Embarcatech - Unidade 04

Aluna: Marina Medeiros de Araújo Leite (Matrícula: 202521511920065)

Configuração do Mosquitto MQTT com autenticação PSK e BitDogLab.

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.

Adicionando Criptografia PSK

Em seguida, 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 /etc/mosquitto/pskfile*" e "*psk_hint hint*"

Após essas configurações, foram enviadas mensagens de teste para o broker usando dois terminais diferentes. O primeiro terminal era o destinatário, e foi utilizado o comando *mosquitto_sub*. Já o segundo era o remetente, e nele foi utilizado o comando *mosquitto_pub*.

```
(base) marina@marinaLinux:~$ mosquitto_sub -h localhost -p 8883 -u aluno65 -P aluno65 -t /aluno65/pub --p
sk-identity aluno65 --psk ABCD65EF1234
mensagem
Embarcatech
```

```
(base) marina@marinaLinux:~$ mosquitto_pub -h localhost -p 8883 -u aluno65 -P aluno65 -t /aluno65/pub --p
sk-identity aluno65 --psk ABCD65EF1234 -m "mensagem"
(base) marina@marinaLinux:~$ mosquitto_pub -h localhost -p 8883 -u aluno65 -P aluno65 -t /aluno65/pub --p
sk-identity aluno65 --psk ABCD65EF1234 -m "Embarcatech"
```

Implementação na BitDogLab

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 da BitDogLab e publicar

periodicamente no broker.

Cada publicação acende brevemente um LED na placa para indicar sucesso no envio. O código faz uso das bibliotecas *lwipipts*, *mqtt_conn* e *wifi_conn*.

A biblioteca *mqtt_conn* não oferece suporte para publicações utilizando PSK. Por isso, foi necessário fazer as publicações utilizando apenas o usuário e senha.

```
^C(base) marina@marinaLinux:~$ mosquitto_sub -h localhost -p 1883 -u aluno65 -P aluno65 -t /aluno65/bitdoglab/sensor
Embarcatech - tarefa 04
Temperatura: 33.22

Botão: 1

Botão: 1

Temperatura: 35.56

Botão: 0

Temperatura: 35.56
```

Mensagens do terminal do VS Code:

```
---- Opened the serial port /dev/ttyACM0 ----
Conectado ao Wi-Fi
Wi-Fi conectado!
Temperatura: 33.22
MQTT enviado!
Mensagem: Botão: 1

MQTT enviado!
Conectado ao broker MQTT com sucesso!
Publicação MQTT enviada com sucesso!
Publicação MQTT enviada com sucesso!
Mensagem: Botão: 1

MQTT enviado!
Publicação MQTT enviada com sucesso!
Temperatura: 35.56
MQTT enviado!
Publicação MQTT enviada com sucesso!
Mensagem: Botão: 0

MQTT enviado!
Temperatura: 35.56
MQTT enviado!
Publicação MQTT enviada com sucesso!
Publicação MQTT enviada com sucesso!
Mensagem: Botão: 1
```

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. Porém, devido a limitações da biblioteca, não foi possível enviar os dados utilizando PSK.