

Laborationsrapport nr.4

av Lasse Karagiannis

2016-01-28

1. Uppgiften

Uppgiften är att få LED2 på kortet NUCLEO-411RE att blinka en gång per sekund, med användning av FreeRTOS, samt besvara frågorna

- Det finns två olika anrop som pausar ett task. Vilka är det och vad är det för skillnad mellan dem?
- Vad använder FreeRTOS sig av för typstandard för enkla typer?
- Vad betyder `x i xTaskCreate` och `v i vTaskDelete`?

2. Tillvägagångssätt

Atolic Studio används som är ett utvecklingssystem byggt på Eclipse IDE. Kodgenerering av FreeRTOS specifika filer görs av STM32CubeMX.

Applikationskoden initiera ett task och startar operativsystemet.

Både Atolic Studio och Cube är buggiga program.

3. Applikationskoden

```
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();

    osThreadDef(defaultTask, StartDefaultTask, osPriorityNormal, 0, 128);
    defaultTaskHandle = osThreadCreate(osThread(defaultTask), NULL);
    vOtherFunction();

    osKernelStart();

    /* We should never get here as control is now taken by the scheduler */

    while (1);
}
```

Där `vOtherFunction` är den enda koden som inte automatgenererats.

```
void vTaskCode( void * pvParameters )
{
    TickType_t xLastWakeTime;
    const TickType_t xFrequency = 1;
    xLastWakeTime = xTaskGetTickCount();

    for( ;; )
    {
        // Wait for the next cycle.
        vTaskDelayUntil( &xLastWakeTime, xFrequency );
        /* Task code goes here. */
        HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
    }
}

void vOtherFunction( void )
{
    static unsigned char ucParameterToPass;
    TaskHandle_t xHandle = NULL;

    xTaskCreate( vTaskCode,
                "NAME",
                STACK_SIZE,
                &ucParameterToPass,
                tskIDLE_PRIORITY + 1,
                &xHandle );
}
```

4. Svaren på ställda frågor

Det finns två olika anrop som pausar ett task. Vilka är det och vad är det för skillnad mellan dem?

De två typerna är av "delay" och "suspend"

Det ges två stycken delay funktioner

```
void vTaskDelay( const TickType_t xTicksToDelay );  
samt  
void vTaskDelayUntil( TickType_t *pxPreviousWakeTime,  
const TickType_t xTimeIncrement );
```

Dokumentationen säger följande:

Delay a task until a specified time. This function can be used by periodic tasks to ensure a constant execution frequency.

This function differs from vTaskDelay() in one important aspect: vTaskDelay() specifies a time at which the task wishes to unblock relative to the time at which vTaskDelay() is called, whereas vTaskDelayUntil() specifies an absolute time at which the task wishes to unblock.

Detta betyder att `vTaskDelay()` anger tiden för unblock relativt anropningstidpunkten, dvs. den anger tiden för när den är villig att köras, relativt dess anrop, medan `vTaskDelayUntil()` anger i absolut tid när den tillåter att den körs.

Jag hoppas verkligen att denna blockering inte blockerar processorn utan endast är en övilja att låta sig exekveras.

Suspend –varianten implementeras med en funktion

```
void vTaskSuspend( TaskHandle_t xTaskToSuspend );
```

Dokumentationen anger följande:

Suspend any task. When suspended a task will never get any microcontroller processing time, no matter what its priority.

Detta betyder att schemaläggaren kommer inte att kunna ge denna körtid oavsett dess prioritet.

Vad använder FreeRTOS sig av för typstandard för enkla typer?

För att FreeRTOS ska kunna kompileras med så många olika kompilatorer som möjligt, så använder de ett minimum av inbyggda typer som kan skilja från kompilator till kompilator. Detta enda som används, och som förmodas vara gemensamt för samtliga kompilatorer på marknaden är biblioteket `stdint`.

Dokumentationen anger följande avseende `stdint`

Variables of type `uint32_t` are prefixed `ul`, where the 'u' denotes 'unsigned' and the 'l' denotes 'long'.

Variables of type `uint16_t` are prefixed `us`, where the 'u' denotes 'unsigned' and the 's' denotes 'short'.

Variables of type `uint8_t` are prefixed `uc`, where the 'u' denotes 'unsigned' and the 'c' denotes 'char'.

Vad betyder x i xTaskCreate och v i vTaskDelete?

x anger att returtypen inte är att finna i stdint. Dokumentationen anger följande

Variables of non stdint types are prefixed x. Examples include BaseType_t and TickType_t, which are portable layer defined typedefs for the natural or most efficient type for the architecture and the type used to hold the RTOS tick count respectively.

Bokstaven v I början av ett funktionsnamn är att den returnerar void.