

Pinout & Configuration

Clock Configuration

Project Manager

Project

Code Generator

Advanced Settings

STM32Cube MCU packages and embedded software packs

☒ Copy all used libraries into the project folder

☐ Copy only the necessary library files

☐ Add necessary library files as reference in the toolchain project configuration file

Generated files

☒ Generate peripheral initialization as a pair of '.c/.h' files per peripheral

☐ Backup previously generated files when re-generating

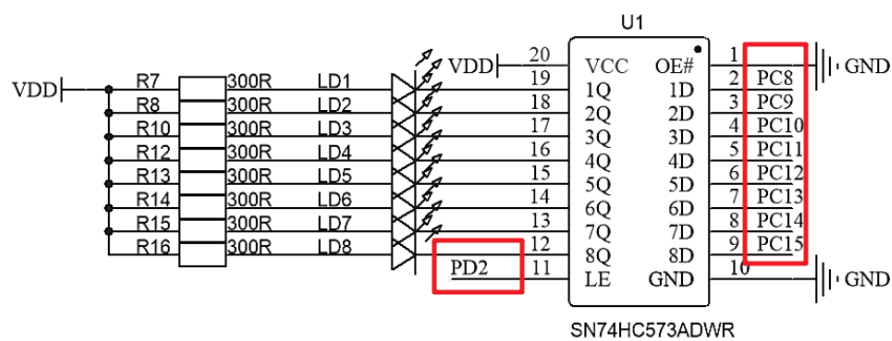
☒ Keep User Code when re-generating

☒ Delete previously generated files when not re-generated

HAL Settings

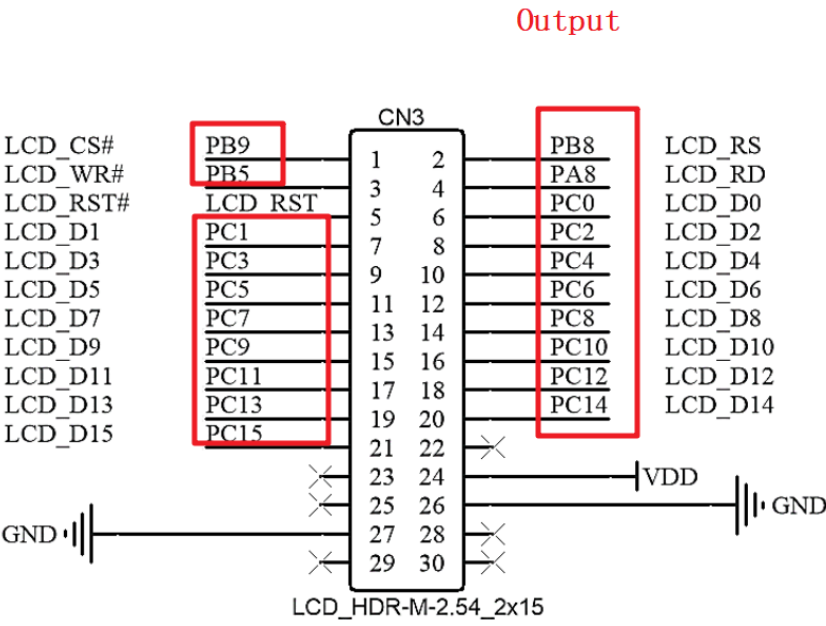
☐ Set all free pins as analog (to optimize the power consumption)

