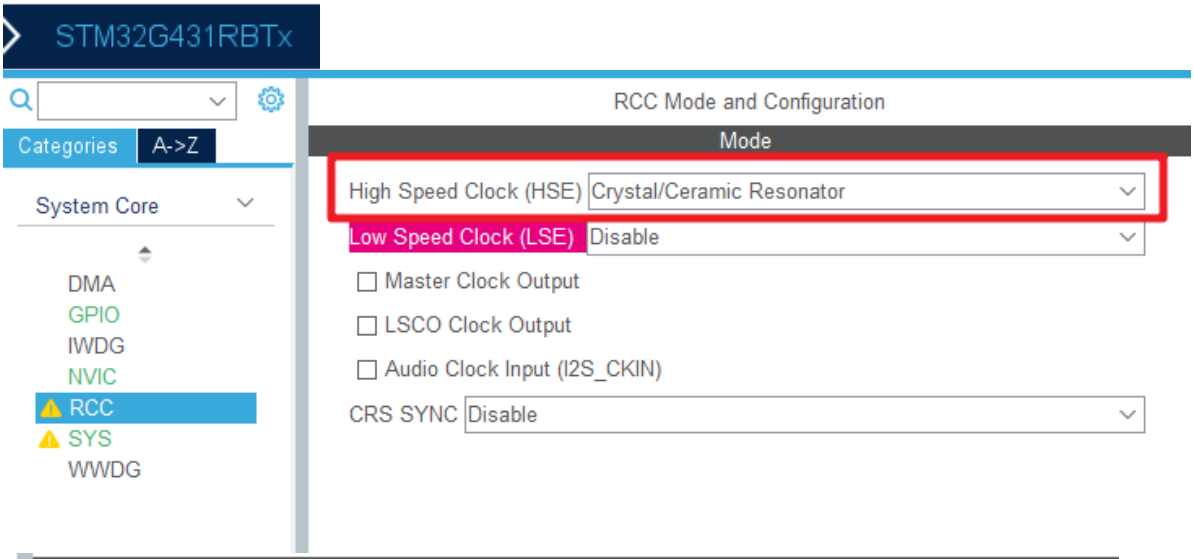
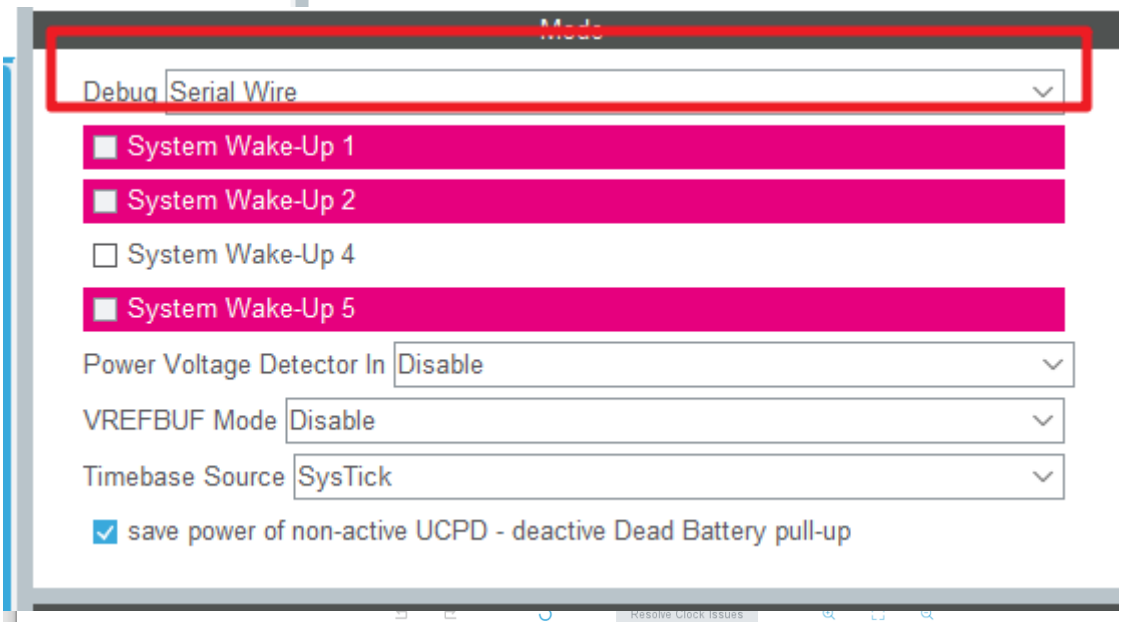
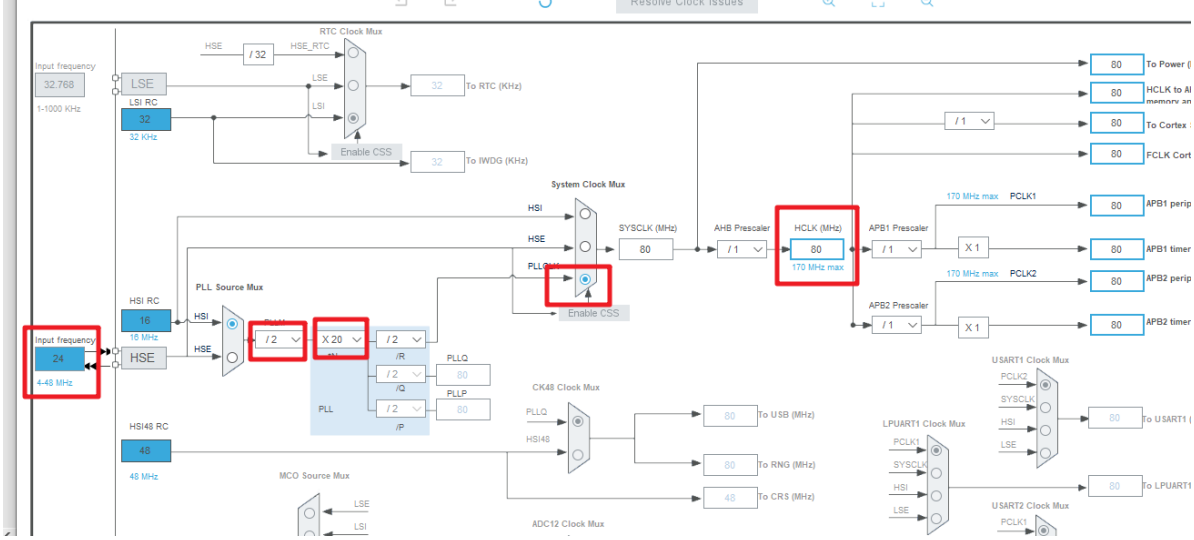
快速上手蓝桥杯比赛流程

