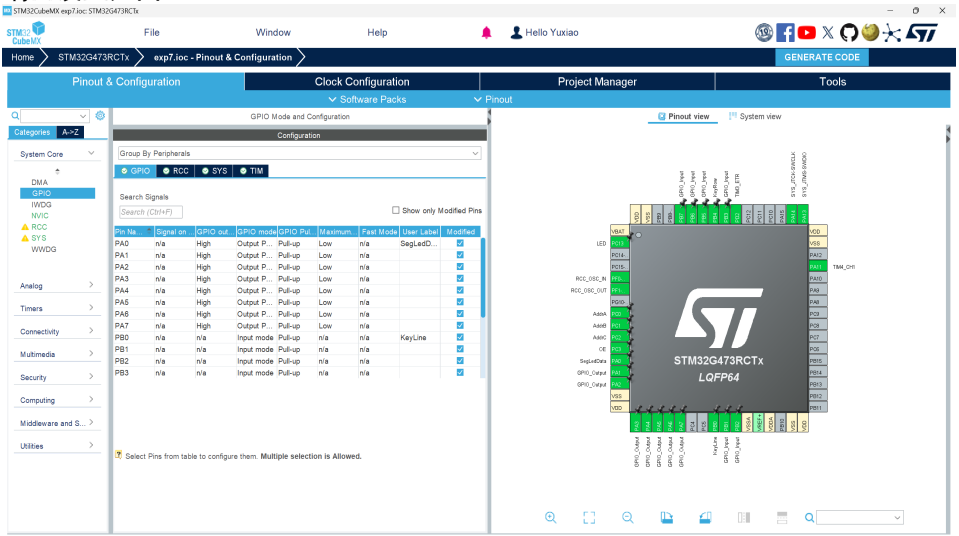
《嵌入式系统原理与实践》作业

10213903403 岳锦鹏
10225001410 朱宇笑

2024 年 12 月 30 日

目录

端口设置如图。


STM32CubeMX exp7.ioc: STM32G473RCTx
File
Window
Help
Hello Yuxiao
Home
STM32G473RCTx
exp7.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
Search (Ctrl+F)
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
PB0 n/a n/a Input mode Pull-up n/a n/a KeyLine
PB1 n/a n/a Input mode Pull-up n/a n/a
PB2 n/a n/a Input mode Pull-up n/a n/a
PB3 n/a n/a Input mode Pull-up n/a n/a
Select Pins from table to configure them. Multiple selection is Allowed.
Categories
A->Z
System Core
DMA
GPIO
IWDG
NVIC
RCC
SYS
WWDG
Analog
Timers
Connectivity
Multimedia
Security
Computing
Middleware and S...
Utilities
Pinout view
System view
STM32G473RCTx
LQFP64

添加 tim.h:

```
20  /* Define to prevent recursive inclusion -------------------------------*/
21  #ifndef __TIM_H__
22  #define __TIM_H__
23
24  #ifdef __cplusplus
25  extern "C" {
26  #endif
27
28  /* Includes ------------------------------------------------------------*/
29  #include "main.h"
30
31  /* USER CODE BEGIN Includes */
32
33  /* USER CODE END Includes */
34
35  extern TIM_HandleTypeDef htim3;
36
37  extern TIM_HandleTypeDef htim4;
38
```

```
39  /* USER CODE BEGIN Private defines */
40
41  /* USER CODE END Private defines */
42
43  void MX_TIM3_Init(void);
44  void MX_TIM4_Init(void);
45
46  void HAL_TIM_MspPostInit(TIM_HandleTypeDef *htim);
47
48  /* USER CODE BEGIN Prototypes */
49
50  /* USER CODE END Prototypes */
51
52  #ifdef __cplusplus
53  }
54  #endif
55
56  #endif /* __TIM_H__ */
```

添加 tim.c:

```
20  /* Includes ------------------------------------------*/
21  #include "tim.h"
22
23  /* USER CODE BEGIN 0 */
24
25  /* USER CODE END 0 */
26
27  TIM_HandleTypeDef htim3;
28  TIM_HandleTypeDef htim4;
29
30  /* TIM3 init function */
31  void MX_TIM3_Init(void)
32  {
33
34    /* USER CODE BEGIN TIM3_Init 0 */
35
36    /* USER CODE END TIM3_Init 0 */
37
38    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
39    TIM_MasterConfigTypeDef sMasterConfig = {0};
40
41    /* USER CODE BEGIN TIM3_Init 1 */
42
43    /* USER CODE END TIM3_Init 1 */
44    htim3.Instance = TIM3;
45    htim3.Init.Prescaler = 0;
46    htim3.Init.CounterMode =
      ↪  TIM_COUNTERMODE_CENTERALIGNED1;
47    htim3.Init.Period = 9;
48    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
49    htim3.Init.AutoReloadPreload =
      ↪  TIM_AUTORELOAD_PRELOAD_ENABLE;
```

