

Primer 1: Napraviti dva zadatka istog prioriteta da šalju podatke preko UART-a

```
void task1(void *argument)
{
    char txt[] = "Task1: Hello world!\n";
    for(;;)
    {
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        osDelay(1000);
    }
}
```

```
void task2(void *argument)
{
    char txt[] = "Task2: Hello world!\n";
    for(;;)
    {
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        osDelay(1000);
    }
}
```

Primer 2: Evolutivno doći do ovog primera

- Prvo ubaciti umesto HAL_Delay osDelay i uvideti problem
- Onda dodati blokiranje zadataka
- Pokazati kako se onda koristi osDelayUntil

```
void task1(void *argument)
{
    char txt[] = "Task1: Hello world!\n";
    uint32_t ticks = osKernelGetTickCount();
    for(;;)
    {
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        osThreadResume(Task2Handle);
        ticks+=1000;
        osDelayUntil(ticks);
    }
}
```

```
void task2(void *argument)
{
    char txt[] = "Task2: Hello world!\n";
    for(;;)
    {
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        osThreadSuspend(Task2Handle);
    }
}
```

Primer3:

```
void task1(void *argument)
{
    char txt[] = "Task1: Hello world!\n";
    for(;;)
    {
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        osThreadSetPriority(Task2Handle, osPriorityLow2);
        HAL_Delay(1000);
    }
}
void task2(void *argument)
{
    char txt[] = "Task2: Hello world!\n";
    for(;;)
    {
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        osThreadSetPriority(Task2Handle, osPriorityLow);
    }
}
```

Primer 4: Uključiti FreeRTOS Config-> USE_IDLE_HOOK

```
void task1(void *argument)
{
    char txt[20];
    uint32_t ticks = osKernelGetTickCount();
    for(;;)
    {
        sprintf(txt,"Trenutno %lu\n",cnt);
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        ticks+=1000;
        osDelayUntil(ticks);
    }
}
```

main.h

extern uint32_t cnt;

freeRTOS.c

uint32_t cnt=0;

```
void vApplicationIdleHook( void )
{
    cnt++;
}
```

Primer 5:

```
void task1(void *argument)
{
char txt[]="Task1: Hello World\n";
uint32_t ticks = osKernelGetTickCount();
for(;;)
{
    HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
    ticks+=1000;
    osDelayUntil(ticks);
    osThreadNew(task2,Task2Handle, NULL);
}
}
```

```
void task2(void *argument)
{
    char txt[]="Task2: Hello World\n";
    for(;;)
    {
        HAL_UART_Transmit(&huart2, (uint8_t *)txt, strlen(txt), HAL_MAX_DELAY);
        osThreadExit();
    }
}
```