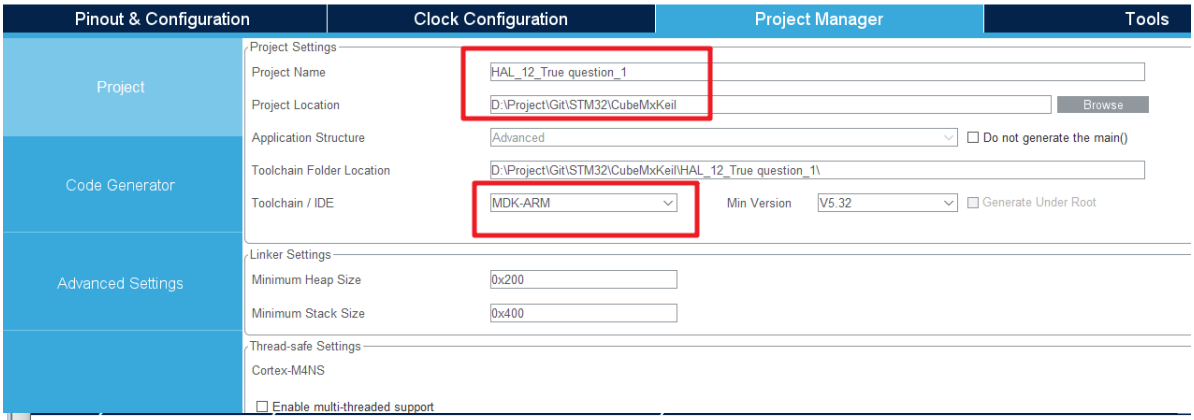
1.新建工程

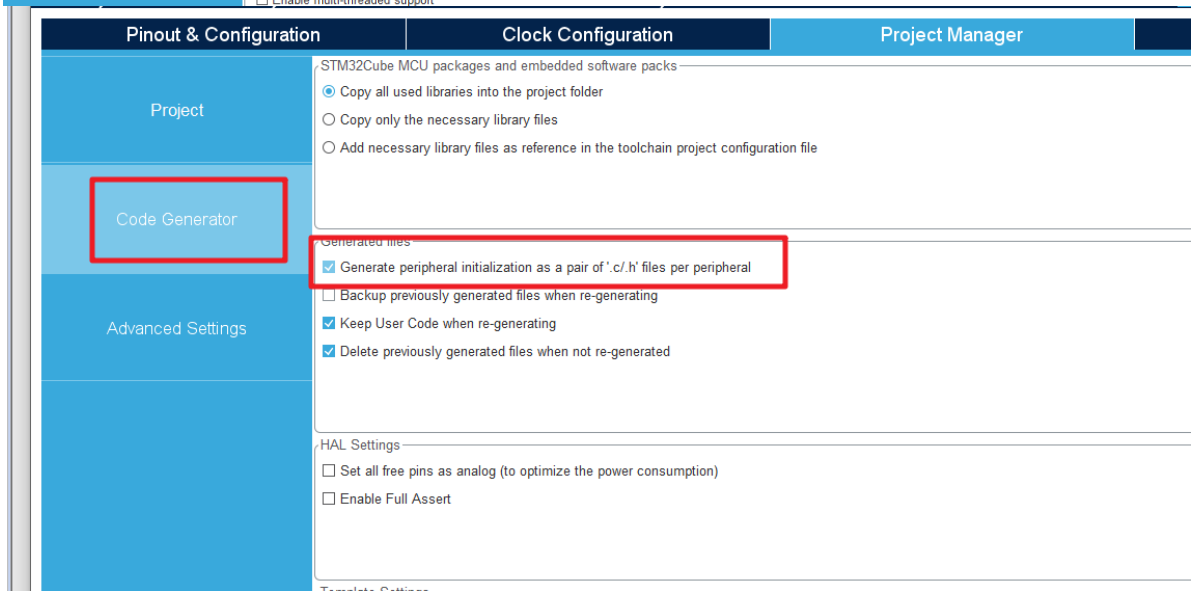
stm32G431RBT6



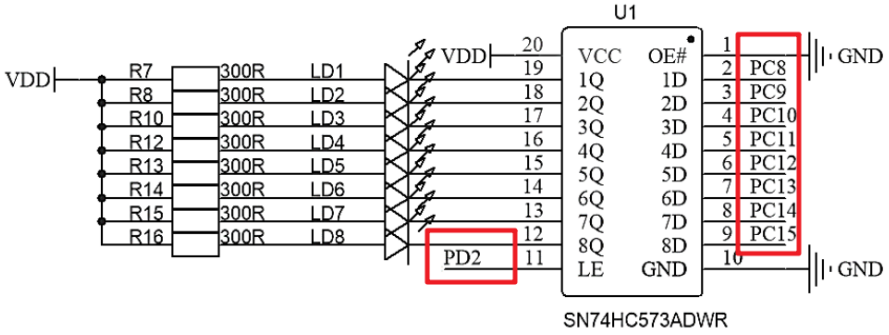




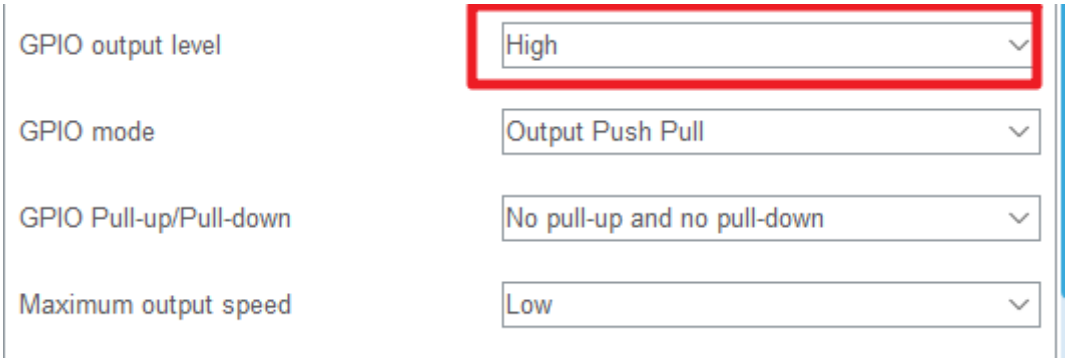




7.4 LED 指示灯

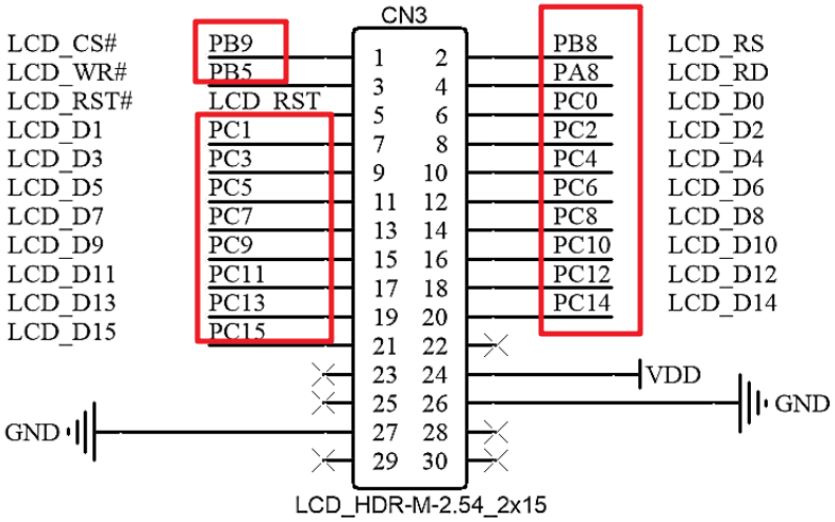


PC8-PC15



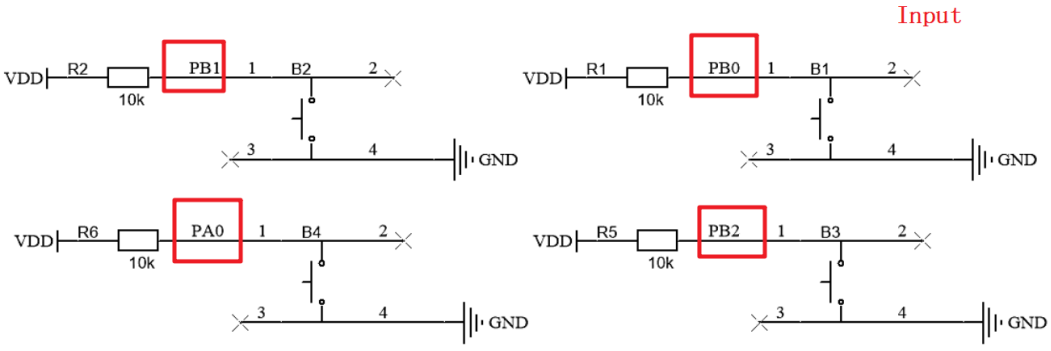
7.6 LCD

Output

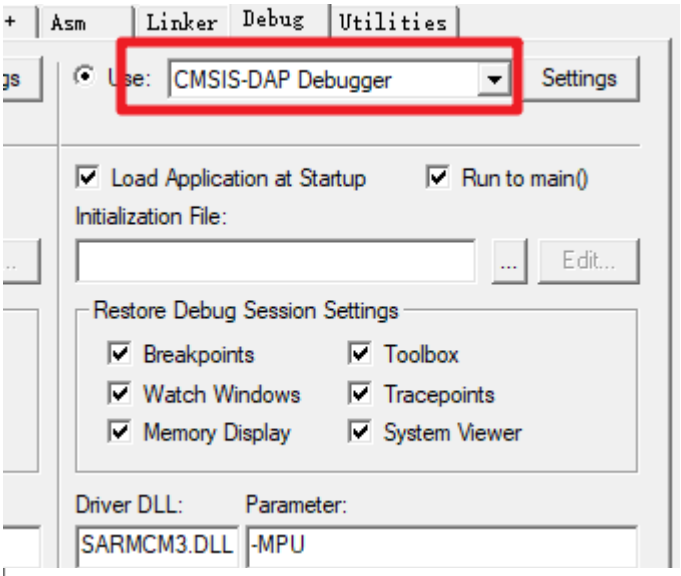


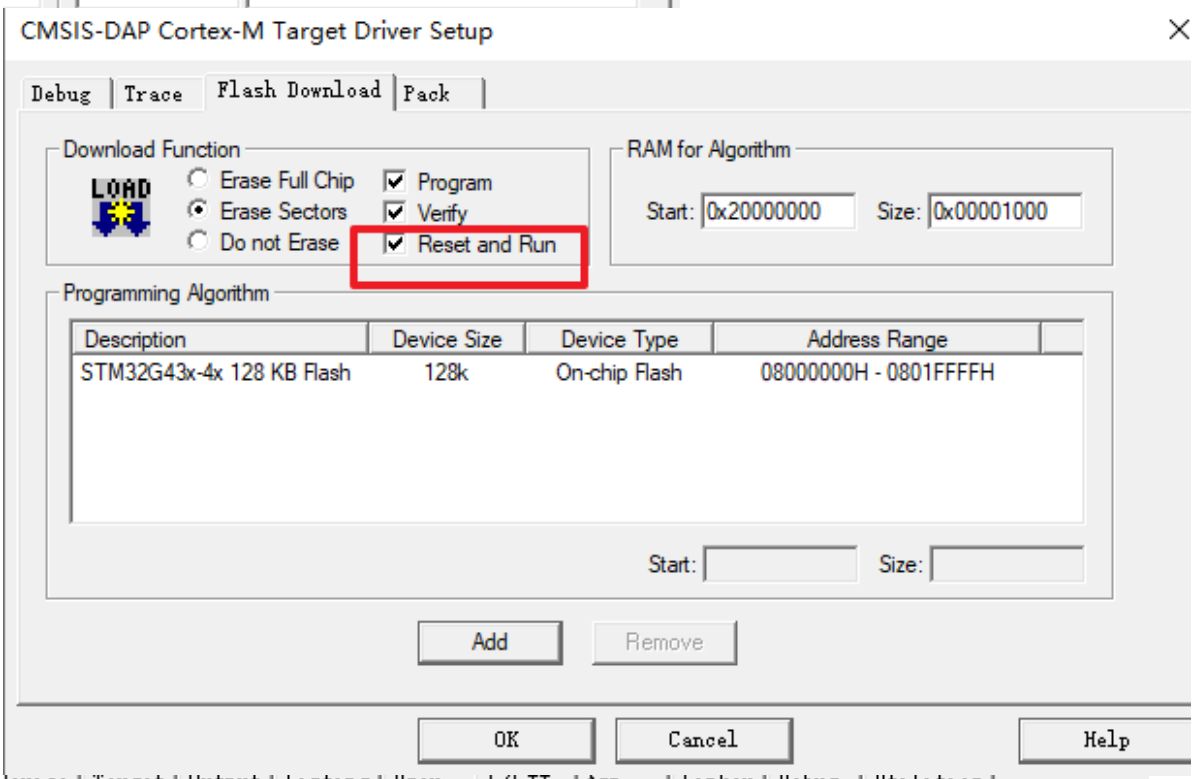
7.7 按键

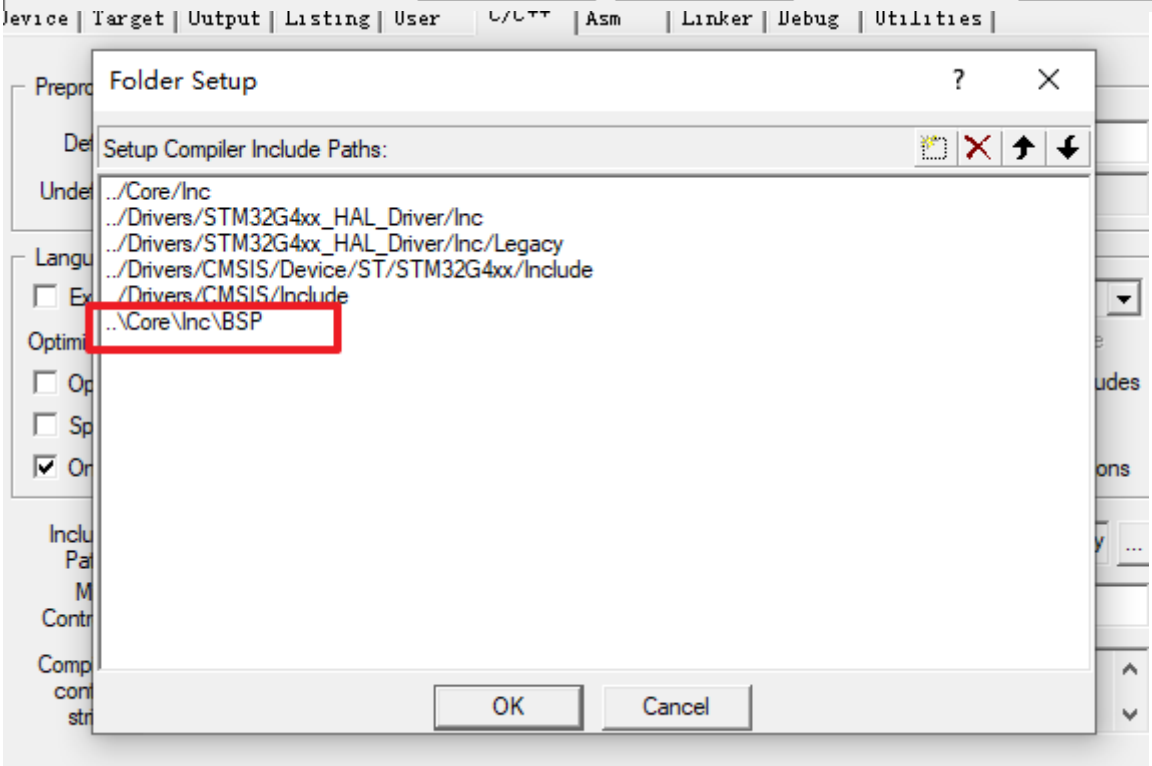
Input



LCD测试


Asm
Linker
Debug
Utilities
Use:
CMSIS-DAP Debugger
Settings
Load Application at Startup
Run to main()
Initialization File:
Edit...
Restore Debug Session Settings
Breakpoints
Toolbox
Watch Windows
Tracepoints
Memory Display
System Viewer
Driver DLL:
Parameter:
SARMCM3.DLL
-MPU


CMSIS-DAP Cortex-M Target Driver Setup
Debug
Trace
Flash Download
Pack
Download Function
LOAD
Erase Full Chip
Erase Sectors
Do not Erase
Program
Verify
Reset and Run
RAM for Algorithm
Start: 0x20000000
Size: 0x00001000
Programming Algorithm
Description
Device Size
Device Type
