

#### 4. freeRTOS の使い方

[STM32F4DISCOVERY/STM32CubeIDE で、freeRTOS を使う]

2020年8月29日

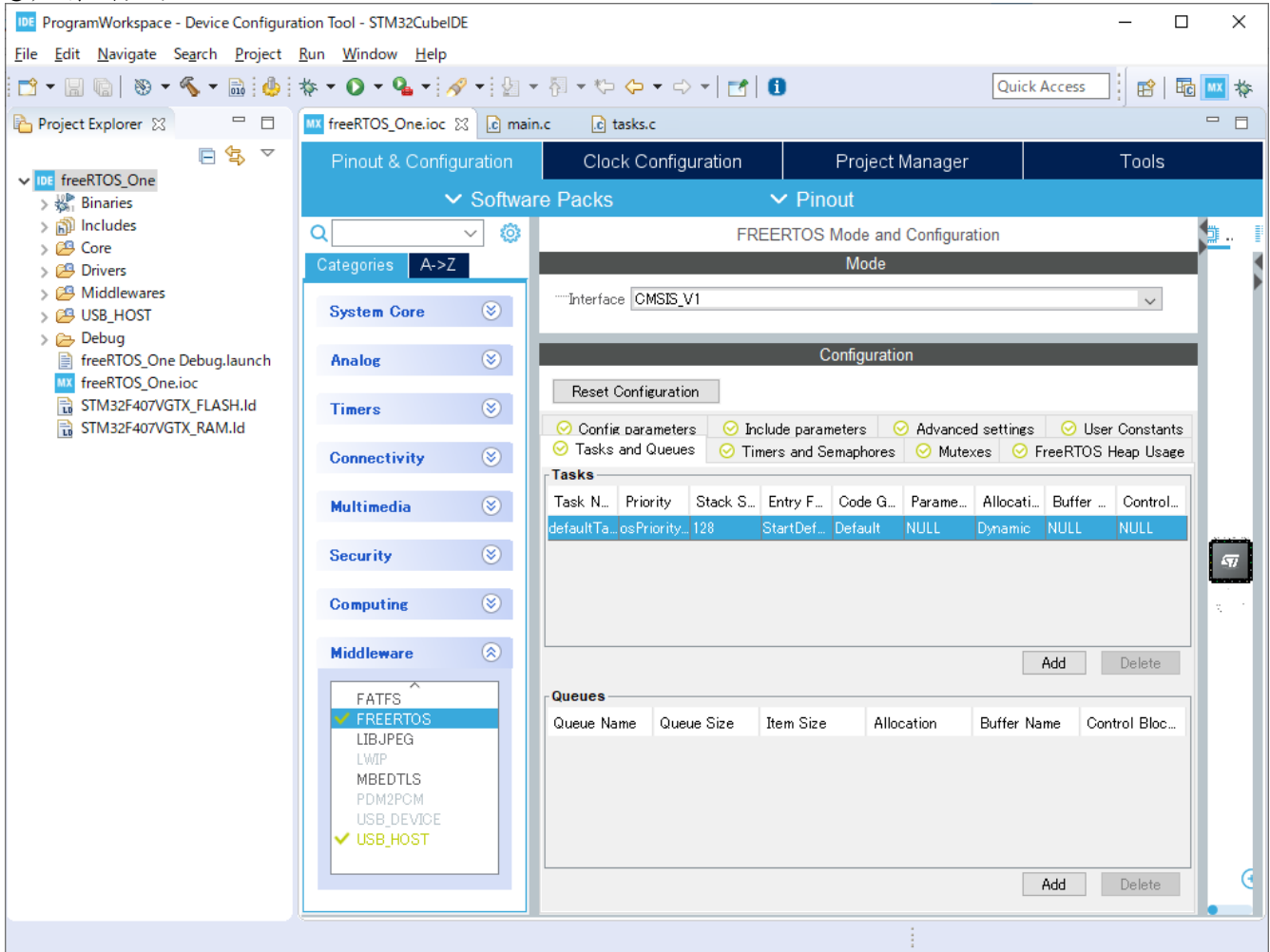
freeRTOS に、CMSIS のラップがかかっている (freeRTOS とは、API が違う)。  
API は CMSIS-RTOS だが、仕様は CMSIS-RTOS とは違う部分もあるらしい。

##### ■タスク

DefaultTask が最初から用意されているので、それを使ってみる。  
タスクの実態は、Core\Src\main.c につぎのように生成されている。

```
/* USER CODE BEGIN Header_StartDefaultTask */
/**
 * @brief Function implementing the defaultTask thread.
 * @param argument: Not used
 * @retval None
 */
/* USER CODE END Header_StartDefaultTask */
void StartDefaultTask(void const * argument)
{
    /* init code for USB_HOST */
    MX_USB_HOST_Init();
    /* USER CODE BEGIN 5 */
    /* Infinite loop */
    for(;;)
    {
        osDelay(1);
    }
    /* USER CODE END 5 */
}
```

## ● デフォルトタスク



## ● ふたつ目のタスクの追加(タスク2)

Edit Task

Task Name	myTask02
Priority	osPriorityHigh
Stack Size (Words)	128
Entry Function	StartTask02
Code Generation Option	Default
Parameter	NULL
Allocation	Dynamic
Buffer Name	NULL
Control Block Name	NULL

OK Cancel

※優先度を高く(osPriorityHigh)している。

## ■ 時間待ち

osDelay(1); ms 単位。

## ■ シグナル(イベントフラグの代わり?)

```
int32_t osSignalClear( osThreadId thread_id, int32_t signals )
int32_t osSignalSet( osThreadId thread_id, int32_t signals)
osEvent osSignalWait(int32_t signals, uint32_t millisec)
```

(TRON 系フラグとの違い)

- ・指定したタスクにしか発生できない(タスク付属型)。