### 13. 7 PWM输出方波的例程

本节描述了在微芯PIC32MX220F032B型芯片上的PWM输出示例。通过4个PWM输出通道输出循环变化占空比的PWM方波，调整4个LED灯的亮度，达到“呼吸灯”的效果。

适用范围：本节所描述的代码适用于PIC32MX220F032B型芯片（28 引脚SOIC封装），对于其他型号或封装的芯片，未经测试，不确定其可用性。

表13-1 PWM输出引脚选择硬件配置表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能符号 | 引脚号 | 复用端口选择指定功能所需代码 | 说明 |
| 1 | RPB7 | 16 | RPB7Rbits.RPB7R = 0b0101 | 复用引脚RPB7，配置为OC1输出 |
| 2 | RPB8 | 17 | RPB8Rbits.RPB8R = 0b0101 | 复用引脚RPB8，配置为OC2输出 |
| 3 | RPB9 | 18 | RPB9Rbits.RPB9R = 0b0101 | 复用引脚RPB9，配置为OC3输出 |
| 4 | RPB13 | 24 | RPB13Rbits.RPB13R = 0b0101 | 复用引脚RPB13，配置为OC4输出 |

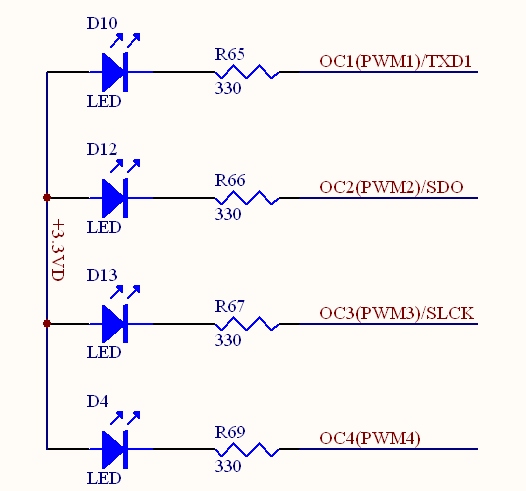


图13-2：PWM输出硬件接口示意图

图13-3：主函数流程框图 图13-4：PWM初始化函数流程框图 图13-5：定时器中断函数流程框图

**1、主函数例程（程序流程框图见图13-3）**

|  |
| --- |
| int **main**(void)  {  SYSTEMConfig(SYS\_FREQ, SYS\_CFG\_WAIT\_STATES | SYS\_CFG\_PCACHE);  INTDisableInterrupts();  INTConfigureSystem(INT\_SYSTEM\_CONFIG\_MULT\_VECTOR);  PWMinit();  Timer1Init();  INTEnableInterrupts();  while(1)  ;  return 0;  } |

**2、PWM初始化函数例程（程序流程框图见图13-4）**

|  |
| --- |
| void **PWMinit**()  {  //PWM引脚关联  RPB7Rbits.RPB7R = 0b0101; //PWM1  RPB8Rbits.RPB8R = 0b0101; //PWM2  RPB9Rbits.RPB9R = 0b0101; //PWM3  RPB13Rbits.RPB13R = 0b0101; //PWM4  //PWM1初始化  OC1CON = 0x0000; // Turn off OC1 while doing setup.  OC1RS = pwm1; // Initialize secondary Compare Register  OC1CON = 0x0006; // Configure for PWM mode  //PWM2初始化  OC2CON = 0x0000; // Turn off OC1 while doing setup.  OC2RS = pwm1; // Initialize secondary Compare Register  OC2CON = 0x0006; // Configure for PWM mode  //PWM3初始化  OC3CON = 0x0000; // Turn off OC1 while doing setup.  OC3RS = pwm1; // Initialize secondary Compare Register  OC3CON = 0x0006; // Configure for PWM mode  //PWM4初始化  OC4CON = 0x0000; // Turn off OC1 while doing setup.  OC4RS = pwm1; // Initialize secondary Compare Register  OC4CON = 0x0006; // Configure for PWM mode  //定时器2周期设定+开启  PR2 = PWM\_PR; // Set period  T2CONSET = 0x8000; // 使能 Timer2  //PWM1~4开启  OC1CONSET = 0x8000; // 使能 OC1  OC2CONSET = 0x8000; // 使能 OC2  OC3CONSET = 0x8000; // 使能 OC3  OC4CONSET = 0x8000; // 使能 OC4  } |

**3、定时器中断函数例程（程序流程框图见图13-5）**

|  |
| --- |
| void \_\_ISR(\_TIMER\_1\_VECTOR, ipl2) **Timer1Handler**(void)  {  // Clear the interrupt flag  INTClearFlag(INT\_T1);  if(pwm1\_d == 0)  {  pwm1 ++;  if(pwm1 > PWM\_PR)  {  pwm1 = PWM\_PR;  pwm1\_d = 1;  }  }  else  {  if(pwm1 == 0)  {  pwm1 = 0;  pwm1\_d = 0;  }  else  pwm1 --;  }  OC1RS = pwm1;  OC2RS = pwm1;  OC3RS = pwm1;  OC4RS = pwm1;  } |

