

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PPGSESAI03 - REDES DE SENSORES SEM FIO DE LONGO ALCANCE
ATIVIDADE DE LABORATÓRIO

Professor: Guilherme Luiz Moritz e Ohara Kerusauskas Rayel

Tema: Enviando e recebendo mensagens via MQTT

Data: 19 de abril de 2022

Introdução

Na prática passada utilizamos o software mosquitto_sub e mosquitto_pub para enviar e receber mensagens MQTT para nosso nó conectado no broker da disciplina. Nesta prática incrementaremos o exemplo a fim de utilizar Python para consumir e enviar estas mensagens e desta maneira criar uma aplicação em nuvem.

Atividade 1 - Enviando comandos MQTT utilizando Python

1. Para programar Python no ambiente de desenvolvimento da disciplina utilizaremos o Visual Studio Code. Por outro lado, a versão do Pendrive está desatualizada. Atualize o Visual Studio Code acessando https://code.visualstudio.com/docs/?dv=linux64_deb.
2. Será feito o download de um arquivo .deb. Utilizando um Terminal do Linux, navegue até a pasta onde o arquivo está salvo e digite o comando `sudo dpkg -i code_1.66.2-1649664567_amd64.deb` (ou qualquer outro nome do .deb, caso a versão seja diferente da deste exemplo)
3. Instale a biblioteca MQTT do Python com o comando `python3 -m pip install paho-mqtt`
4. Agora faça download do stub de conexão MQTT via Python, disponível no moodle, para uma pasta, de preferência vazia.
5. Abra a pasta criada no Item 4 utilizando o comando Open Folder do Visual Studio Code e acesse o arquivo recém adquirido;
6. Instale a Extensão Python utilizando o menu Extensions do Visual Studio Code (tecla de atalho Ctrl+Shift+X)
7. Agora inicie a edição do arquivo.
Observe que, na linha 11 há o comando `client.subscribe("labscim/example")` que serve para se inscrever em um tópico, chamado, neste exemplo de `labscim/example`;
8. Altere este tópico para o mesmo tópico do nó, já previamente utilizado na prática anterior.
9. Observe que, na linha 19 há o comando `client.connect('localhost', 1883, 60)` que serve para nos conectarmos no servidor MQTT do gateway.
10. Altere a linha para utilizar o endereço do nosso broker MQTT, já previamente utilizado na prática anterior (`labscpi.eletrica.eng.br`).
11. Execute o script e inicie a simulação. Observe que, a cada mensagem MQTT recebida, o callback `on_message()` é executado. No momento, tudo que ele faz é imprimir as mensagens recebidas no terminal;
12. Faça um pull do upstream do repositório da disciplina. Observe que um novo código de exemplo, chamado mqtt-example.c foi adicionado. Este código é uma versão aperfeiçoada do código utilizado na aula passada, onde é mais fácil adicionar novos tópicos;

13. Por este motivo, modifique o código do nó para que ele se registre em dois tópicos de controle, um chamado '0000000000000000/control/red' e outro chamado '0000000000000000/control/green'. Sempre que um valor 1 for publicado nestes tópicos, o led correspondente do kit deve acender. Caso um valor diferente de 1 seja enviado, o led correspondente deve apagar. Para realizar este comportamento:
- (a) Observe a linha 67 do código, e declare duas novas strings com os tópicos a serem criados;
 - (b) Observe a linha 449, onde há dois `memcpy` que preenchem os tópicos com o ID do nó. Altere para refletir os tópicos recém criados;
 - (c) Altere as variáveis `static char* topics[]` e `static char* subscribe_topics[]` para que elas contenham as variáveis declaradas no item 13a;
 - (d) Altere o `enum` da linha 77 para que ela contenha dois itens que representem as posições das variáveis declaradas em 13c;
 - (e) Finalmente, complete o switch da função `publish_receiver` (linha 237) para que ele trate os dois itens recém adicionados ao `enum`, no Item 13d. É neste switch que os leds devem ser acionados.
14. Modifique a mensagem que está sendo publicada periodicamente no tópico '0000000000000000/uptime' para que ela envie um número aleatório entre 0 e 99. Dica: O código abaixo cria uma string que contém um número aleatório com as especificações desejadas.

```
char buf[32];
uint32_t buf_len;
int32_t r = rand() % 100;
sprintf(buf, "%d", r);
buf_len = strlen(buf);
```

15. Ao receber a mensagem via Python, interprete o número e comande os leds utilizando o comportamento programado no Item 13, para que:
- (a) Caso o número enviado for maior que 70, acione o led vermelho e apague o led verde;
 - (b) Caso o número enviado estiver entre 30 e 70, acione o led verde e apague o led vermelho;
 - (c) Caso o número enviado for menor que 30, apague ambos os leds;

Dica → O seguinte código python interpreta um array de bytes contido na variável `msg.payload` como um int:

```
value = int(msg.payload.decode("utf-8"))
```

Dica 2 → O seguinte código python pode ser adicionado ao callback de recebimento via MQTT para que o mesmo envie uma resposta:

```
tg = '{:s}/control/green'.format(topics[0])
client.publish(tg, "1")
```