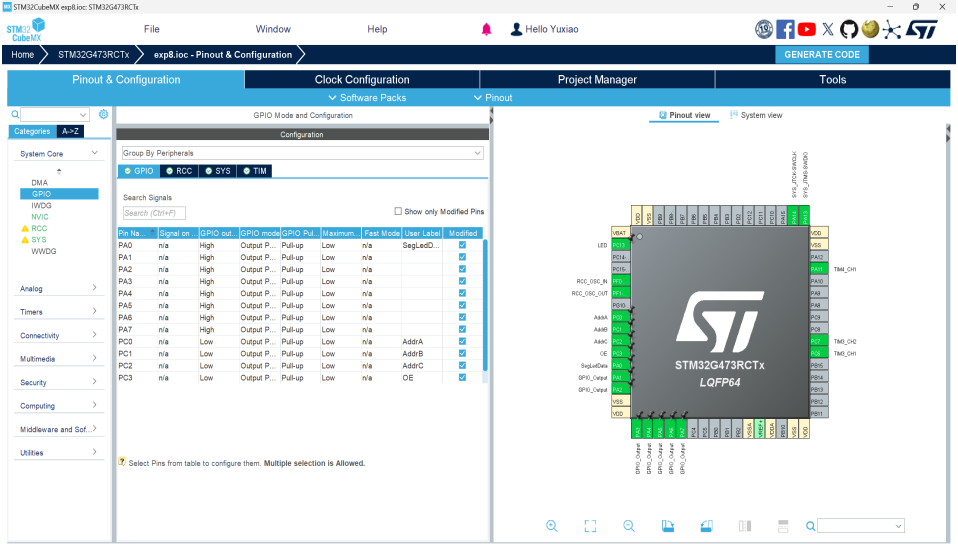
《嵌入式系统原理与实践》作业

10213903403 岳锦鹏
10225001410 朱宇笑

2024 年 12 月 30 日

目录

端口设置如图。


STM32CubeMX exp8.ioc: STM32G473RCTx
File
Window
Help
Hello Yuxiao
Home
STM32G473RCTx
exp8.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
TIM
Search Signals
Show only Modified Pins
Pin Na...
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...
PA1 n/a High Output P... Pull-up Low n/a
PA2 n/a High Output P... Pull-up Low n/a
PA3 n/a High Output P... Pull-up Low n/a
PA4 n/a High Output P... Pull-up Low n/a
PA5 n/a High Output P... Pull-up Low n/a
PA6 n/a High Output P... Pull-up Low n/a
PA7 n/a High Output P... Pull-up Low n/a
PC0 n/a Low Output P... Pull-up Low n/a AddrA
PC1 n/a Low Output P... Pull-up Low n/a AddrB
PC2 n/a Low Output P... Pull-up Low n/a AddrC
PC3 n/a Low Output P... Pull-up Low n/a OE
Select Pins from table to configure them. Multiple selection is Allowed.
Pinout view
System view
STM32G473RCTx
LQFP64

第八次实验
端口修改
代码修改

main.c:

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


38  /* USER CODE BEGIN PD */
39  typedef struct {
40    uint32_t CapValue1;
41    uint32_t CapValue2;
42    float     Duty;
43    float     Frequency;
44    uint8_t   CapIsFinish;
45  } stPWMCap;
46  stPWMCap sPwmCap = {0};
47  /* USER CODE END PD */
```

```
56  /* USER CODE BEGIN PV */
57  stSysTickTimer sSysTickTimer = {
58    0, 0, 0, 0
59  };
60  uint8_t tempValue;
61  structTime stTime = {
62    .mSecond = 50,
63    .mMinute = 45,
64    .mHour = 8,
65    .mTimeCount = 0,
66    .bSecondIsOk = 0,
67    .mTenMilCount = 0,
68    .bTenMilIsOk = 0
69  };
70  /* USER CODE END PV */
```

main.c:

```c
/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{

  /* USER CODE BEGIN 1 */
  uint8_t KeyValue = 0;
  /* USER CODE END 1 */

  /* MCU Configuration--------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_TIM3_Init();
  MX_TIM4_Init();
  /* USER CODE BEGIN 2 */
  // FlashLeds_GPIO_Port->ODR &= 0xff01;
  // HAL_TIM_Base_Start(&htim4);
  HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_1);
  // HAL_TIM_Base_Start_IT(&htim3);
  HAL_TIM_IC_Start_IT(&htim3, TIM_CHANNEL_1);
  HAL_TIM_IC_Start_IT(&htim3, TIM_CHANNEL_2);
  TimeToBuff(&stTime);
  /* USER CODE END 2 */
```

main.c:

```
124     /* Infinite loop */
125     /* USER CODE BEGIN WHILE */
126     while (1)
127     {
128       /* USER CODE END WHILE */
129
130       /* USER CODE BEGIN 3 */
131       if (sPwmCap.CapIsFinish) {
132         DispToBuff(sPwmCap.CapValue2 + 1);
133         sPwmCap.CapIsFinish = 0;
134         sPwmCap.CapValue1 = 0;
135         sPwmCap.CapValue2 = 0;
136         sPwmCap.Duty = 0;
137         sPwmCap.Frequency = 0;
138       }
139       if (stTime.bTenMilIsOk) {
140         stTime.bTenMilIsOk = 0;
141       }
142       if (stTime.bSecondIsOk) {
143         stTime.bSecondIsOk = 0;
144         if (++stTime.mSecond >= 60) {
145           stTime.mSecond = 0;
146           if (++stTime.mMinute >= 60) {
147             stTime.mMinute = 0;
148             if (++stTime.mHour >= 24) {
149               stTime.mHour = 0;
150             }
151           }
152         }
153         TimeToBuff(&stTime);
154       }
155       if (sSysTickTimer.bTimeOk) {
156         sSysTickTimer.bTimeOk = 0;
157         // TimeToBuff();
158         HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
159       }
160     }
161     /* USER CODE END 3 */
162   }
```

main.c:

```
112     /* Infinite loop */
113     /* USER CODE BEGIN WHILE */
114     while (1)
115     {
116       /* USER CODE END WHILE */
117
118       /* USER CODE BEGIN 3 */
119       if (stTime.bTenMilIsOk) {
120         stTime.bTenMilIsOk = 0;
121       }
122       if (stTime.bSecondIsOk) {
123         stTime.bSecondIsOk = 0;
124         if (++stTime.mSecond >= 60) {
125           stTime.mSecond = 0;
126           if (++stTime.mMinute >= 60) {
127             stTime.mMinute = 0;
128             if (++stTime.mHour >= 24) {
129               stTime.mHour = 0;
130             }
131           }
132         }
133         TimeToBuff(&stTime);
134       }
135       if (sSysTickTimer.bTenMilSecOk) {
136         sSysTickTimer.bTenMilSecOk = 0;
137         KeyValue = MatrixKeyScan();
138         if (KeyValue != NO_KEY) {
139           for (int i = 0; i < 16; i++) {
140             if (KeyValue == KeyTable[i]) {
141               tempValue = i;
142             }
143           }
144           // DispToBuff(tempValue);
145         }
146       }
147       if (sSysTickTimer.bTimeOk) {
148         sSysTickTimer.bTimeOk = 0;
149         // TimeToBuff();
150         HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
151       }
152     }
153     /* USER CODE END 3 */
154   }
```