

# 《嵌入式系统原理与实践》作业

10213903403 岳锦鹏

10225001410 朱宇笑

2024 年 12 月 30 日

# 目录

第五次实验  
端口修改  
代码修改  
实验结果

# 第五次实验

## 端口修改

## 代码修改

## 实验结果

## 端口设置如图。

STM32CubeMX exp5.ioc: STM32G473RCTx

File Window Help Hello Yuxiao

Home STM32G473RCTx exp5.ioc - Pinout & Configuration GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Software Packs Pinout

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO RCC SYS

Search Signals

Search (Ctrl+F)  Show only Modified Pins

Pin No.	Signal on	GPIO out	GPIO Mode	GPIO Pul.	Maximum	Fast Mode	User Label	Modified
PA0	n/a	High	Output P...	Pull-up	Low	n/a	SegLedD...	<input checked="" type="checkbox"/>
PA1	n/a	High	Output P...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA2	n/a	High	Output P...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA3	n/a	High	Output P...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA4	n/a	High	Output P...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA5	n/a	High	Output P...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA6	n/a	High	Output P...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PA7	n/a	High	Output P...	Pull-up	Low	n/a		<input checked="" type="checkbox"/>
PB0	n/a	Low	Output P...	Pull-up	Low	n/a	AddrA	<input checked="" type="checkbox"/>
PB1	n/a	Low	Output P...	Pull-up	Low	n/a	AddrB	<input checked="" type="checkbox"/>
PB2	n/a	Low	Output P...	Pull-up	Low	n/a	AddrC	<input checked="" type="checkbox"/>
PB3	n/a	High	Output P...	Pull-up	Low	n/a	OE	<input checked="" type="checkbox"/>

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

Pinout view System view

# 第五次实验

端口修改  
代码修改  
实验结果

## 添加 MatrixKey.h:

```
1  #ifndef __MATRIXKEY_H          13
2  #define __MATRIXKEY_H          14
3
4
5  #include<stdint.h>             15
6  #include "directkey.h"         16
7  // enum {State0=0, State1, State2}; 17
8  #define NO_KEY 0xFF            18
9
10 #ifdef __cplusplus              19
11 extern "C" {                    20
12                                 21
13                                 22
14                                 23
15 uint8_t MatrixKeyScan(void);
16 extern uint8_t const KeyTable[16];
17
18
19 #ifdef __cplusplus
20 }
21 #endif
22
23 #endif
```

## 添加 MatrixKey.c:

```
1  #include "MatrixKey.h"
2  #include "main.h"
3
4  // EDB7 xx EDB7
5  uint8_t const KeyTable[16] = {
6      0xEE, 0xDE, 0xBE, 0x7E,
7      0xED, 0xDD, 0xBD, 0x7D,
8      0xEB, 0xDB, 0xBB, 0x7B,
9      0xE7, 0xD7, 0xB7, 0x77,
10 };
11
12 void LineIn_RowOut(void) {
13     GPIO_InitTypeDef GPIO_InitStructure = {0};
14     /*Configure GPIO pin : PtPin */
15
16     /*Configure GPIO pins : PCPin PC1 PC2 PC3 */
17     GPIO_InitStructure.Pin = KeyLine_Pin | GPIO_PIN_1 |
18     ↵ GPIO_PIN_2 | GPIO_PIN_3;
19     GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
20     GPIO_InitStructure.Pull = GPIO_PULLUP;
21     HAL_GPIO_Init(KeyLine_GPIO_Port, &GPIO_InitStructure);
22
23     GPIO_InitStructure.Pin = KeyRow_Pin | GPIO_PIN_5 |
24     ↵ GPIO_PIN_6 | GPIO_PIN_7;
25     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
26     GPIO_InitStructure.Pull = GPIO_PULLUP;
27     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
28     HAL_GPIO_Init(KeyRow_GPIO_Port, &GPIO_InitStructure);
29 }
```

## 添加 MatrixKey.c:

```

28 void LineOut_RowIn(void) {                               45
29     GPIO_InitTypeDef GPIO_InitStructure = {0};          46
30     /*Configure GPIO pin : PtPin */                     47
31     /*Configure GPIO pins : PCPin PC1 PC2 PC3 */        48
32     GPIO_InitStructure.Pin = KeyLine_Pin | GPIO_PIN_1 | 49
↪ GPIO_PIN_2 | GPIO_PIN_3;                               50
33     GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;     51
34     GPIO_InitStructure.Pull = GPIO_PULLUP;             52
35     GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;    53
36     HAL_GPIO_Init(KeyLine_GPIO_Port, &GPIO_InitStructure);
37
38     GPIO_InitStructure.Pin = KeyRow_Pin | GPIO_PIN_5 | 54
↪ GPIO_PIN_6 | GPIO_PIN_7;
39     GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
40     GPIO_InitStructure.Pull = GPIO_PULLUP;             55
41     HAL_GPIO_Init(KeyRow_GPIO_Port, &GPIO_InitStructure);
42 }
43
44 uint8_t MatrixKeyScan(void) {                            56
                                     static uint8_t Keystate = State0;
                                     static uint8_t KeyLine, KeyRow;
                                     static uint16_t KeyOld;

                                     uint8_t KeyValue = NO_KEY;

                                     LineOut_RowIn();
                                     /*Configure GPIO pin Output Level */
                                     HAL_GPIO_WritePin(KeyLine_GPIO_Port, KeyLine_Pin |
↪ GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,
↪ GPIO_PIN_RESET); // 这四个按位或操作后是 0x0F
                                     HAL_GPIO_WritePin(KeyLine_GPIO_Port, KeyLine_Pin |
↪ GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,
↪ GPIO_PIN_RESET); // 重复三次使更稳定
                                     HAL_GPIO_WritePin(KeyLine_GPIO_Port, KeyLine_Pin |
↪ GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3,
↪ GPIO_PIN_RESET); // 刚写入后不能直接读要等稳定了
↪ 再读
                                     KeyRow = KeyRow_GPIO_Port->IDR & 0x00F0;