Address Range
STM32G43x-4x 128 KB Flash
128k
On-chip Flash
08000000H - 0801FFFFH
Start:
Size:
Add
Remove
OK
Cancel
Help


Device
Target
Output
Listing
User
Asm
Linker
Debug
Utilities
Folder Setup
Setup Compiler Include Paths:
../Core/Inc
../Drivers/STM32G4xx_HAL_Driver/Inc
../Drivers/STM32G4xx_HAL_Driver/Inc/Legacy
../Drivers/CMSIS/Device/ST/STM32G4xx/Include
../Drivers/CMSIS/Include
..\Core\Inc\BSP
OK
Cancel

```
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
  ******************************************************************************
  */
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "gpio.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "lcd.h"


/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
```

```
  /* 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();
  /* USER CODE BEGIN 2 */

    LCD_Init();
    LCD_Clear(Black);
    LCD_SetTextColor(White);
    LCD_SetBackColor(Black);

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
```

```
    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
      /* USER CODE END WHILE */

      /* USER CODE BEGIN 3 */

      LCD_DisplayStringLine(Line0,(uint8_t*)"     hello      ");
    }
    /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
```

```
/* Private includes ----------
/* USER CODE BEGIN Includes */
#include "lcd.h"
#include "led.h"
#include "stdio.h"

/* USER CODE END Includes */

/* USER CODE BEGIN PV */

    //LCD相关变量
    uint8_t lcd_string[22];
    __IO uint32_t lcd_uwtick=0;


/* USER CODE END PV */

/* Private function prototypes ----------
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
void lcd_proc(void);
/* USER CODE END PFP */

/* Private user code ----------
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/* USER CODE BEGIN 4 */

void lcd_proc(void)
{
    if(uwTick-lcd_uwtick<500) return;
    lcd_uwtick=uwTick;
    sprintf((char*)lcd_string,"hello");
    LCD_DisplayStringLine(Line0,lcd_string)
}

/* USER CODE END 4 */
```

```
05    /* USER CODE BEGIN WHILE */
06    while (1)
07    {
08      /* USER CODE END WHILE */
09
10      /* USER CODE BEGIN 3 */
11  //    led_disp(0x22);
12  //    HAL_Delay(500);
13  //    led_disp(0x00);
14  //    HAL_Delay(500);
15        lcd_proc();
16
17    }
18    /* USER CODE END 3 */
19  }
20
21  /**
```

LED测试

```c
#include "led.h"

void led_disp(uint8_t led)
{
    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_All,GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_RESET);

    HAL_GPIO_WritePin(GPIOC,led<<8,GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOD,GPIO_PIN_2,GPIO_PIN_RESET);
}

```

```c
#include "gpio.h"

void led_disp(uint8_t led);

  /* Private includes ----------------------
  /* USER CODE BEGIN Includes */
  #include "lcd.h"
  #include "led.h"

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
      /* USER CODE END WHILE */

      /* USER CODE BEGIN 3 */
       led_disp(0x22);
       HAL_Delay(500);
       led_disp(0x00);
       HAL_Delay(500);
//     LCD_DisplayStringLine(Line0,(uint8_t*)"      hello
    }
    /* USER CODE END 3 */
}

/**
```

