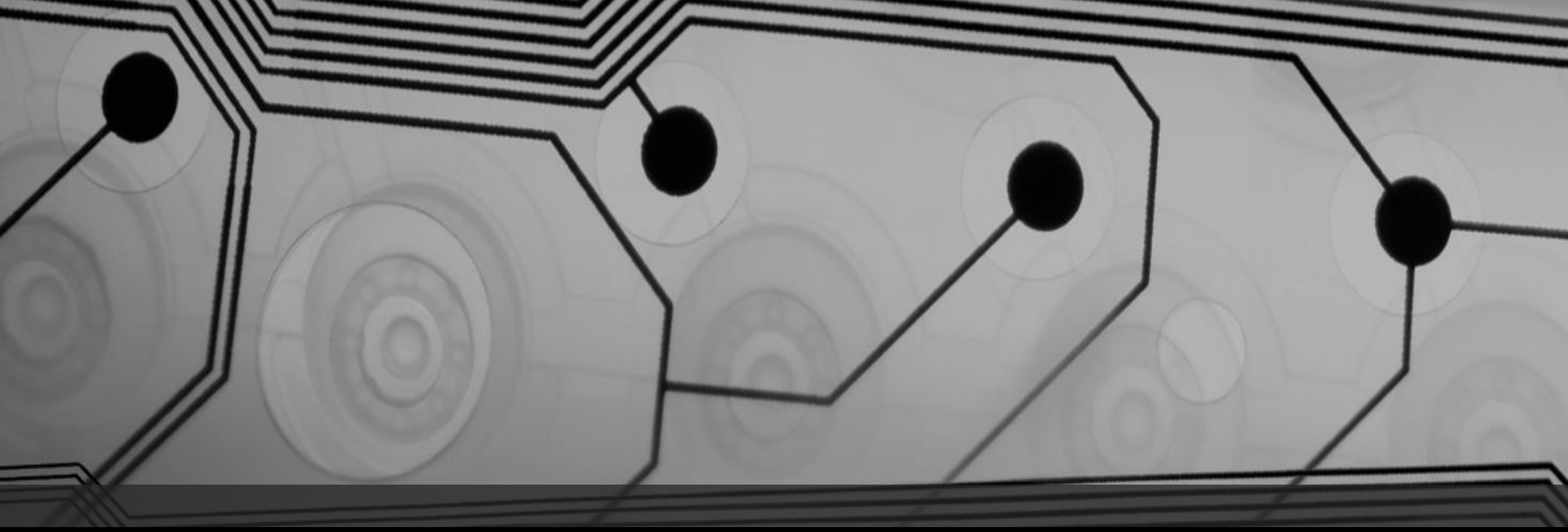
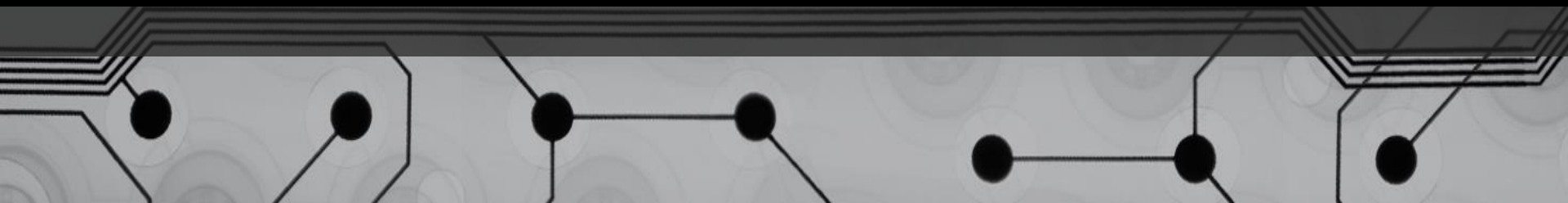


MICROCONTROLADORES

Unidade II - Introdução a ESP8266/ESP32
Aula 3

Prof. Ewerton Salvador



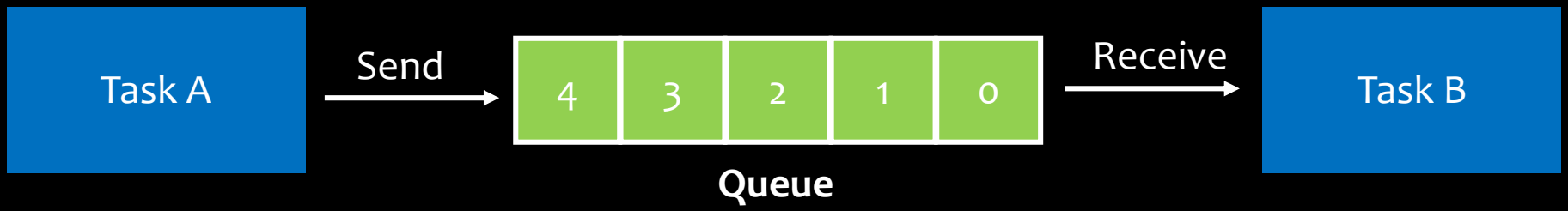
Destancado pontos importantes...