☐ Enable Full Assert

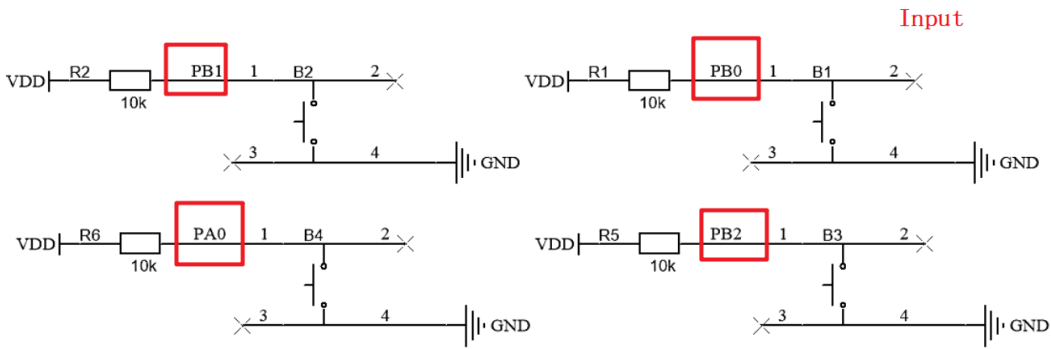


GPIO output level	High
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low

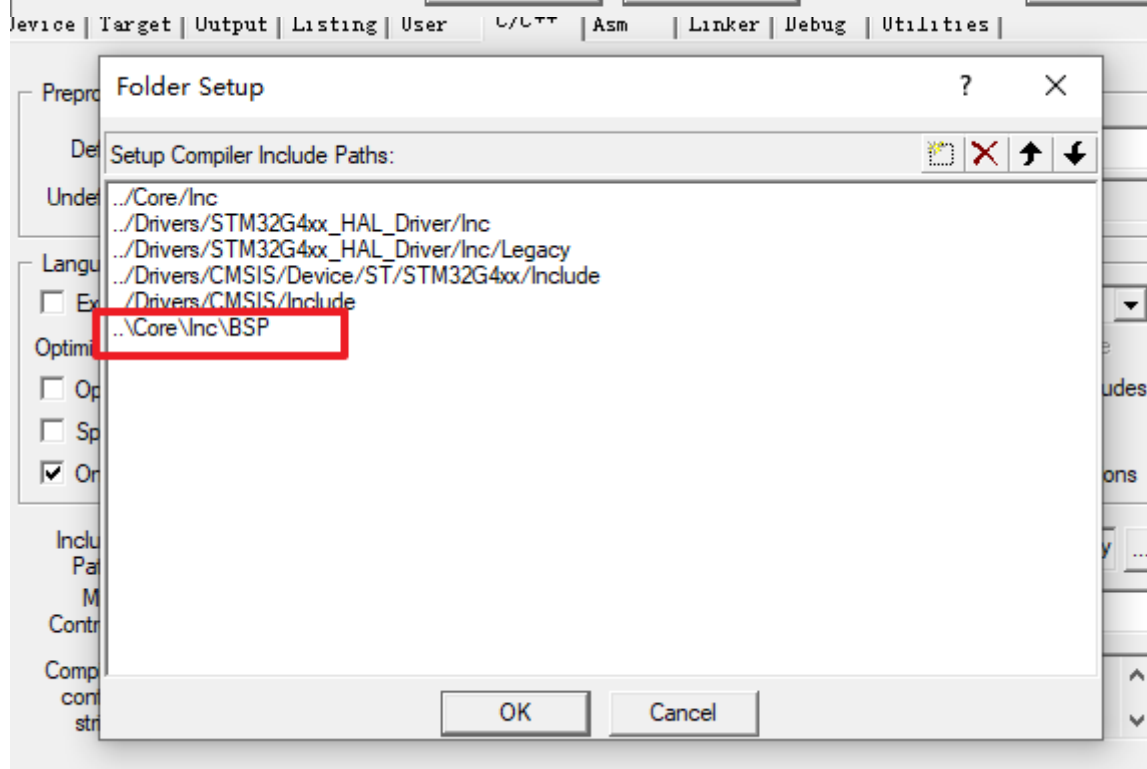
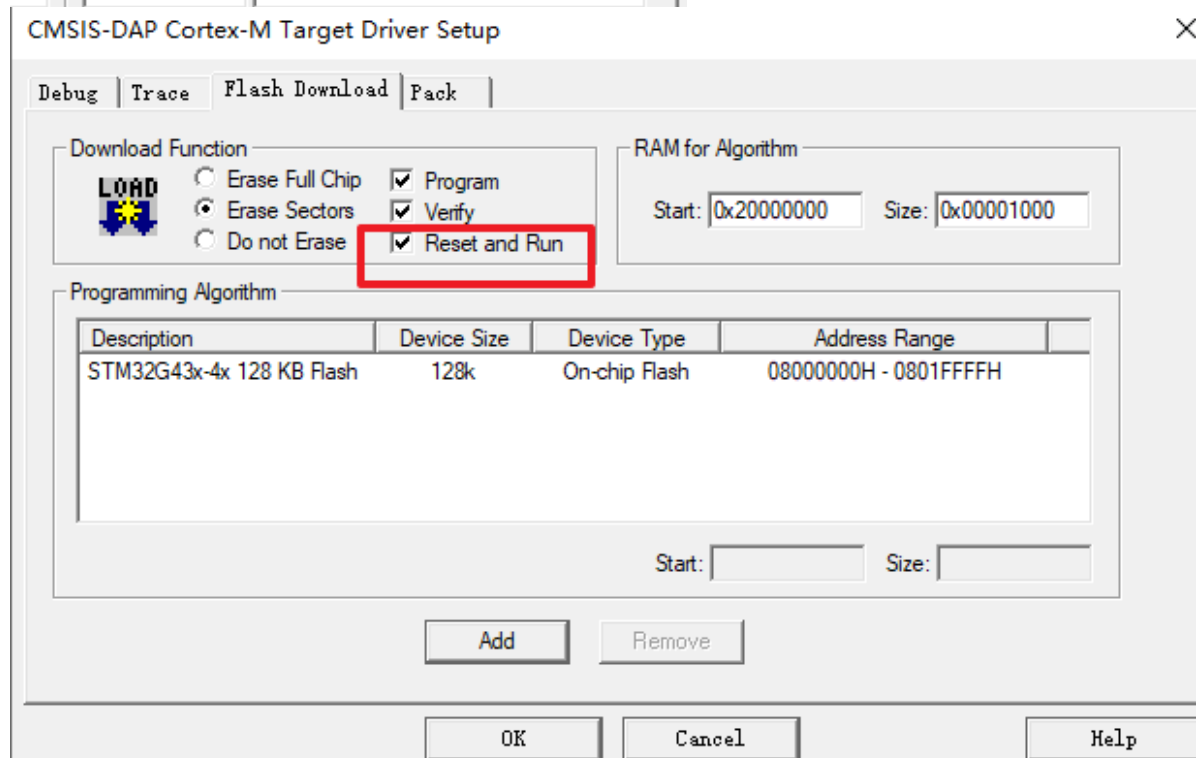
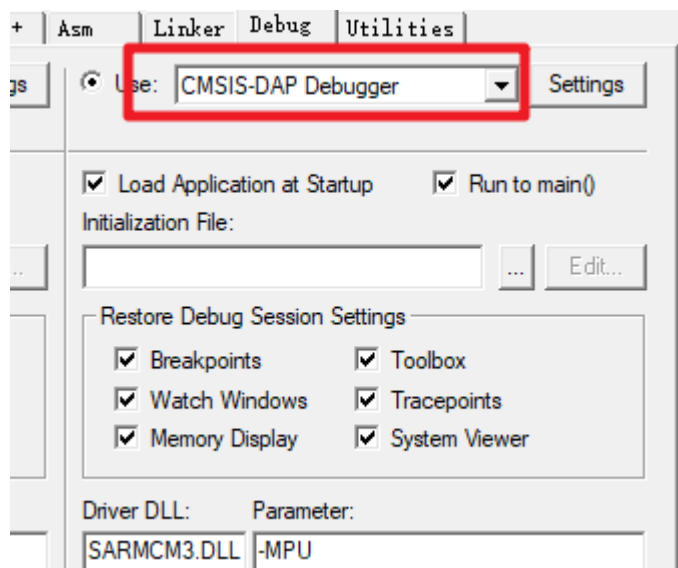
# 7.6 LCD



