sSysTickTimer.bTenMilSecOk = 1;
197         if (++sSysTickTimer.mTimeCount >= 100) {
198             sSysTickTimer.mTimeCount = 0;
199             sSysTickTimer.bTimeOk = 1;
200         }
201     }
202     /* USER CODE END SysTick_IRQn 1 */
203 }
```

main.c:

```
24  /* USER CODE BEGIN Includes */
25  #include "variable.h"
26  #include "directkey.h"
27  /* USER CODE END Includes */

46  /* USER CODE BEGIN PV */
47  stSysTickTimer sSysTickTimer = {
48      0, 0, 0, 0
49  };
50  uint8_t tempValue;
51  uint16_t display_tab[] = {
52      0x3f,
53      0x06,
54      0x5b,
55      0x4f,
56      0x66,
57      0x6d,
58      0x7d,
59      0x07,
60      0x7f,
61      0x6f,
62      0x77,
63      0x7c,
64      0x39,
65      0x5e,
66      0x79,
67      0x71
68  };
69  uint8_t DispBuff[8];
70  uint16_t PosSel = 0;
71  /* USER CODE END PV */
```

main.c:

```

79  /* Private user code -----*/ 98  }
80  /* USER CODE BEGIN 0 */ 99
81  void DisplayOneLed(uint8_t dat, uint8_t pos, uint8_t 100  void TimeToBuff(void) {
    ↪ dot) { 101      uint8_t i;
82      uint16_t temp; 102      uint8_t temp;
83  103      temp = DispBuff[0];
84      temp = display_tab[dat]; 104      for (int i = 0; i < 7; i++) {
85      if (dot) 105          DispBuff[i] = DispBuff[i + 1];
86          temp |= 0x80; 106      }
87  107      DispBuff[7] = temp;
88      GPIOA->ODR &= 0xFF00; 108  }
89      GPIOA->ODR |= temp; // 数据段 109
90      GPIOB->ODR &= 0xFF0; 110  void DispToBuff(uint8_t val) {
91      GPIOB->ODR |= pos; // 位选 111      uint8_t i;
92  } 112      for (int i = 0; i < 7; i++) {
93  113          DispBuff[i] = DispBuff[i + 1];
94  void Display(void) { 114      }
95      static uint8_t mPos = 0; 115      DispBuff[7] = val;
96      DisplayOneLed(DispBuff[mPos], mPos, 0); 116  }
97      if (++mPos >= 8) mPos = 0; 117  /* USER CODE END 0 */

```


main.c:

```

119  /**
120   * @brief The application entry point.
121   * @retval int
122   */
123  int main(void)
124  {
125
126     /* USER CODE BEGIN 1 */
127     uint8_t KeyValue = 0;
128     /* USER CODE END 1 */
129
130     /* MCU Configuration-----*/
131
132     /* Reset of all peripherals, Initializes the Flash
133    ↪ interface and the Systick. */
134     HAL_Init();
135
136     /* USER CODE BEGIN Init */
137
138     /* USER CODE END Init */
139
140     /* Configure the system clock */
141     SystemClock_Config();
142
143     /* USER CODE BEGIN SysInit */
144
145     /* USER CODE END SysInit */
146
147     /* Initialize all configured peripherals */
148     MX_GPIO_Init();
149     /* USER CODE BEGIN 2 */
150     // FlashLeds_GPIO_Port->ODR &= 0xff01;
151     for (int i = 0; i < 8; i++) {
152         DispBuff[i] = i;
153     }
154     /* USER CODE END 2 */

```

main.c:

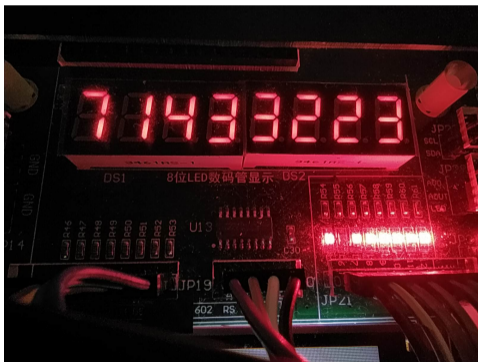
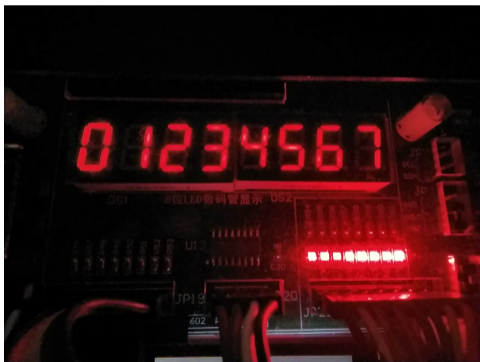
```
155  /* Infinite loop */
156  /* USER CODE BEGIN WHILE */
157  while (1)
158  {
159      /* USER CODE END WHILE */
160
161      /* USER CODE BEGIN 3 */
162      if (sSysTickTimer.bTenMilSecOk) {
163          sSysTickTimer.bTenMilSecOk = 0;
164          KeyValue = ReadDirectKey();
165          if (KeyValue != NoKey) {
166              switch (KeyValue)
167              {
168                  case 0x0E: // K1
169                      tempValue = 1;
170                      break;
171
172                  case 0x0D: // K5
173                      tempValue = 2;
174                      break;
175
176                  case 0x0B: // K9
177                      tempValue = 3;
178                      break;
179
180                  case 0x07: // K13
181                      tempValue = 4;
182                      break;
183
184                  default:
185                      break;
186              }
187              DispToBuff(tempValue);
188          }
189      }
190      if (sSysTickTimer.bTimeOk) {
191          sSysTickTimer.bTimeOk = 0;
192          // TimeToBuff();
193          HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
194      }
195  }
196  /* USER CODE END 3 */
197  }
```

第四次实验

准备
端口修改
代码修改
实验结果

数码管初始显示 01234567，按下 K1、K5、K9、K13 并抬起时，数码管左移一位，并在右侧添加 1、2、3、4。因此记 K1、K5、K9、K13 分别为 1、2、3、4。例如，逐个按下 1433223，数码管显示的数字变化为：

01234567 → 12345671 → 23456714 → ... → 71433223



完整视频可以查看：

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