|  |  |  |  |
| --- | --- | --- | --- |
| 112 2 嵌入式作業系統分析與實作 Lab Report | | | |
| Lab Date: | 4/18 | Lab No: | Lab 3 |
| 學號: | P76121110 | 姓名: | 張家菖 |
| Q1 簡述這次lab實驗內容:  Normal: 只有綠色LED燈在閃爍  當晃動板子，trigger sensor interrupt:  ISR會給semaphore讓handler task醒來  ISR會啟動/關閉紅色LED，但若是橘色LED在閃爍時晃動，則無效  handler task的任務是讓橘色LED閃爍5次  當橘色LED燈在閃爍時，此時若晃動板子不可 trigger sensor interrupt | | | |
| Q2 簡述這次lab遇到的困難或是完成心得:  #define WAKEUP\_STATE\_MACHINE\_CONFIG \  MEMS\_Write(0x21, 0x01); \  MEMS\_Write(0x23, 0x48); \  MEMS\_Write(0x20, 0x67); \  MEMS\_Write(0x24, 0x00); \  MEMS\_Write(0x57, 0x55); \  MEMS\_Write(0x40, 0x05); \  MEMS\_Write(0x41, 0x11); \  MEMS\_Write(0x59, 0xfc); \  MEMS\_Write(0x5a, 0xfc); \  MEMS\_Write(0x5b, 0x01); \  WAKEUP\_STATE\_MACHINE\_CONFIG 將register的值都恢復到 Wake-up state machine一開始的設定  /\*\*  \* @brief EXTI line detection callbacks.  \* @param GPIO\_Pin Specifies the pins connected EXTI line  \* @retval None  \*/  \_\_weak void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)  {  /\* Prevent unused argument(s) compilation warning \*/  UNUSED(GPIO\_Pin);  /\* NOTE: This function Should not be modified, when the callback is needed,  the HAL\_GPIO\_EXTI\_Callback could be implemented in the user file  \*/  }  void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)  {  /\* toggle Red LED \*/  HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_14);  /\* Give the semaphore to unblock the handler task \*/  BaseType\_t xHigherPriorityTaskWoken = pdFALSE;  xSemaphoreGiveFromISR( xSemaphore, &xHigherPriorityTaskWoken );  portYIELD\_FROM\_ISR( xHigherPriorityTaskWoken );  }  \_\_weak 是一個 GCC (GNU Compiler Collection) 的 extension，它用來標記一個 function，表示這個 function 是一個 weak symbol，這種標記通常用於允許 programmer 重新定義該function，並且這種重新定義不會導致linking error。如果這個function沒有被使用者重新定義，則使用預設的function。  void vHandlerTask( void \*pvParameters ){……  // reset interrupt register  MEMS\_Read(LIS3DSH\_OUTS1\_ADDR, &data);  // WAKEUP\_STATE\_MACHINE\_CONFIG // 也可 }  最後記得要reset interrupt registers，否則只能發出一次interrupt | | | |
| Q3 其他(optional):  // FreeRTOS/include/FreeRTOS.h  #define portTICK\_RATE\_MS portTICK\_PERIOD\_MS  // FreeRTOS/portable/ARM\_CM4F/portmacro.h  #define portTICK\_PERIOD\_MS ( ( TickType\_t ) 1000 / configTICK\_RATE\_HZ )  // FreeRTOS/include/FreeRTOSConfig.h  #define configTICK\_RATE\_HZ ( ( TickType\_t ) 1000 )  Priority的範圍是 0 ~ (configMAX\_PRIORITIES - 1)  // FreeRTOS/include/FreeRTOSConfig.h  #define configMAX\_PRIORITIES ( 5 ) // 預設是5 | | | |
|  | | | |