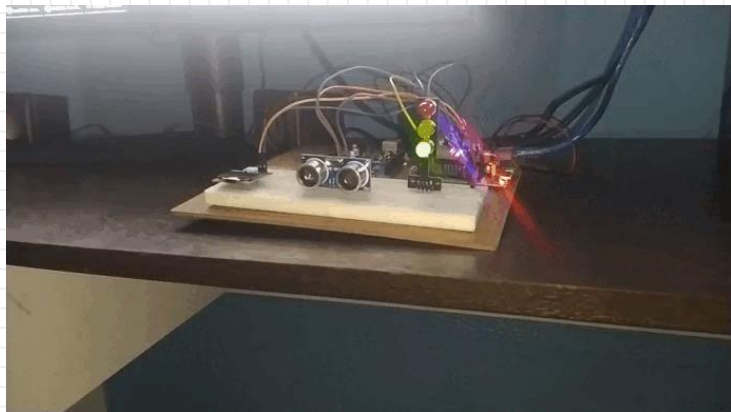


Projeto Final - Curso de Extensão

Semáforo controlado por sensor de distância

Raimundo Neto
Victor Matheus

Semáforo controlado por sensor de distância



O nosso projeto emprega um módulo LED de 8mm para simular o funcionamento de um semáforo, sendo controlado por um Sensor de Distância Ultrassônico. Além disso, integramos um display OLED para apresentar, em tempo real, a medida da distância capturada pelo sensor, expressa em centímetros (cm).



Wi-Fi



Projeto Final - Curso de Extensão



Distância: 36.79 cm

Modo Station (Cliente Wi-Fi):

- ESP32 opera como cliente em rede Wi-Fi.
- Conecta-se a uma rede usando credenciais.
- Espera até que a conexão seja estabelecida.
- Obtém um endereço IP atribuído pela rede.



Objetos FreeRTOS



Tarefas:

- "ultrasonic_task" lê o sensor ultrassônico, atualiza a distância e controla o semáforo.
- "led_task" controla o estado dos LEDs com base na distância lida.
- "monitor_task" monitora o uso de recursos, como a pilha de tarefas.

Semáforo:

- Usa ``xSemaphoreCreateBinary`` para criar o semáforo "my_semaphore".
- O semáforo é usado para coordenar a leitura do sensor e a atualização dos LEDs.
- "ultrasonic_task" libera o semáforo após cada leitura, permitindo que "led_task" acesse os dados.
- Ajuda a evitar conflitos de acesso aos recursos compartilhados.

Filas:

- Usa a função ``xQueueCreate`` para criar uma fila chamada "distance_queue".
- A fila é usada para armazenar valores de distância lidos pelo sensor.
- "ultrasonic_task" envia a distância para a fila, e "led_task" a recebe.



Monitoramento de Heap e Stack



Heap:

O heap é a região de memória onde são alocados dinamicamente objetos e variáveis durante a execução do programa. É importante gerenciar adequadamente para evitar vazamentos de memória e falta de memória.

O monitoramento do heap é realizado pela função `monitor_task`.

```
void monitor_task(void *pvParameters) {  
    while (1) {  
        vTaskDelay(pdMS_TO_TICKS(5000));  
  
        size_t free_heap = esp_get_free_heap_size();  
  
        Serial.print("Heap Livre: ");  
        Serial.print(free_heap);  
        Serial.println(" bytes");  
  
        vTaskDelay(pdMS_TO_TICKS(1000));  
    }  
}
```

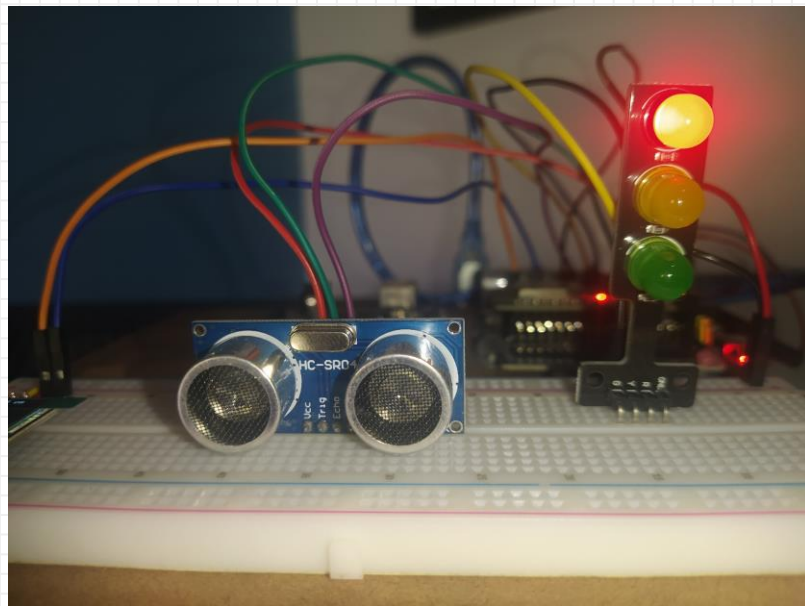
Stack:

Cada tarefa possui sua própria pilha (stack) para armazenar suas variáveis locais e informações de contexto. Monitorar o uso da pilha é importante para evitar estouro de pilha (stack overflow) e falhas no sistema. O monitoramento das pilhas das tarefas utiliza 'ultrasonic_task' e 'led_task'

```
...  
if (stack_used > ultrasonic_stack_highwater) {  
    ultrasonic_stack_highwater = stack_used;  
}  
  
...  
if (stack_used > led_stack_highwater) { led_stack_highwater =  
    stack_used; }
```

```
10:15:33.921 -> Heap Livre: 154516 bytes  
10:15:33.921 -> Stack Ultrasonic: 3108 bytes  
10:15:33.921 -> Stack LED: 3572 bytes
```

Serial monitor



MUITO OBRIGADO!!!!