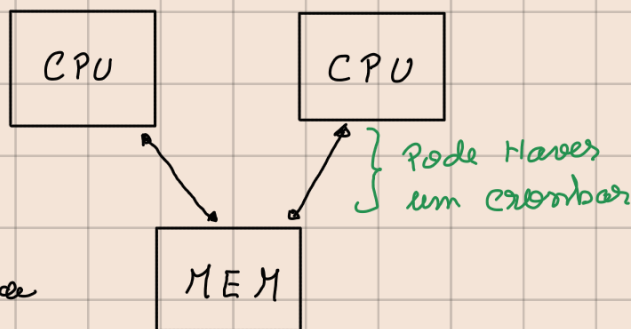


## ↳ FreeRTOS

- Ver a documentação online

### • Symmetric MultiProcessing:

- O FreeRTOS box é single-core, mas existem versões multi-core
- NUMA
- Cada vez menos há micro-controladores que corram em Baremetal
- Vantagem RTOS: Abstração das funções da arquitetura do microcontrolador. Simplifica e balanceia o controle de tarefas. Decomposição do programa em tarefas, com primitivos de sincronização já embutidos (menos código espalhado). Stacks de comunicação já incluídos.
- Uma tarefa em RTOS tem estruturas de dados que precisam de ser guardadas. "xTaskCreate" faz o armazenamento dessa estrutura fora o RTOS, que o pode colocar na Heap, enquanto que "xTaskCreateStatic", define essa estrutura no local estático, passando a quem o "fora" esse controle → "TaskControlBlock" ou "StaticControlBlock"
- Usando uma função como "xTimerCreate", cria timers de software, mas requer que haja timers de hardware. O ESP tem um timer interno ao processador. Os Periféricos são timers (GPTimers). Ao nível do SO, podemos configurar timers que ao final faz uma ação. Temos 3 tipos de timers: system timer, GPTimers e timers dedicados.
- O vTaskDelay não é um timer, mas tem um campo no TaskBlock, que é decrementado a cada tick do SO. Durante esse tempo, a tarefa fica "suspendida" e no final a "Ready" outra vez.
- Necessidade de temporizações que não alinhem com o tempo do tick (10 ms), ou mudamos o tempo do tick, ou criamos um timer dedicado, mas ajustando sempre a tick rate do SO (ou levando em conta).
- Cuidado em imitações em stack no task, visto que está não crescendo



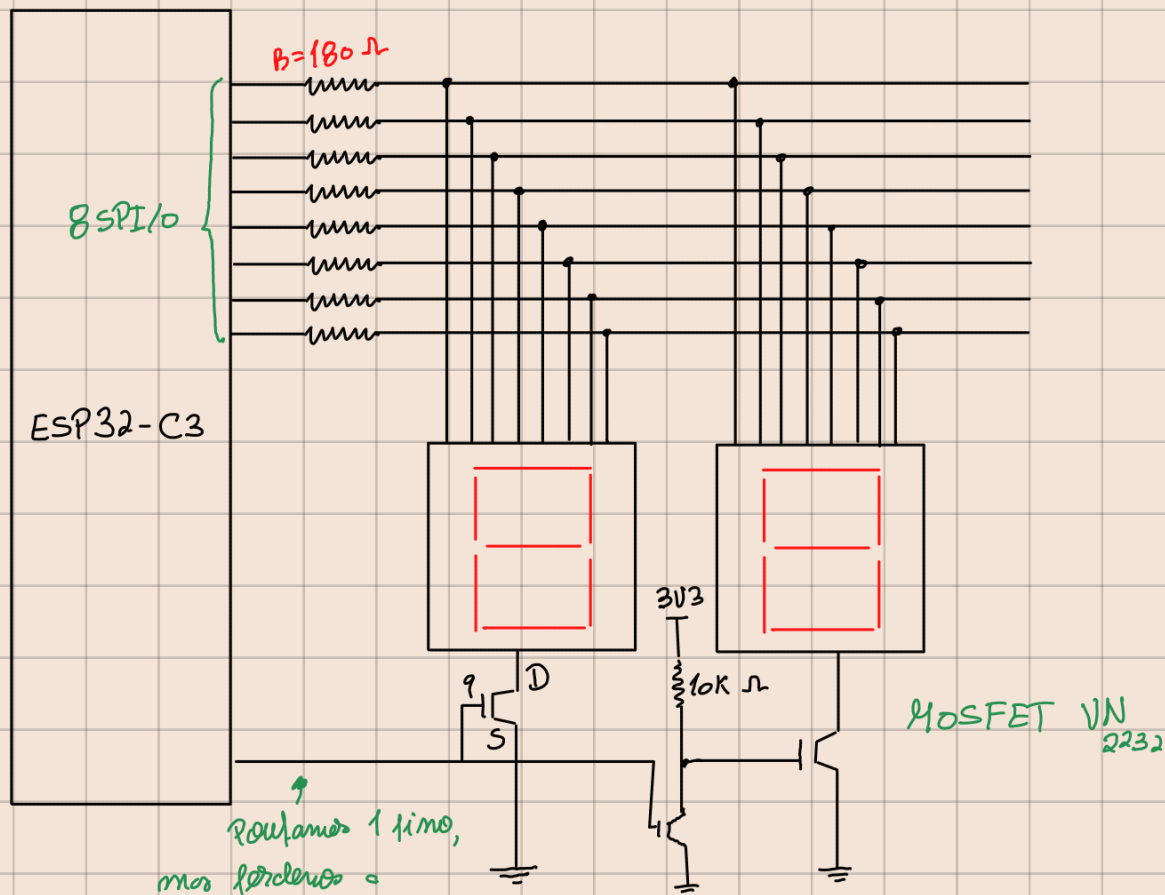
Compartando a conexão a outras funções

2 I2C - TC74

7 Display (outros pontos)

1 Gato robot

• Não usar pines:



Rotacionamos 1 pino,  
mas perdemos a  
funcionalidade de ligar  
os dois displays, sendo  
mais 1 robot