- Tasks geralmente não terminam, são loops infinitos. Se uma task precisar ser concluída, a última coisa que ela deve executar é a função vTaskDelete();
- Tasks entram em estado de bloqueio quando estão aguardando a liberação de algum recurso. Por exemplo, uma task pode ficar bloqueada esperando receber o dado de uma fila, ou esperando um MUTEX ser liberado para acessar uma região crítica;
- Programadores(as) devem ter cuidado com starvation ao usar tasks com maior nível de prioridade. Em geral, tasks de prioridade alta devem ser breves, para que a CPU seja liberada para executar tasks de menor prioridade

ESP-IDF RTOS: Queues

- Filas (*queues*) são um dos mecanismos utilizados para comunicação entre tasks
  - Outros dois mecanismos proporcionados pelo FreeRTOS são semáforos e mutexes
- É basicamente um buffer circular (FIFO)
- É multi-thread safe
  - Não provoca condições de corrida (atomicidade), diferentemente de uma simples variável global utilizada para comunicação entre tasks
- Filas podem armazenar qualquer tipo de dado
- Pode “acordar” (desbloquear) tasks que estejam esperando a chegada de algum dado

ESP-IDF RTOS: Queues



ESP-IDF RTOS: Queues

- Comportamento da fila:
  - Quando existe espaço disponível na fila, um item é adicionado imediatamente
  - Tasks recebem sempre o item mais antigo da fila, retirando o item após a cópia
  - Quando a fila está cheia, a task aguarda um período de tempo máximo pré-determinado por liberação de espaço
  - Tasks que tentam receber de filas vazias ficarão bloqueadas por um período de tempo máximo pré-determinado. Também pode ser utilizado tempo infinito

ESP-IDF RTOS: Queues

- Declarando uma fila como variável global
  - static QueueHandle_t [nome_fila] = NULL

- Função: xQueueCreate
- Finalidade: Aloca a memória necessária para funcionamento da fila
- Parâmetros:
  1. Número máximo de elementos na fila
  2. Número de bytes de cada elemento da fila
- Retorno: handle da fila, caso a mesma seja criada corretamente. 0 se a fila não foi criada corretamente

ESP-IDF RTOS: Queues

- Função: xQueueSend
- Finalidade: Escreve no final da fila. Não deve ser utilizada de dentro de uma rotina de interrupção (existe uma variante específica para isso)
- Parâmetros:
  1. Handle da fila onde a escrita deverá acontecer
  2. Endereço do elemento a ser copiado para a fila (não é passagem por referência!)
  3. Tempo de espera máximo definido em ticks. Constante portTICK_PERIOD_MS deve ser utilizada para conversão em milissegundos
- Retorno: constante pdTRUE se item for escrito com sucesso. Constante errQUEUE_FULL se fila cheia.

ESP-IDF RTOS: Queues

- Função: xQueueSendFromISR
- Finalidade: Escreve no final da fila. Versão segura para utilização em rotinas de serviço de interrupção (ISR)
- Parâmetros:
  1. Handle da fila onde a escrita deverá acontecer
  2. Endereço do elemento a ser copiado para a fila (não é passagem por referência!)
  3. Endereço de variável para receber parâmetro de saída. Recebe pdTRUE se o envio para a fila resultou em uma task sendo desbloqueada, com a task tendo prioridade maior do que a task atual. Nesse caso, antes da interrupção concluir, uma troca de contexto deve ser solicitada (ex.: portYIELD_FROM_ISR();)
- Retorno: constante pdTRUE se item for escrito com sucesso. Constante errQUEUE_FULL se fila cheia.

ESP-IDF RTOS: Queues

- Função: xQueueReceive
- Finalidade: Recebe do início da fila. Não deve ser utilizada de dentro de uma rotina de interrupção (existe uma variante específica para isso)
- Parâmetros:
  1. Handle da fila onde a leitura deverá acontecer
  2. Endereço do elemento a ser copiado da fila
  3. Tempo de espera máximo definido em ticks. Constante portTICK_PERIOD_MS deve ser utilizada para conversão em milissegundos. Constante portMAX_DELAY pode ser usada para espera indefinida.
- Retorno: constante pdTRUE se item for recebido com sucesso. pdFALSE se recepção mal-sucedida.

ESP-IDF RTOS: Queues

- Função: xQueueReceiveFromISR
- Finalidade: Recebe do início da fila. Versão segura para utilização em rotinas de serviço de interrupção (ISR)
- Parâmetros:
  1. Handle da fila onde a leitura deverá acontecer
  2. Endereço do elemento a ser copiado da fila
  3. Endereço de variável para receber parâmetro de saída. Recebe pdTRUE se o recebimento da fila resultou em uma task sendo desbloqueada (aguardando espaço na fila), com a task tendo prioridade maior do que a task atual. Nesse caso, antes da interrupção concluir, uma troca de contexto deve ser solicitada (ex.: portYIELD_FROM_ISR();)
- Retorno: constante pdTRUE se item for escrito com sucesso. Constante errQUEUE_FULL se fila cheia.

ESP-IDF RTOS: Queues

```c
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"

//Queue handle
static QueueHandle_t fila = NULL;

//Task handles
static TaskHandle_t task1_handle = NULL;
static TaskHandle_t task2_handle = NULL;

//Tasks
void task1(void *parameter){
  int i=0;

  while(1){
    xQueueSend(fila,&i,0);
    i++;
    vTaskDelay(1000/portTICK_PERIOD_MS);
  }
}

void task2(void *parameter){
  int recebido;

 while(1){
    if (xQueueReceive(fila,&recebido,portMAX_DELAY)){
      printf("%d\n",recebido);
    }
  }
}


void app_main(void){
  fila = xQueueCreate(10, sizeof(int));
  xTaskCreate(task1,"Task 1",2048,NULL,1,&task1_handle);
  xTaskCreate(task2,"Task 2",2048,NULL,1,&task2_handle);
}
```

Exemplo de uso de filas no FreeRTOS