添加 tim.c:

```
58    if (HAL_TIM_ConfigClockSource(&htim3,
      ↪  &sClockSourceConfig) != HAL_OK)
59    {
60      Error_Handler();
61    }
62    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
63    sMasterConfig.MasterSlaveMode =
      ↪  TIM_MASTERSLAVEMODE_DISABLE;
64    if (HAL_TIMEx_MasterConfigSynchronization(&htim3,
      ↪  &sMasterConfig) != HAL_OK)
65    {
66      Error_Handler();
67    }
68    /* USER CODE BEGIN TIM3_Init 2 */
69
```

```
70    /* USER CODE END TIM3_Init 2 */
71
72  }
73  /* TIM4 init function */
74  void MX_TIM4_Init(void)
75  {
76
77    /* USER CODE BEGIN TIM4_Init 0 */
78
79    /* USER CODE END TIM4_Init 0 */
80
81    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
82    TIM_MasterConfigTypeDef sMasterConfig = {0};
83    TIM_OC_InitTypeDef sConfigOC = {0};
```

添加 tim.c:

```
85    /* USER CODE BEGIN TIM4_Init 1 */
86
87    /* USER CODE END TIM4_Init 1 */
88    htim4.Instance = TIM4;
89    htim4.Init.Prescaler = 170-1;
90    htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
91    htim4.Init.Period = 99;
92    htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
93    htim4.Init.AutoReloadPreload =
      ↪  TIM_AUTORELOAD_PRELOAD_ENABLE;
94    if (HAL_TIM_Base_Init(&htim4) != HAL_OK)
95    {
96      Error_Handler();
97    }
98    sClockSourceConfig.ClockSource =
      ↪  TIM_CLOCKSOURCE_INTERNAL;
99    if (HAL_TIM_ConfigClockSource(&htim4,
      ↪  &sClockSourceConfig) != HAL_OK)
100   {
101     Error_Handler();
102   }
103   if (HAL_TIM_PWM_Init(&htim4) != HAL_OK)
104   {
105     Error_Handler();
106   }
107   sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
108   sMasterConfig.MasterSlaveMode =
      ↪  TIM_MASTERSLAVEMODE_DISABLE;
109   if (HAL_TIMEx_MasterConfigSynchronization(&htim4,
      ↪  &sMasterConfig) != HAL_OK)
110   {
111     Error_Handler();
112   }
113   sConfigOC.OCMode = TIM_OCMODE_PWM1;
114   sConfigOC.Pulse = 50;
115   sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
116   sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
117   if (HAL_TIM_PWM_ConfigChannel(&htim4, &sConfigOC,
      ↪  TIM_CHANNEL_1) != HAL_OK)
118   {
119     Error_Handler();
120   }
121   /* USER CODE BEGIN TIM4_Init 2 */
122
123   /* USER CODE END TIM4_Init 2 */
124   HAL_TIM_MspPostInit(&htim4);
125
126 }
```

添加 tim.c:

```
128  void HAL_TIM_Base_MspInit(TIM_HandleTypeDef*
     ↪  tim_baseHandle)
129  {
130
131    GPIO_InitTypeDef GPIO_InitStruct = {0};
132    if(tim_baseHandle->Instance==TIM3)
133    {
134    /* USER CODE BEGIN TIM3_MspInit 0 */
135
136    /* USER CODE END TIM3_MspInit 0 */
137      /* TIM3 clock enable */
138      __HAL_RCC_TIM3_CLK_ENABLE();
139
140      __HAL_RCC_GPIOD_CLK_ENABLE();
141      /**TIM3 GPIO Configuration
142      PD2      ------> TIM3_ETR
143      */
144      GPIO_InitStruct.Pin = GPIO_PIN_2;
145      GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
146      GPIO_InitStruct.Pull = GPIO_NOPULL;
147      GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
148      GPIO_InitStruct.Alternate = GPIO_AF2_TIM3;
149      HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
150
151      /* TIM3 interrupt Init */
152      HAL_NVIC_SetPriority(TIM3_IRQn, 0, 0);
153      HAL_NVIC_EnableIRQ(TIM3_IRQn);
154    /* USER CODE BEGIN TIM3_MspInit 1 */
155
156    /* USER CODE END TIM3_MspInit 1 */
157    }
158    else if(tim_baseHandle->Instance==TIM4)
159    {
160    /* USER CODE BEGIN TIM4_MspInit 0 */
161
162    /* USER CODE END TIM4_MspInit 0 */
163      /* TIM4 clock enable */
164      __HAL_RCC_TIM4_CLK_ENABLE();
165    /* USER CODE BEGIN TIM4_MspInit 1 */
166
167    /* USER CODE END TIM4_MspInit 1 */
168    }
169  }
```