测试按键

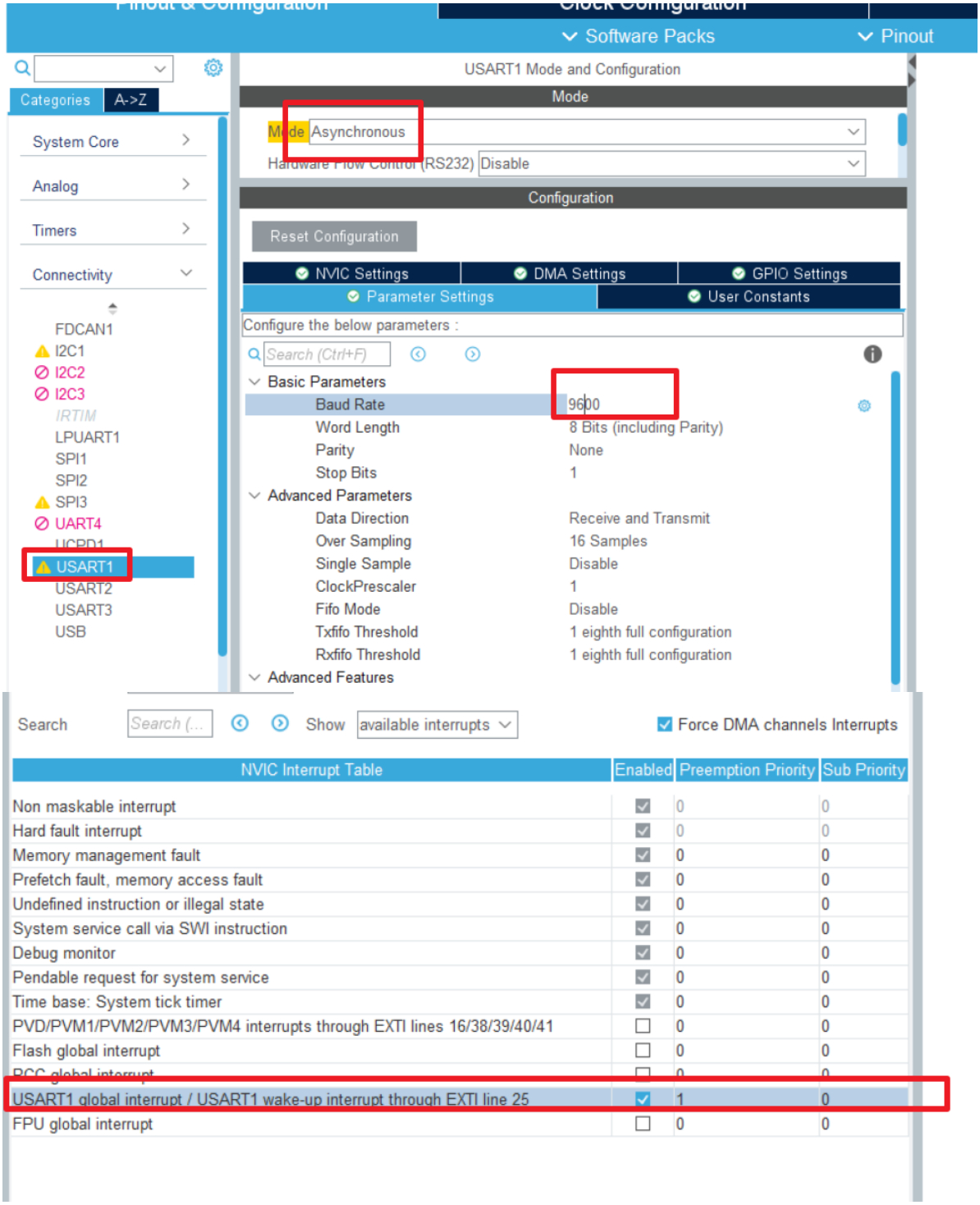
```c
#include "gpio.h"

uint32_t key_scan(void)
{

}
```

```c
#include "gpio.h"

uint32_t key_scan(void)
{
    uint32_t key=0;
    if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_0)==GPIO_PIN_RESET)
        key=1;
    if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_1)==GPIO_PIN_RESET)
        key=2;
    if(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_2)==GPIO_PIN_RESET)
        key=3;
    if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0)==GPIO_PIN_RESET)
        key=4;
    return key;
}
```

```c
#include "main.h"
#include "gpio.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "lcd.h"
#include "led.h"
#include "key.h"
#include "stdio.h"

/* USER CODE END Includes */
    IO uint32_t lcd_uwtick=0;

    //key相关变量 key_proc
    uint32_t key_value;
    uint32_t key_down;
    uint32_t key_up;
    uint32_t key_old;
    __IO uint32_t key_uwtick=0;

/* USER CODE END PV */
```

```c
/* USER CODE BEGIN PFP */

void lcd_proc(void);
void key_proc(void);

/* USER CODE END PFP */

/* Private user code ----------
/* USER CODE BEGIN 0 */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
//    led_disp(0x22);
//    HAL_Delay(500);
//    led_disp(0x00);
//    HAL_Delay(500);
    lcd_proc();
    key_proc();

  }
    lcd_uwTick=uwTick;
    sprintf((char*)lcd_string,"hello");
    LCD_DisplayStringLine(Line0,lcd_string);

}

void key_proc(void)
{
    if(uwTick-key_uwtick<50) return;
    key_uwtick=uwTick;
    key_value=key_scan();
    key_down=key_value&(key_old^key_value);
    key_up=~key_value&(key_old^key_value);
    key_old=key_value;

    if(key_down!=0)
    {
        sprintf((char*)lcd_string,"key:%d",key_down);
        LCD_DisplayStringLine(Line1,lcd_string);
    }

}

/* USER CODE END 4 */
```

