Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

CEG4566/CSI4541/SEG4545

Conception de systèmes informatiques en temps réel

Hiver 2024

Professeur : Mohamed Ali Ibrahim, ing., Ph.D.

Notifications de tâches

Source: https://www.freertos.org/RTOS-task-notifications.html

Notifications: Introduction

- Les notifications permettent aux tâches de communiquer directement entre elles sans passer par des objets de communication tels que les files d'attente, les sémaphores, les groupes d'événements.

Notifications: Activation

Active

configUSE_TASK_NOTIFICATIONS 1

Dans le fichier FreeRTOSConfig.h

Notifications: États

- En attente

  Lorsqu'une tâche reçoit une notification, son état de notification est mis en attente.

- Non en attente

  Lorsqu'une tâche lit la valeur de sa notification, l'état de sa notification est défini comme "non en attente".

Notifications: Avantages

- Plus rapide que l'utilisation de files d'attente, de sémaphores et de groupes d'événements pour effectuer une opération équivalente.
- Nécessite beaucoup moins de mémoire vive qu'une file d'attente, un sémaphore ou un groupe d'événements.

Notifications: Limitations

- Contrairement aux files d'attente, aux sémaphores et aux groupes d'événements, les notifications de tâches ne peuvent pas être utilisées pour envoyer des événements ou des données à un ISR.

- Contrairement aux files d'attente, aux sémaphores et aux groupes d'événements, les notifications de tâches ne peuvent pas être envoyées à plusieurs tâches.

```c
int main(void)
{
   HAL_Init();
   SystemClock_Config();
   MX_GPIO_Init();
   USART2_UART_TX_Init();
   p13_interrupt();
   printf("Hello from stm32f446\r\n");
   xTaskCreate(HandlerTask,"Handler Task",100,NULL,3,&xTaskHandler);
   vTaskStartScheduler();
   while (1)
   {
   }
}
```

```
void HandlerTask(void *pvParameters)
{
   const TickType_t xMaxExpectedBlockTime= pdMS_TO_TICKS(100);
   while(1)
   {
      if((ulTaskNotifyTake(pdFALSE, xMaxExpectedBlockTime)) != 0)
      {
         printf("Handler task ------------ processing events\r\n");
      }
   }
}
```

```c
void EXTI15_10_IRQHandler(void)
{
   if((EXTI->PR & EXTI_PR_PR13) != 0)
   {
      /*Clear PR flag*/
      EXTI->PR |= EXTI_PR_PR13;
      //Do something...
      BaseType_t xHigherPriorityTaskWoken = pdFALSE;
      vTaskNotifyGiveFromISR(xTaskHandler, &xHigherPriorityTaskWoken);
      portYIELD_FROM_ISR(xHigherPriorityTaskWoken);
   }
}
```