|  |  |  |  |
| --- | --- | --- | --- |
| 姓名：欧阳芳霞 | 专业：物联网工程 | 班级：20181191 | 学号：2018119129 |
| 科目：嵌入式系统概论 | | 实验日期：2020.1.1 | |
| 实验题目：Led呼吸 | | | |
| 【实验目的】   控制LED灯的呼吸 | | | |
| 【实验内容】  主要代码：  main.c  #include "main.h"  #include "tim.h"  #include "gpio.h"  void SystemClock\_Config(void);  int main(void)  {  uint16\_t pwmVal=0;  uint8\_t dir=1;  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  MX\_TIM2\_Init();  HAL\_TIM\_PWM\_Start(&htim2,TIM\_CHANNEL\_1);  while (1)  {  while(pwmVal<500)  {  pwmVal++;  \_\_HAL\_TIM\_SET\_COMPARE(&htim2,TIM\_CHANNEL\_1,pwmVal);  HAL\_Delay(1);  }  while(pwmVal)  {  pwmVal--;  \_\_HAL\_TIM\_SET\_COMPARE(&htim2,TIM\_CHANNEL\_1,pwmVal);  HAL\_Delay(1);  }  HAL\_Delay(200);  }  }  void SystemClock\_Config(void)  {  RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};  RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};  RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSE;  RCC\_OscInitStruct.HSEState = RCC\_HSE\_ON;  RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_ON;  RCC\_OscInitStruct.PLL.PLLSource = RCC\_PLLSOURCE\_HSE;  RCC\_OscInitStruct.PLL.PLLM = 1;  RCC\_OscInitStruct.PLL.PLLN = 20;  RCC\_OscInitStruct.PLL.PLLP = RCC\_PLLP\_DIV7;  RCC\_OscInitStruct.PLL.PLLQ = RCC\_PLLQ\_DIV2;  RCC\_OscInitStruct.PLL.PLLR = RCC\_PLLR\_DIV2;  if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)  {  Error\_Handler();  }  RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK  |RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;  RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_PLLCLK;  RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;  RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV2;  RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;  if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_4) != HAL\_OK)  {  Error\_Handler();  }  if (HAL\_PWREx\_ControlVoltageScaling(PWR\_REGULATOR\_VOLTAGE\_SCALE1) != HAL\_OK)  {  Error\_Handler();  }  }  #ifdef USE\_FULL\_ASSERT  #endif /\* USE\_FULL\_ASSERT \*/  gpio.c  #include "gpio.h"  void MX\_GPIO\_Init(void)  {  GPIO\_InitTypeDef GPIO\_InitStruct = {0};  \_\_HAL\_RCC\_GPIOC\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();  HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  GPIO\_InitStruct.Pin = GPIO\_PIN\_13;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  HAL\_GPIO\_Init(GPIOC, &GPIO\_InitStruct);  }  tim.c  #include "tim.h"  TIM\_HandleTypeDef htim2;  void MX\_TIM2\_Init(void)  {  TIM\_ClockConfigTypeDef sClockSourceConfig = {0};  TIM\_MasterConfigTypeDef sMasterConfig = {0};  TIM\_OC\_InitTypeDef sConfigOC = {0};  htim2.Instance = TIM2;  htim2.Init.Prescaler = 79;  htim2.Init.CounterMode = TIM\_COUNTERMODE\_UP;  htim2.Init.Period = 499;  htim2.Init.ClockDivision = TIM\_CLOCKDIVISION\_DIV1;  htim2.Init.AutoReloadPreload = TIM\_AUTORELOAD\_PRELOAD\_DISABLE;  if (HAL\_TIM\_Base\_Init(&htim2) != HAL\_OK)  {  Error\_Handler();  }  sClockSourceConfig.ClockSource = TIM\_CLOCKSOURCE\_INTERNAL;  if (HAL\_TIM\_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL\_OK)  {  Error\_Handler();  }  if (HAL\_TIM\_PWM\_Init(&htim2) != HAL\_OK)  {  Error\_Handler();  }  sMasterConfig.MasterOutputTrigger = TIM\_TRGO\_RESET;  sMasterConfig.MasterSlaveMode = TIM\_MASTERSLAVEMODE\_DISABLE;  if (HAL\_TIMEx\_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL\_OK)  {  Error\_Handler();  }  sConfigOC.OCMode = TIM\_OCMODE\_PWM1;  sConfigOC.Pulse = 0;  sConfigOC.OCPolarity = TIM\_OCPOLARITY\_LOW;  sConfigOC.OCFastMode = TIM\_OCFAST\_DISABLE;  if (HAL\_TIM\_PWM\_ConfigChannel(&htim2, &sConfigOC, TIM\_CHANNEL\_1) != HAL\_OK)  {  Error\_Handler();  }  HAL\_TIM\_MspPostInit(&htim2);  }  void HAL\_TIM\_Base\_MspInit(TIM\_HandleTypeDef\* tim\_baseHandle)  {  if(tim\_baseHandle->Instance==TIM2)  {  \_\_HAL\_RCC\_TIM2\_CLK\_ENABLE();  HAL\_NVIC\_SetPriority(TIM2\_IRQn, 0, 0);  HAL\_NVIC\_EnableIRQ(TIM2\_IRQn);  }  }  void HAL\_TIM\_MspPostInit(TIM\_HandleTypeDef\* timHandle)  {  GPIO\_InitTypeDef GPIO\_InitStruct = {0};  if(timHandle->Instance==TIM2)  {    \_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();  GPIO\_InitStruct.Pin = GPIO\_PIN\_0;  GPIO\_InitStruct.Mode = GPIO\_MODE\_AF\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  GPIO\_InitStruct.Alternate = GPIO\_AF1\_TIM2;  HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);  }  }  void HAL\_TIM\_Base\_MspDeInit(TIM\_HandleTypeDef\* tim\_baseHandle)  {  if(tim\_baseHandle->Instance==TIM2)  {    \_\_HAL\_RCC\_TIM2\_CLK\_DISABLE();    HAL\_NVIC\_DisableIRQ(TIM2\_IRQn);    }  } | | | |
| 【实验结果】  运行时led像呼吸般亮暗  【实验结论】（写结论，心得，和收获）  实验结论、心得和收获：  ①掌握了 Led 驱动电路及开关 Led 的原理  ②掌握了呼吸灯原理  ③基本了解了实验的操作过程 | | | |
| **【**教师评语和成绩**】**  **成绩：** **指导教师：** **日期：** | | | |