

```

1  #include "interrupt.h"
2
3  struct keys key[4] = {0};
4
5  uchar rx_array[50];
6  uchar rx_data;
7  uchar rx_pointer;
8
9  void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
10 {
11     if(huart->Instance == USART1)
12     {
13         rx_array[rx_pointer++] = rx_data;
14         HAL_UART_Receive_IT(huart, (uint8_t *)&rx_data, 1);
15     }
16 }
17 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
18 {
19     if(htim->Instance == TIM6)
20     {
21         key[0].value = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0);
22         key[1].value = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_1);
23         key[2].value = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2);
24         key[3].value = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
25
26         for(int i=0; i<4; i++)
27         {
28             switch(key[i].state)
29             {
30                 case 0:
31                     if(key[i].value == 0) key[i].state = 1;
32                     break;
33                 case 1:
34                     if(key[i].value == 0)
35                     {
36                         key[i].state = 2;
37                         key[i].click_time = 0;
38                     }
39                     else key[i].state = 0;
40                     break;
41                 case 2:
42                     if(key[i].value == 0)
43                         key[i].click_time++;
44                     else if(key[i].click_time>70)
45                     {
46                         key[i].long_flag = 1;
47                         key[i].state = 0;
48                     }
49                     else
50                     {
51                         switch(key[i].double_state)
52                         {
53                             case 0:
54                                 key[i].double_state = 1;
55                                 break;
56                             case 1:
57                                 key[i].double_state = 0;
58                                 key[i].double_flag = 1;
59                                 break;
60                         }
61
62                         key[i].state = 0;
63                         key[i].double_time = 0;
64                     }
65
66                     if(key[i].value == 0)
67                         key[i].click_time++;
68                     else
69                     {
70                         if(key[i].click_time >70)
71                             key[i].long_flag = 1;

```

```
72         else key[i].short_flag = 1;
73
74         key[i].state = 0;
75     }
76
77     break;
78
79 }
80
81 // if(key[i].double_state == 1)
82 // {
83 //     key[i].double_time++;
84 //     if(key[i].double_time>30)
85 //     {
86 //         key[i].short_flag = 1;
87 //         key[i].double_state = 0;
88 //     }
89 // }
90 // }
91 // }
92
93 }
```