

Projeto final

Joaquim Ramos (88812)
Bruno Lopes (89179)

Sistemas Operativos de Tempo Real
2021/2022
MIECT
DETI

Tick updater do TMAN

Fazemos uso de uma função periódica com período igual ao argumento TMAN tick period passado na função init.

É esta tarefa que trata de garantir que as tarefas adicionadas à framework são acordadas após o seu período/offset inicial, com o auxílio de um semáforo em cada tarefa, isto para as tarefas que têm período.

Todas as tarefas estão guardadas num array, numa variável global, cujos elementos são objetos de um struct que contém as informações das tasks.

Como é que sabemos que estamos no tick correto para acordar uma tarefa?

No geral:

$$\text{resultado} = (\text{current_tick} - \text{phase}) \% \text{period}$$

Se o resultado fôr igual a zero estamos num release, a tarefa tem de ser acordada.

Como implementamos a precedência?

Cada tarefa tem um semáforo, sobre o qual ela sinaliza (executa give) quando acaba cada execução e onde as tarefas dependentes ficam à espera após esperarem o seu período/offset inicial.

Exemplo de função?

```
void pvTaskA(void *pvParam)
{
    uint8_t mesg[80];
    volatile uint32_t counter1, counter2;
    float x=100.1;
    TickType_t ticker;
    int number;

    TMan_TaskWaitPeriod(id_TaskA);

    for(;;) {

        ticker = xTaskGetTickCount();
        number = TMan_TaskStats(id_TaskA);

        sprintf(mesg,"A, %d, %d\n\r", (int) ticker, number);
        PrintStr(mesg);

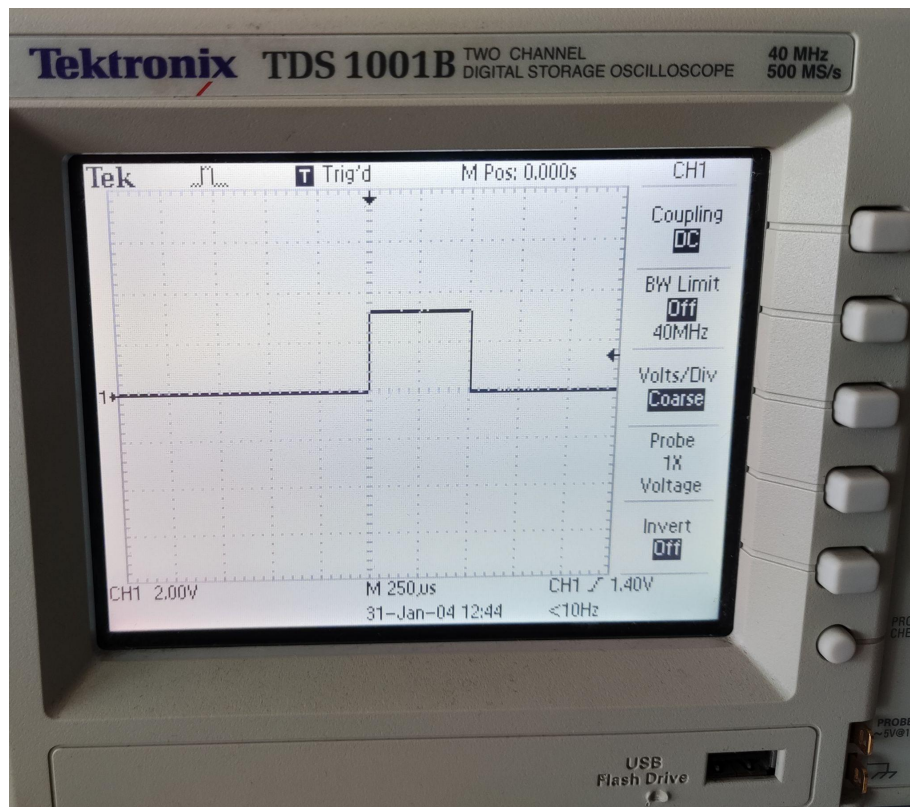
        int busy = 4;

        for(int i = 0; i < 100; i++)
            for(int j = 0; j < 30; j++)
                busy = busy + 2;

        TMan_TaskWaitPeriod(id_TaskA);

    }
}
```

Confirmação da duração da secção de busy wait



Questões?