Программирование микроконтроллеров STM32

FreeRTOS and Tasks

Концепция ОС



Credits: freertos.org

М. •мальная ОС. Куда двиг зся дальше?

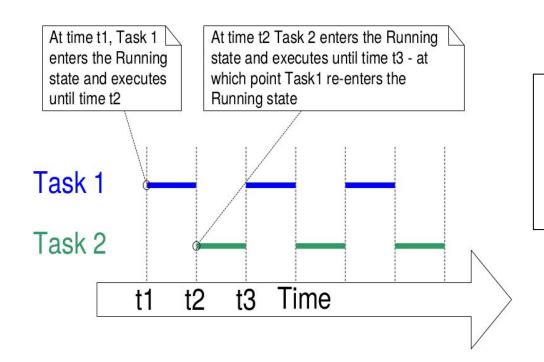
- 1. Добавление приор. задачам
- 2. Добавление менед
- 3. Добавление сремежпр эного взаимодействия
- 4. Добавление держки систем зовов для удобной работы с драйвер де имеет смысла для (-M0)
- 5. Анал' мингов



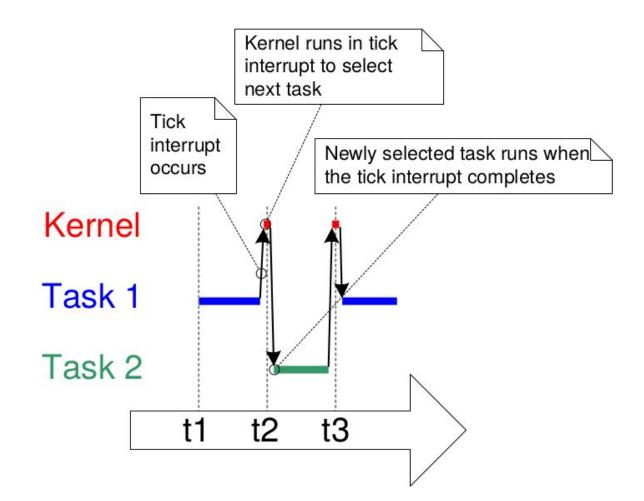
FreeRTOS. Планировщик

```
BaseType_t xTaskCreate(TaskFunction_t pvTaskCode,
                        const char *const pcName,
                        uint16_t usStackDepth,
                        void *pvParameters,
                        UBaseType_t uxPriority,
                        TaskHandle_t *pxCreatedTask );
void ATaskFunction(void *pvParameters);
```

FreeRTOS. Планировщик



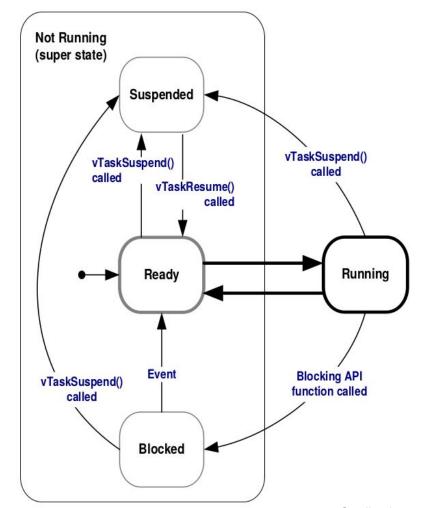
Задачи не могут выполняться параллельно на одном ядре микроконтроллера!



FreeRTOS.
Планировщик

FreeRTOS. Планировщик

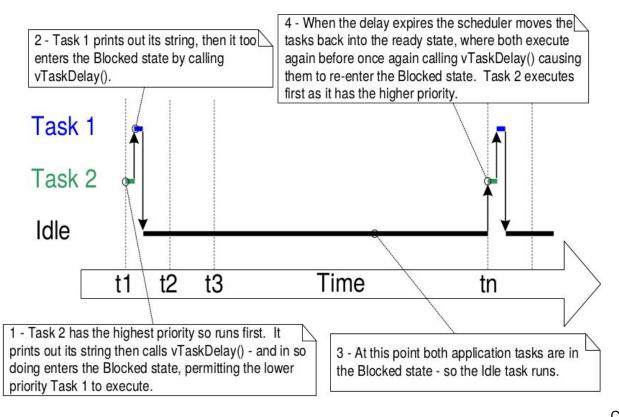
- Блокированное состояние (blocked state) - задержка или синхронизация
- Приостановленное состояние (suspended state) - задача недоступна планировщику



Credits: https://freertos.org/

- От 0 до (configMAX_PRIORITIES 1)
- Выше численный приоритет выше логический приоритет
- Выше приоритет больше времени на исполнение

```
void vTaskFunction(void *pvParameters)
   char *pcTaskName;
   const TickType_t xDelay250ms = pdMS_TO_TICKS(250);
   for (;;) {
      vPrintString(pcTaskName);
      vTaskDelay(xDelay250ms);
```



Credits: https://freertos.org/

state. The Periodic task is the highest priority task so immediately then enters the Running state where it prints out its string exactly once before calling vTaskDelayUntil() to return to the Blocked state. 5 - The Periodic task entering the 1 - Continuous task 1 runs for a Blocked state means the scheduler has complete tick period (time slice again to choose a task to enter the between times t1 and t2) - during Running state - in this case Continuous which time it could print out its 1 is chosen and it runs up to the next tick string many times. interrupt - during which time it could print out its string many times. Periodic Continuous 1 The Idle task never enters the Running state as there are Continuous 2 always higher priority task that are able to do so. Idle t2 t3 Time t5 **t1** 2 - The tick interrupt occurs during which the scheduler selects a new task to run. As both 3 - At time t3 the tick interrupt Continuous tasks have the same priority and runs again, causing a switch back both are always able to run the scheduler to Continuous 1, and so it goes shares processing time between the two - so on. Continuous 2 enters the Running state where it remains for the entire tick period - during which

Credits: https://freertos.org/

time it could print out its string many times.

4 - At time t5 the tick interrupt finds that the Periodic task block period has expired so moved the Periodic task into the Ready

FreeRTOS. Пустая задача (idle task)

- Бесконечный цикл, всегда в состоянии Ready
- Самый низкий приоритет
- Можно использовать для выполнения фоновый задач
- Не должна находиться в состоянии Blocked или Suspended (иначе не будет ни одной задачи на выполнение)
- void vApplicationIdleHook(void);

FreeRTOS. Алгоритмы планировщика

configUSE_PREEMPTION	configUSE_TIME_SLICING	Алгоритм
1	1	Prioritized Pre-emptive Scheduling with Time Slicing
1	0	Prioritized Pre-emptive Scheduling (without Time Slicing)
0	Any	Co-operative Scheduling

Репозиторий

https://github.com/edosedgar/stm32f0_ARM