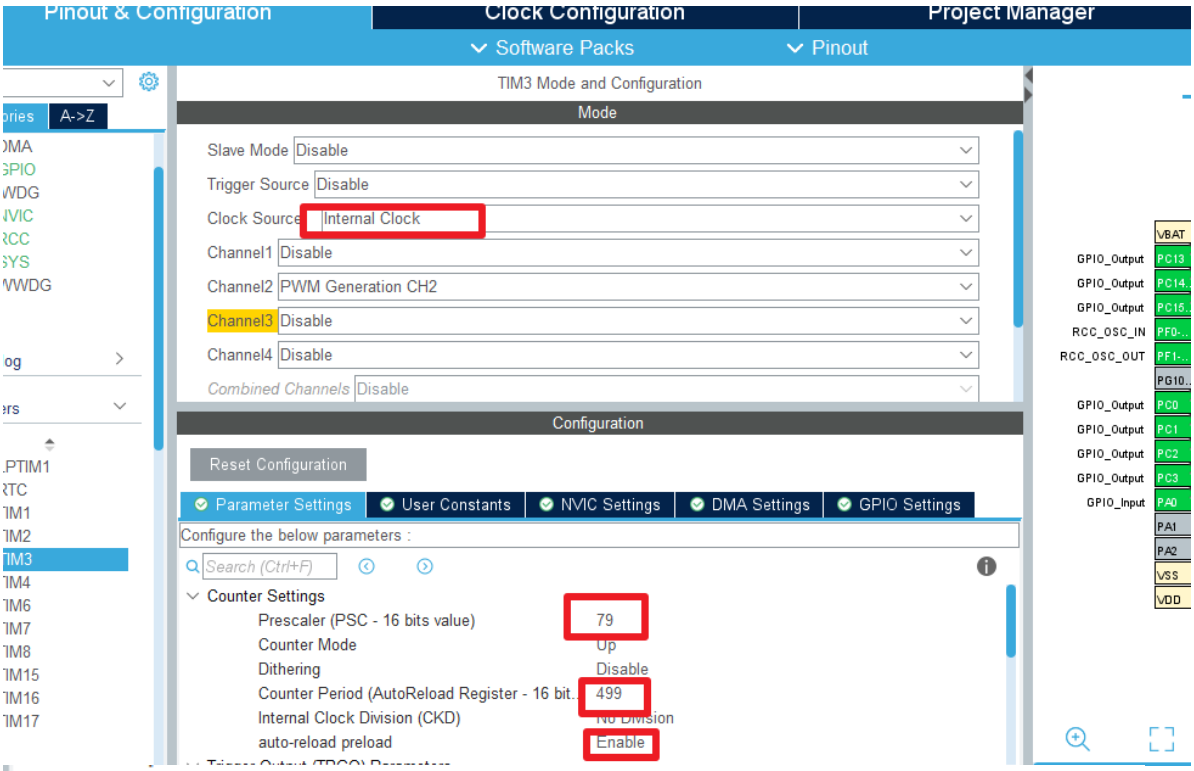
测试UART

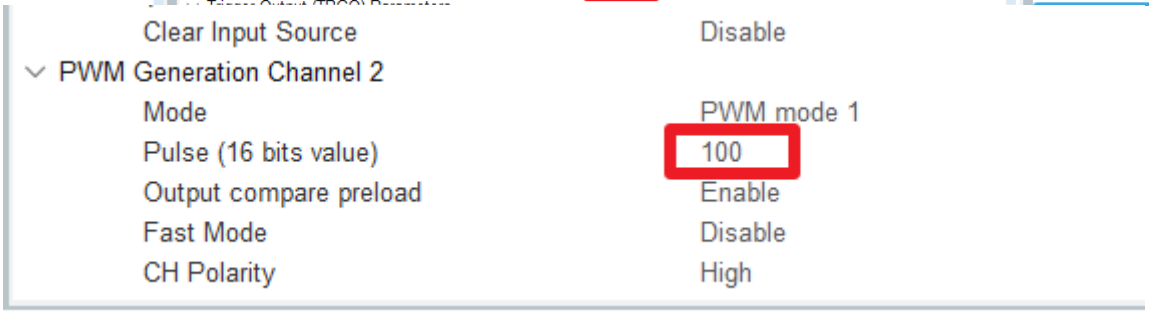

Pinout & Configuration
Clock Configuration
Software Packs
Pinout
Categories
A->Z
System Core
Analog
Timers
Connectivity
FDCAN1
I2C1
I2C2
I2C3
IRTIM
LPUART1
SPI1
SPI2
SPI3
UART4
USART1
USART2
USART3
USB
USART1 Mode and Configuration
Mode
Mode Asynchronous
Hardware Flow Control (RS232) Disable
Configuration
Reset Configuration
NVIC Settings
DMA Settings
GPIO Settings
Parameter Settings
User Constants
Configure the below parameters :
Search (Ctrl+F)
Basic Parameters
Baud Rate 9600
Word Length 8 Bits (including Parity)
Parity None
Stop Bits 1
Advanced Parameters
Data Direction Receive and Transmit
Over Sampling 16 Samples
Single Sample Disable
ClockPrescaler 1
Fifo Mode Disable
Txfifo Threshold 1 eighth full configuration
Rxfifo Threshold 1 eighth full configuration
Advanced Features
Search
Search (...
Show available interrupts
Force DMA channels Interrupts
NVIC Interrupt Table
Enabled
Preemption Priority
Sub Priority
Non maskable interrupt 0 0
Hard fault interrupt 0 0
Memory management fault 0 0
Prefetch fault, memory access fault 0 0
Undefined instruction or illegal state 0 0
System service call via SWI instruction 0 0
Debug monitor 0 0
Pendable request for system service 0 0
Time base: System tick timer 0 0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/38/39/40/41 0 0
Flash global interrupt 0 0
RCC global interrupt 0 0
USART1 global interrupt / USART1 wake-up interrupt through EXTI line 25 1 0
FPU global interrupt 0 0

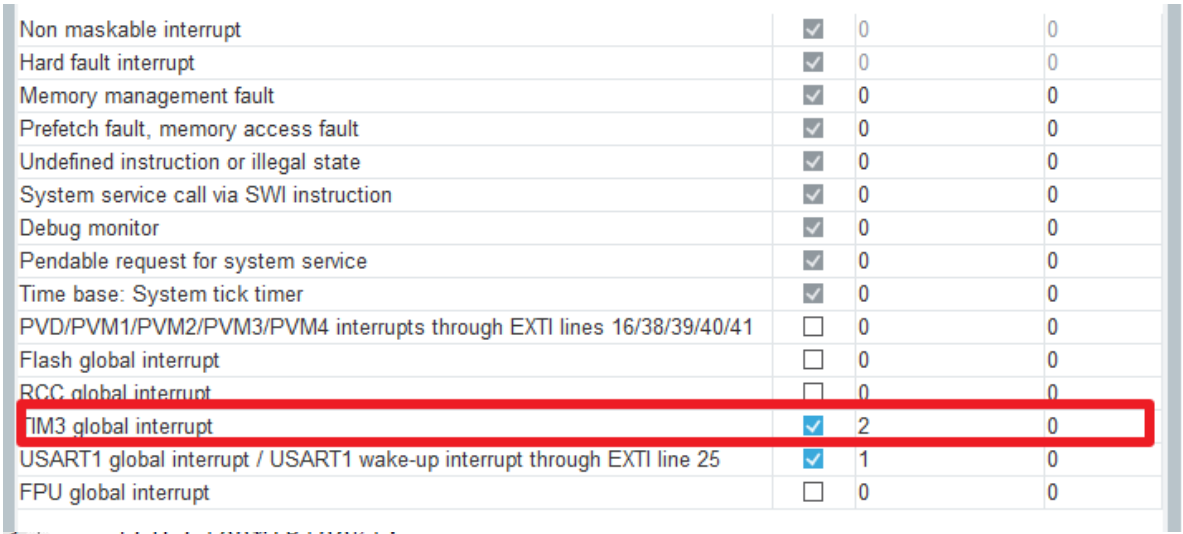
```c
    //key相关变量 key_proc
    uint32_t key_value;
    uint32_t key_down;
    uint32_t key_up;
    uint32_t key_old;