# 7.7 按键



# LCD测试



```

14  * If no LICENSE file comes with this software, it is provided AS-IS.
15  *
16  ****
17  */
18  /* USER CODE END Header */
19  /* Includes -----
20  #include "main.h"
21  #include "gpio.h"
22
23  /* Private includes -----
24  /* USER CODE BEGIN Includes */
25  #include "lcd.h"
26
27
28  /* USER CODE END Includes */
29
30  /* Private typedef
31
32  /* USER CODE END Init */
33
34  /* Configure the system clock */
35  SystemClock_Config();
36
37  /* USER CODE BEGIN SysInit */
38
39  /* USER CODE END SysInit */
40
41  /* Initialize all configured peripherals */
42  MX_GPIO_Init();
43  /* USER CODE BEGIN 2 */
44
45  LCD_Init();
46  LCD_Clear(Black);
47  LCD_SetTextColor(White);
48  LCD_SetBackColor(Black);
49
50  /* USER CODE END 2 */
51
52  /* Infinite loop */
53  /* USER CODE BEGIN WHILE */
54  while (1)
55  {
56      /* USER CODE END WHILE */
57
58      /* USER CODE BEGIN 3 */
59
98     /* Infinite loop */
99     /* USER CODE BEGIN WHILE */
100    while (1)
101    {
102        /* USER CODE END WHILE */
103
104        /* USER CODE BEGIN 3 */
105
106        LCD_DisplayStringLine(Line0, (uint8_t*)"      hello      ");
107    }
108    /* USER CODE END 3 */
109  }
110
111  /**
112   * @brief System Clock Configuration
113   * @retval None