```



## 添加 MatrixKey.c:

```

58     LineIn_RowOut();                                75
59     HAL_GPIO_WritePin(KeyRow_GPIO_Port, KeyRow_Pin | 76
↪     GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,         77
↪     GPIO_PIN_RESET);                               78
60     HAL_GPIO_WritePin(KeyRow_GPIO_Port, KeyRow_Pin | 79
↪     GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,         80
↪     GPIO_PIN_RESET); // 这里也是重复三次        81
61     HAL_GPIO_WritePin(KeyRow_GPIO_Port, KeyRow_Pin | 82
↪     GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,         83
↪     GPIO_PIN_RESET);
62     KeyLine = KeyLine_GPIO_Port->IDR & 0x000F;    84
63                                                     85
64     switch (Keystate) {                             86
65     case State0:                                    87
66         KeyOld = KeyLine | KeyRow;                 88
67         if (KeyOld != NO_KEY) {                    89
68             Keystate = State1;                     90
69         } else {                                    91
70             Keystate = State0;                     92
71         }                                          93
72         break;                                     94
73                                                     95
74     case State1:

```

```

if ((KeyLine | KeyRow) == KeyOld) {
    Keystate = State2;
} else {
    Keystate = State0;
}
break;

case State2:
    if ((KeyLine | KeyRow) == NO_KEY) { // 键已释
↪     放
        Keystate = State0;
        KeyValue = KeyOld;
    } else {
        Keystate = State2;
    }
    break;

default:
    break;
}
return KeyValue;
}

```

## main.c:

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

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

## main.c:

```

80  /* Private user code -----*/          99  }
81  /* USER CODE BEGIN 0 */                100
82  void DisplayOneLed(uint8_t dat, uint8_t pos, uint8_t 101  void TimeToBuff(void) {
    ↪ dot) {                               102      uint8_t i;
83      uint16_t temp;                      103      uint8_t temp;
84                                          104      temp = DispBuff[0];
85      temp = display_tab[dat];            105      for (int i = 0; i < 7; i++) {
86      if (dot)                            106          DispBuff[i] = DispBuff[i + 1];
87          temp |= 0x80;                   107      }
88                                          108      DispBuff[7] = temp;
89      GPIOA->ODR &= 0xFF00;                109  }
90      GPIOA->ODR |= temp; // 数据段        110
91      GPIOB->ODR &= 0xFF00;                111  void DispToBuff(uint8_t val) {
92      GPIOB->ODR |= pos; // 位选          112      uint8_t i;
93  }                                        113      for (int i = 0; i < 7; i++) {
94                                          114          DispBuff[i] = DispBuff[i + 1];
95  void Display(void) {                    115      }
96      static uint8_t mPos = 0;            116      DispBuff[7] = val;
97      DisplayOneLed(DispBuff[mPos], mPos, 0); 117  }
98      if (++mPos >= 8) mPos = 0;          118  /* USER CODE END 0 */

```

## main.c:

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

## main.c:

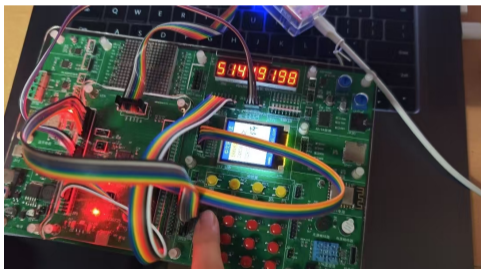
```
156     /* Infinite loop */
157     /* USER CODE BEGIN WHILE */
158     while (1)
159     {
160         /* USER CODE END WHILE */
161
162         /* USER CODE BEGIN 3 */
163         if (sSysTickTimer.bTenMilSecOk) {
164             sSysTickTimer.bTenMilSecOk = 0;
165             KeyValue = MatrixKeyScan();
166             if (KeyValue != NO_KEY) {
167                 for (int i = 0; i < 16; i++) {
168                     if (KeyValue == KeyTable[i]) {
169                         tempValue = i;
170                     }
171                 }
172                 DispToBuff(tempValue);
173             }
174         }
175         if (sSysTickTimer.bTimeOk) {
176             sSysTickTimer.bTimeOk = 0;
177             // TimeToBuff();
178             HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
179         }
180     }
181     /* USER CODE END 3 */
182 }
```

# 第五次实验

端口修改  
代码修改  
实验结果

数码管初始显示 01234567，按下  $K_x$  并抬起时，数码管左移一位，并在右侧添加  $x-1$  的十六进制数。例如，逐个按下  $K_1$ ,  $K_8$ ,  $K_3$ ,  $K_2$ ，数码管显示的数字变化为：

01234567  $\rightarrow$  12345670  $\rightarrow$  23456707  $\rightarrow$  34567072  $\rightarrow$  45670721



完整视频可以查看：

[https://gitea.librastalker.top/423A35C7/STM32CubeMX-Keil\\_uVision5](https://gitea.librastalker.top/423A35C7/STM32CubeMX-Keil_uVision5)