

780EP CSDK 外设复用指南（使用EC718、716系列）

1. 外设复用函数

注意：780EP的外设复用与780E（618系列）不一样，函数不能通用

780EP依赖下面函数：

```
1  #include "luat_mcu.h"
2  /**
3   * @brief 用户控制外设的IOMUX，如果不配置或者取消，则外设初始化时使用默认配置
4   * @param type 外设类型 LUAT_MCU_PERIPHERAL_XXX
5   * @param sn 外设序号，0~7 比如要设置UART2，这里就填2，设置PWM0 这里就填0
6   * @param pad_index pad序号（要看模块的GPIO_table表），具体看芯片，可能是GPIO序号，可能是PAD序号。如果是-1，则表示取消配置
7   * @param alt 复用功能序号（要看模块的GPIO_table表），具体看芯片
8   * @param is_input，是否是单纯输入功能
9   * @return 无
10 */
11 void luat_mcu_iomux_ctrl(uint8_t type, uint8_t sn, int pad_index, uint8_t alt, uint8_t is_input);
```

2. 示例演示

注意：这里以GPIO1，复用为PWM0 为例

通过下图，可知，将GPIO1复用为PWM0时：pad 为16，alt为5，因此代码设置如下：

```
1  luat_mcu_iomux_ctrl(LUAT_MCU_PERIPHERAL_PWM,0,16,5,0);
```

Air780EP PIN Name		Pad Name	paddr [7:2]	Powerup default	Alt Func0	Alt Func1	Alt Func2	Alt Func3	Alt Func4	Alt Func5	Alt Func6
LDO_CTL	97	SWCLK0	11	I&PU	SWCLKA				GPIO16		G
IO_SEL	100	SWDIO0	12	I&PU	SWDIOA				GPIO17		G
I2C_SCL	67	SWCLK1	13	I&PU	SWCLKC		I2C0_SCL	I2C1_SCL	GPIO18	PWM0	KPC_R4
I2C_SDA	66	SWDIO1	14	I&PU	SWDIOC		I2C0_SDA	I2C1_SDA	GPIO19	PWM1	KPC_C4
USB_BOOT	82	GPIO0	15	I&PD	GPIO0						KPC_R4
UART1_CTS/AU_OUT	22/5	GPIO1	16	NI&NP	GPIO1		UART1_DCDn	UART1_RTSn	PWM1n	PWM0	KPC_R3
UART1_RTS	23	GPIO2	17	NI&NP	GPIO2		UART1_DTRn	UART1_CTSn	ONEW	PWM1	KPC_R2

3. 具体实例demo

```

1  #include "common_api.h"
2  #include "luat_rtos.h"
3  #include "luat_debug.h"
4
5  #include "luat_pwm.h"
6  #define PWM_ALT_FUNC 0
7
8  #include "luat_mcu.h"
9  /*
10     1. PWM的3、5通道不能使用
11  */
12  luat_rtos_task_handle pwm_task_handle;
13
14  static int32_t pwm_test_callback(void *pdata, void *param)
15  {
16      LUAT_DEBUG_PRINT("pwm done!");
17  }
18
19  static void task_test_pwm(void *param)
20  {
21      /*
22         出现异常后默认为死机重启
23         demo这里设置为LUAT_DEBUG_FAULT_HANG_RESET出现异常后尝试上传死机信息给PC
24         工具，上传成功或者超时报重启
25         如果为了方便调试，可以设置为LUAT_DEBUG_FAULT_HANG，出现异常后死机不重启
26         但量产出货一定要设置为出现异常重启!!!!!!!!!!!!1
27      */
28      luat_debug_set_fault_mode(LUAT_DEBUG_FAULT_HANG_RESET);
29      luat_rtos_task_sleep(2000);
30      #if (PWM_ALT_FUNC==1)//演示将PWM0 复用到GPIO1
31          uint8_t channel = 0;
32          luat_mcu_iomux_ctrl(LUAT_MCU_PERIPHERAL_PWM, channel, 16, 5, 0);
33      #else
34          uint8_t channel = 4;
35      #endif
36      luat_pwm_set_callback(channel, pwm_test_callback, NULL);
37      //测试13M, 50%占空比连续输出，看示波器
38      LUAT_DEBUG_PRINT("测试13MHz, 50占空比连续输出，看示波器");
39      luat_pwm_open(channel, 13000000, 500, 0);
40      luat_rtos_task_sleep(10000);
41      LUAT_DEBUG_PRINT("测试1Hz, 50占空比连续输出，输出10个波形停止");
42      luat_pwm_open(channel, 1, 500, 10);
43      luat_rtos_task_sleep(20000);
44      LUAT_DEBUG_PRINT("测试26KHz, 连续输出，占空比每5秒增加1，从0循环到100");
45      luat_pwm_open(channel, 26000, 0, 0);
46      uint32_t pulse_rate = 0;
47      while(1)

```

```

47     {
48         luat_rtos_task_sleep(5000);
49         pulse_rate += 10;
50         if (pulse_rate > 1000)
51         {
52             pulse_rate = 0;
53         }
54         LUAT_DEBUG_PRINT("当前占空比%u", pulse_rate/10);
55         luat_pwm_update_dutycycle(channel, pulse_rate);
56     }
57
58 }
59
60 static void task_demo_pwm(void)
61 {
62     luat_rtos_task_create(&pwm_task_handle, 2048, 20, "pwm", task_test_pwm
63 , NULL, 0);
64 }
65
66 INIT_TASK_EXPORT(task_demo_pwm, "1");

```