Relatório de Projeto: Broker MQTT com Mosquitto e Raspberry Pi Pico W

# 1. Introdução

Este trabalho tem como objetivo implementar um sistema de comunicação utilizando o protocolo MQTT, com o broker Mosquitto rodando em um servidor Linux, e um cliente embarcado baseado na Raspberry Pi Pico W. Foram seguidos dois tutoriais: 'Instalando Mosquitto' e 'Adicionando Criptografia Tipo PSK no Mosquitto'. O projeto integra sensores e botões na Pico W, que enviam dados para o broker via rede Wi-Fi.

# 2. Instalação do Broker Mosquitto

Inicialmente foi realizada a instalação do Mosquitto no servidor Linux. Foram abertas as portas 1883 e 8883 no firewall, e em seguida o serviço foi instalado com o comando:  
  
 sudo apt install mosquitto mosquitto-clients -y  
  
Após a instalação, o serviço foi verificado com:  
  
 sudo systemctl status mosquitto | grep Active  
  
Em seguida, foram criados usuários e senhas para acesso seguro, utilizando o arquivo passwd do Mosquitto, e configurado o arquivo /etc/mosquitto/conf.d/default.conf para exigir autenticação.

# 3. Adicionando Criptografia PSK

Seguindo o tutorial de criptografia PSK, foi configurada uma chave pré-compartilhada (Pre-Shared Key) para aumentar a segurança das conexões. A chave foi convertida para hexadecimal e armazenada em /etc/mosquitto/pskfile. No arquivo de configuração, foi definida a porta 8883, e adicionadas as linhas psk\_file e psk\_hint.  
  
Para fins de teste com a Raspberry Pi Pico W, a configuração também foi ajustada para operar sem PSK, utilizando apenas autenticação por usuário e senha na porta 1883.

# 4. Implementação na Raspberry Pi Pico W

No lado do cliente, foi implementado um firmware em C utilizando o Pico SDK, FreeRTOS e a pilha lwIP. O código conecta a Pico W a uma rede Wi-Fi local e, em seguida, estabelece comunicação MQTT com o broker. Foram implementadas duas tarefas principais:  
  
 - vButtonTask: responsável por ler o estado de um botão físico e enviar a informação via MQTT.  
 - vTemperatureTask: responsável por ler a temperatura interna do chip RP2040 (via ADC) e publicar periodicamente no broker.  
  
Cada publicação acende brevemente um LED na placa para indicar sucesso no envio. O código faz uso da biblioteca lwIP MQTT.

# 5. Problemas Encontrados e Soluções

Durante os testes, foi identificado o erro:  
  
 \*\*\* PANIC \*\*\* pcb->snd\_queuelen >= pbuf\_clen(next->p)  
  
Este erro ocorre devido ao esgotamento do buffer de envio da pilha lwIP, causado por múltiplas tarefas publicando simultaneamente no cliente MQTT e pelo uso de sleep\_ms em vez de vTaskDelay. As soluções propostas foram:  
  
 - Substituir sleep\_ms por vTaskDelay em todas as tasks.  
 - Centralizar as publicações MQTT em uma única task, utilizando filas (xQueue) ou mutexes para sincronização.  
 - Aumentar os intervalos de publicação para evitar congestionamento da fila de pacotes.

# 6. Conclusão

O projeto demonstrou a viabilidade de integrar a Raspberry Pi Pico W a um broker Mosquitto configurado em um servidor Linux, permitindo comunicação segura via MQTT. Através de ajustes na configuração do Mosquitto e no firmware da Pico W, foi possível implementar a publicação periódica de dados de sensores e estados de botões. Apesar de desafios com buffers da pilha lwIP, foram propostas soluções para tornar o sistema estável e confiável.