    __IO uint32_t key_uwtick=0;

    //uart 相关变量
    uint8_t rx_buffer;
    char str[50];
MX_USART1_UART_Init();
/* USER CODE BEGIN 2 */

  LCD_Init();
  LCD_Clear(Black);
  LCD_SetTextColor(White);
  LCD_SetBackColor(Black);

  led_disp(0x00);

  HAL_UART_Receive_IT(&huart1,&rx_buffer,1);

/* USER CODE END 2 */

}
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if(rx_buffer=='a')
    {
        sprintf(str,"success\n");

    }
    else
    {
        sprintf(str,"error\n");

    }
    HAL_UART_Transmit(&huart1,(uint8_t*)str,strlen(str),50);
    HAL_UART_Receive_IT(&huart1,&rx_buffer,1);
}
/* USER CODE END 4 */
```

PA10 PA9

测试PWM







Non maskable interrupt	☑	0	0
Hard fault interrupt	☑	0	0
Memory management fault	☑	0	0
Prefetch fault, memory access fault	☑	0	0
Undefined instruction or illegal state	☑	0	0
System service call via SWI instruction	☑	0	0
Debug monitor	☑	0	0
Pendable request for system service	☑	0	0
Time base: System tick timer	☑	0	0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/38/39/40/41	☐	0	0
Flash global interrupt	☐	0	0
RCC global interrupt	☐	0	0
TIM3 global interrupt	☑	2	0
USART1 global interrupt / USART1 wake-up interrupt through EXTI line 25	☑	1	0
FPU global interrupt	☐	0	0

```
    LCD_Clear(Black);
    LCD_SetTextColor(White);
    LCD_SetBackColor(Black);

    led_disp(0x00);

    HAL_UART_Receive_IT(&huart1, &rx_buffer, 1);

    HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_2);

/* USER CODE END 2 */

/* Infinite loop */
```

```
            pwm_flag=1;
            __HAL_TIM_SET_COMPARE(&htim3,TIM_CHANNEL_2,0);

    }
```

```
__HAL_TIM_SET_AUTORELOAD(&htim2,499);//频率
__HAL_TIM_SET_COMPARE(&htim2,TIM_CHANNEL_2,50);//占空比
```