```

```

23 /* Private includes -----
24 /* USER CODE BEGIN Includes */
25 #include "lcd.h"
26 #include "led.h"
27 #include "stdio.h"
28
29 /* USER CODE END Includes */
30
31
32
33
34 /* USER CODE BEGIN PV */
35
36 //LCD相关变量
37 uint8_t lcd_string[22];
38 __IO uint32_t lcd_uwtick=0;
39
40 /* USER CODE END PV */
41
42
43 /* Private function prototypes -----
44 void SystemClock_Config(void) .
45 /* USER CODE END PV */
46
47
48
49
50
51
52
53
54
55
56 /* Private function prototypes -----
57 void SystemClock_Config(void);
58 /* USER CODE BEGIN PFP */
59 void lcd_proc(void);
60 /* USER CODE END PFP */
61
62 /* Private user code -----
63 /* USER CODE BEGIN 0 */
64
65 /* USER CODE END 0 */
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167 /* USER CODE BEGIN 4 */
168
169 void lcd_proc(void)
170 {
171     if(uwTick-lcd_uwtick<500) return;
172     lcd_uwtick=uwTick;
173     sprintf((char*)lcd_string,"hello");
174     LCD_DisplayStringLine(Line0,lcd_string)
175 }
176
177 /* USER CODE END 4 */
178

```

```

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  /* USER CODE END 3 */
18  }
19
20
21 /**

```

## LED测试

---

```

1#include "led.h"
2
3void led_disp(uint8_t led)
4{
5    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_A11, GPIO_PIN_SET);
6    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_2, GPIO_PIN_SET);
7    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_2, GPIO_PIN_RESET);
8
9    HAL_GPIO_WritePin(GPIOC, led<<8, GPIO_PIN_RESET);
10   HAL_GPIO_WritePin(GPIOD, GPIO_PIN_2, GPIO_PIN_SET);
11   HAL_GPIO_WritePin(GPIOD, GPIO_PIN_2, GPIO_PIN_RESET);
12}
13
14#include "gpio.h"
15
16void led_disp(uint8_t led);
17
18/* Private includes -----
19/* USER CODE BEGIN Includes */
20#include "lcd.h"
21#include "led.h"
22
23/* Infinite loop */
24/* USER CODE BEGIN WHILE */
25while (1)
26{
27    /* USER CODE END WHILE */
28
29    /* USER CODE BEGIN 2 */
30    led_disp(0x22);
31    HAL_Delay(500);
32    led_disp(0x00);
33    HAL_Delay(500);
34    // LCD_DisplayStringLine(Line0, (uint8_t*)"      hello
35    }
36    /* USER CODE END 3 */
37}
38
39/**
40 * @brief  The main function of the program
41 */