### 附件：代码

|  |
| --- |
| /\* File: PWMExample.c  \*/  #include <plib.h>  // Configuration Bit settings  // SYSCLK = 48 MHz (8MHz Crystal / FPLLIDIV \* FPLLMUL / FPLLODIV)  // PBCLK = 48 MHz (SYSCLK / FPBDIV)  // Primary Osc w/PLL (XT+,HS+,EC+PLL)  // WDT OFF  #pragma config FPLLMUL = MUL\_24, FPLLIDIV = DIV\_2, FPLLODIV = DIV\_2, FWDTEN = OFF  #pragma config POSCMOD = OFF, FNOSC = FRCPLL, FPBDIV = DIV\_1, FSOSCEN = OFF  #pragma config FUSBIDIO = OFF //FUSBIDIO为端口控制  #pragma config JTAGEN = OFF //JTAG disable  // Period needed for timer 1 to trigger an interrupt every 0.1 second  // (48MHz PBCLK / 1 = 48000000KHz Timer 1 clock)  #define PERIOD 9600 //48000/48000000 = 0.001s = 0.2ms  #define BTN\_DELAY 2 //2\*5=10ms  #define SYS\_FREQ (48000000L)  #define PWM\_PR 0x0FFF //PWM周期  UINT16 pwm1=0,pwm1\_d=0;  void **Timer1Init**()  {  // Timer1@1ms  OpenTimer1(T1\_ON | T1\_SOURCE\_INT | T1\_PS\_1\_1, PERIOD);  // Set up the timer interrupt with a priority of 2  INTEnable(INT\_T1, INT\_ENABLED);  INTSetVectorPriority(INT\_TIMER\_1\_VECTOR, INT\_PRIORITY\_LEVEL\_2);  INTSetVectorSubPriority(INT\_TIMER\_1\_VECTOR, INT\_SUB\_PRIORITY\_LEVEL\_0);  }  void \_\_ISR(\_TIMER\_1\_VECTOR, ipl2) **Timer1Handler**(void)  {  // Clear the interrupt flag  INTClearFlag(INT\_T1);  if(pwm1\_d == 0)  {  pwm1 ++;  if(pwm1 > PWM\_PR)  {  pwm1 = PWM\_PR;  pwm1\_d = 1;  }  }  else  {  if(pwm1 == 0)  {  pwm1 = 0;  pwm1\_d = 0;  }  else  pwm1 --;  }  OC1RS = pwm1;  OC2RS = pwm1;  OC3RS = pwm1;  OC4RS = pwm1;  }  void **PWMinit**()  {  //PWM引脚关联  RPB7Rbits.RPB7R = 0b0101; //PWM1  RPB8Rbits.RPB8R = 0b0101; //PWM2  RPB9Rbits.RPB9R = 0b0101; //PWM3  RPB13Rbits.RPB13R = 0b0101; //PWM4  //PWM1初始化  OC1CON = 0x0000; // Turn off OC1 while doing setup.  OC1RS = pwm1; // Initialize secondary Compare Register  OC1CON = 0x0006; // Configure for PWM mode  //PWM2初始化  OC2CON = 0x0000; // Turn off OC1 while doing setup.  OC2RS = pwm1; // Initialize secondary Compare Register  OC2CON = 0x0006; // Configure for PWM mode  //PWM3初始化  OC3CON = 0x0000; // Turn off OC1 while doing setup.  OC3RS = pwm1; // Initialize secondary Compare Register  OC3CON = 0x0006; // Configure for PWM mode  //PWM4初始化  OC4CON = 0x0000; // Turn off OC1 while doing setup.  OC4RS = pwm1; // Initialize secondary Compare Register  OC4CON = 0x0006; // Configure for PWM mode  //定时器2周期设定+开启  PR2 = PWM\_PR; // Set period  T2CONSET = 0x8000; // 使能 Timer2  //PWM1~5开启  OC1CONSET = 0x8000; // 使能 OC1  OC2CONSET = 0x8000; // 使能 OC2  OC3CONSET = 0x8000; // 使能 OC3  OC4CONSET = 0x8000; // 使能 OC4  }  int **main**(void)  {  SYSTEMConfig(SYS\_FREQ, SYS\_CFG\_WAIT\_STATES | SYS\_CFG\_PCACHE);  INTDisableInterrupts();  INTConfigureSystem(INT\_SYSTEM\_CONFIG\_MULT\_VECTOR);  PWMinit();  Timer1Init();  INTEnableInterrupts();  while(1)  ;  return 0;  } |