

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**LABORATÓRIO DE REDE DE SENSORES SEM FIO II**  
**ATIVIDADE DE LABORATÓRIO**

**Professor:** Guilherme Luiz Moritz e Ohara Kerusauskas Rayel

**Tema:** Atividade - Utilização do protocolo MQTT

**Data:** 12 de abril de 2022

## Atividade - MQTT

### 1 Atividade

Nesta atividade, utilizaremos o protocolo MQTT para comunicar com um servidor externo. Observe que o servidor mais utilizado para MQTT (Mosquitto) não possui implementação do protocolo MQTT-SN. Como agravante, a implementação padrão do MQTT do Contiki (disponível em `contiki/apps`) também segue a tendência, não disponibilizando o protocolo MQTT-SN. Desta maneira, serão utilizadas implementações alternativas disponíveis livremente na internet.

1. Realize um pull do upstream do repositório da disciplina, conforme material da Aula
2. Observe que este processo pode gerar conflitos no código. É recomendado que se realize backup da pasta de trabalho antes da realização do pull.
2. Altere o endereço de conexão para o endereço do *broker* MQTT-SN disponibilizado: `labscpi.eletrica.eng.br`;
3. Verifique o código que faz o cliente se inscrever em um tópico chamado `xxxxxxxxxxxxxxxxx/ctrl` onde `xxxxxxxxxxxxxxxxx` é o *time address* do kit (processo `ctrl_subscription_process`);
4. Verifique o código que faz o cliente publicar uma mensagem periodicamente em um tópico chamado `xxxxxxxxxxxxxxxxx/msg` (processo `publish_process`);
5. Execute o exemplo fornecido e anote o *time address* do nó; Desligue o nó após anotar o endereço;
6. Utilize a ferramenta de linha de comando `mosquitto_sub` para registrar-se no tópico `xxxxxxxxxxxxxxxxx/ctrl` (Dica: Utilize o comando `mosquitto_sub -h` para encontrar as opções de configuração do endereço e do tópico. Não se esqueça também de utilizar aspas simples no nome dos tópicos). Outra dica interessante é que existem muitos outros clientes de MQTT que podem ser utilizados, como por exemplo a Extensão VsMqtt disponível para Visual Studio Code (<https://marketplace.visualstudio.com/items?itemName=rpdswtk.vsmqtt>);
7. Utilize a ferramenta de linha de comando `mosquitto_pub` para publicar no tópico `xxxxxxxxxxxxxxxxx/ctrl` e verifique se a mensagem foi recebida corretamente através da ferramenta `mosquitto_sub`;
8. Encerre a execução da ferramenta `mosquitto_sub`. Ligue o nó e verifique se o processo de inscrição no tópico `xxxxxxxxxxxxxxxxx/ctrl` correu corretamente;
9. Repita o item 7 e verifique se o nó recebeu a mensagem corretamente;
10. Utilize a ferramenta de linha de comando `mosquitto_sub` para se inscrever no tópico `xxxxxxxxxxxxxxxxx/msg`. Verifique se a ferramenta recebe periodicamente uma mensagem do nó;

11. Verifique o funcionamento dos *wildcards* do MQTT, registrando-se no tópico `+/msg` e observando as mensagens recebidas de todos os nós da sala que atualmente estão conectados no *broker* MQTT-SN;
12. Altere a função `publish_receiver` para quando receber uma publicação em `xxxxxxxxxxxxxxxx/ctrl` ligar os leds caso o número recebido for maior que zero. Dica: Utilize a função `atoi` (disponível em `stdlib.h`) para converter a string recebida para um valor inteiro;
13. Testar o funcionamento da implementação utilizando a ferramenta `mosquitto_pub`.
14. E se fosse necessário receber mensagens em diferentes tópicos? E enviar em diferentes tópicos?

## 2 Atividade Extra

1. Utilize o software IoT MQTT Dashboard [https://play.google.com/store/apps/details?id=net.routix.mqttdash&hl=pt\\_BR&gl=US](https://play.google.com/store/apps/details?id=net.routix.mqttdash&hl=pt_BR&gl=US) para comandar seu projeto.