```

## 测试按键



```
#include "gpio.h"

uint32_t key_scan(void)
```

```
1#include "gpio.h"
2
3uint32_t key_scan(void)
4{
5    uint32_t key=0;
6    if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0)==GPIO_PIN_RESET)
7        key=1;
8    if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_1)==GPIO_PIN_RESET)
9        key=2;
10   if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2)==GPIO_PIN_RESET)
11       key=3;
12   if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)==GPIO_PIN_RESET)
13       key=4;
14   return key;
15}
16
```

```
20#include main.h
21#include "gpio.h"
22
23/* Private includes -----
24/* USER CODE BEGIN Includes */
25#include "lcd.h"
26#include "led.h"
27#include "key.h"
28#include "stdio.h"
29
30/* USER CODE END Includes */
52    IO uint32_t lcd_uwtick=0;
53
54    //key相关变量 key_proc
55    uint32_t key_value;
56    uint32_t key_down;
57    uint32_t key_up;
58    uint32_t key_old;
59    __IO uint32_t key_uwtick=0;
60
61/* USER CODE END PV */
```

```

65 /* USER CODE BEGIN PFP */
66
67 void lcd_proc(void);
68 void key_proc(void);
69
70 /* USER CODE END PFP */
71
72 /* Private user code -----
73  * USER CODE BEGIN 1 */
74 while (1)
75 {
76     /* USER CODE END WHILE */
77
78     /* USER CODE BEGIN 3 */
79     //    led_disp(0x22);
80     //    HAL_Delay(500);
81     //    led_disp(0x00);
82     //    HAL_Delay(500);
83     lcd_proc();
84     key_proc();
85
86     LCD_UWTICK=UWTICK;
87     sprintf((char*)lcd_string, "hello");
88     LCD_DisplayStringLine(Line0, lcd_string);
89 }
90
91 void key_proc(void)
92 {
93     if(uwTick-key_uwtick<50) return;
94     key_uwtick=uwTick;
95     key_value=key_scan();
96     key_down=key_value&(key_old^key_value);
97     key_up=~key_value&(key_old^key_value);
98     key_old=key_value;
99
100     if(key_down!=0)
101     {
102         sprintf((char*)lcd_string, "key:%d", key_down);
103         LCD_DisplayStringLine(Line1, lcd_string);
104     }
105 }
106
107 /* 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

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	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Prefetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/38/39/40/41	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
BCC global interrupt	<input type="checkbox"/>	0	0
USART1 global interrupt / USART1 wake-up interrupt through EXTI line 25	<input checked="" type="checkbox"/>	1	0
FPU global interrupt	<input type="checkbox"/>	0	0

```

56 //key相关变量 key_proc
57 uint32_t key_value;
58 uint32_t key_down;
59 uint32_t key_up;
60 uint32_t key_old;
61 __IO uint32_t key_uwtick=0;
62
63 //uart 相关变量
64 uint8_t rx_buffer;
65 char str[50];
112 MX_USART1_UART_Init();
113 /* USER CODE BEGIN 2 */
114
115 LCD_Init();
116 LCD_Clear(Black);
117 LCD_SetTextColor(White);
118 LCD_SetBackColor(Black);
119
120 led_disp(0x00);
121
122 HAL_UART_Receive_IT(&huart1,&rx_buffer,1);
123
124 /* USER CODE END 2 */
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
218 {
219     if(rx_buffer=='a')
220     {
221         sprintf(str,"success\n");
222     }
223     else
224     {
225         sprintf(str,"error\n");
226     }
227     HAL_UART_Transmit(&huart1, (uint8_t*)str, strlen(str), 50);
228     HAL_UART_Receive_IT(&huart1,&rx_buffer,1);
229 }
230
231 /* USER CODE END 4 */
232
233

```

PA10 PA9

## 测试PWM

Pinout & Configuration      Clock Configuration      Project Manager

▼ Software Packs      ▼ Pinout

TIM3 Mode and Configuration

Mode

Slave Mode

Trigger Source

Clock Source

Channel1

Channel2

Channel3

Channel4

Combined Channels

Configuration

Reset Configuration

Parameter Settings    User Constants    NVIC Settings    DMA Settings    GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

▼ Counter Settings

Prescaler (PSC - 16 bits value)

Counter Mode

Dithering

Counter Period (AutoReload Register - 16 bit)

Internal Clock Division (CKD)

auto-reload preload

Clear Input Source

▼ PWM Generation Channel 2

Mode

Pulse (16 bits value)

Output compare preload

Fast Mode

CH Polarity

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

```

3 LCD_Clear(Black);
4 LCD_SetTextColor(White);
5 LCD_SetBackColor(Black);
6
7 led_disp(0x00);
8
9 HAL_UART_Receive_IT(&huart1,&rx_buffer,1);
10 HAL_TIM_PWM_Start(&htim3,TIM_CHANNEL_2);
11
12 /* USER CODE END 2 */
13
14 /* Infinite loop */
15 /* USER CODE BEGIN WHILE */

```

```
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
```