添加 tim.c:

```
170 void HAL_TIM_MspPostInit(TIM_HandleTypeDef* timHandle)
171 {
172
173   GPIO_InitTypeDef GPIO_InitStruct = {0};
174   if(timHandle->Instance==TIM4)
175   {
176   /* USER CODE BEGIN TIM4_MspPostInit 0 */
177
178   /* USER CODE END TIM4_MspPostInit 0 */
179
180     __HAL_RCC_GPIOA_CLK_ENABLE();
181     /**TIM4 GPIO Configuration
182     PA11     ------> TIM4_CH1
183     */
184     GPIO_InitStruct.Pin = GPIO_PIN_11;
185     GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
186     GPIO_InitStruct.Pull = GPIO_NOPULL;
187     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
188     GPIO_InitStruct.Alternate = GPIO_AF10_TIM4;
189     HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
190
191   /* USER CODE BEGIN TIM4_MspPostInit 1 */
192
193   /* USER CODE END TIM4_MspPostInit 1 */
194   }
195
196 }
```

添加 tim.c:

```
198 void HAL_TIM_Base_MspDeInit(TIM_HandleTypeDef*
    ↪ tim_baseHandle)
199 {
200
201   if(tim_baseHandle->Instance==TIM3)
202   {
203   /* USER CODE BEGIN TIM3_MspDeInit 0 */
204
205   /* USER CODE END TIM3_MspDeInit 0 */
206     /* Peripheral clock disable */
207     __HAL_RCC_TIM3_CLK_DISABLE();
208
209     /**TIM3 GPIO Configuration
210     PD2     ------> TIM3_ETR
211     */
212     HAL_GPIO_DeInit(GPIOD, GPIO_PIN_2);
213
214     /* TIM3 interrupt Deinit */
215     HAL_NVIC_DisableIRQ(TIM3_IRQn);
216   /* USER CODE BEGIN TIM3_MspDeInit 1 */
217
218   /* USER CODE END TIM3_MspDeInit 1 */
219   }
220   else if(tim_baseHandle->Instance==TIM4)
221   {
222   /* USER CODE BEGIN TIM4_MspDeInit 0 */
223
224   /* USER CODE END TIM4_MspDeInit 0 */
225     /* Peripheral clock disable */
226     __HAL_RCC_TIM4_CLK_DISABLE();
227   /* USER CODE BEGIN TIM4_MspDeInit 1 */
228
229   /* USER CODE END TIM4_MspDeInit 1 */
230   }
231 }
```

main.c:

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


49  /* USER CODE BEGIN PV */
50  stSysTickTimer sSysTickTimer = {
51    0, 0, 0, 0
52  };
53  uint8_t tempValue;
```

```
54  structTime stTime = {
55    .mSecond = 50,
56    .mMinute = 45,
57    .mHour = 8,
58    .mTimeCount = 0,
59    .bSecondIsOk = 0,
60    .mTenMilCount = 0,
61    .bTenMilIsOk = 0
62  };
63  /* USER CODE END PV */
64
65  /* Private function prototypes -----------------*/
66  void SystemClock_Config(void);
```

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();
  /* USER CODE BEGIN 2 */
  // FlashLeds_GPIO_Port->ODR &= 0xff01;
  HAL_TIM_Base_Start_IT(&htim3);
  TimeToBuff(&stTime);
  /* USER CODE END 2 */
```

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   }
```

数码管显示 08-45-50，循环闪烁，逻辑与正常定时器相同，但时间变化很慢。

完整视频可以查看：
https://gitea.librastalker.top/423A35C7/STM32CubeMX-Keil_uVision5