05

Publicando e recebendo mensagens

Transcrição

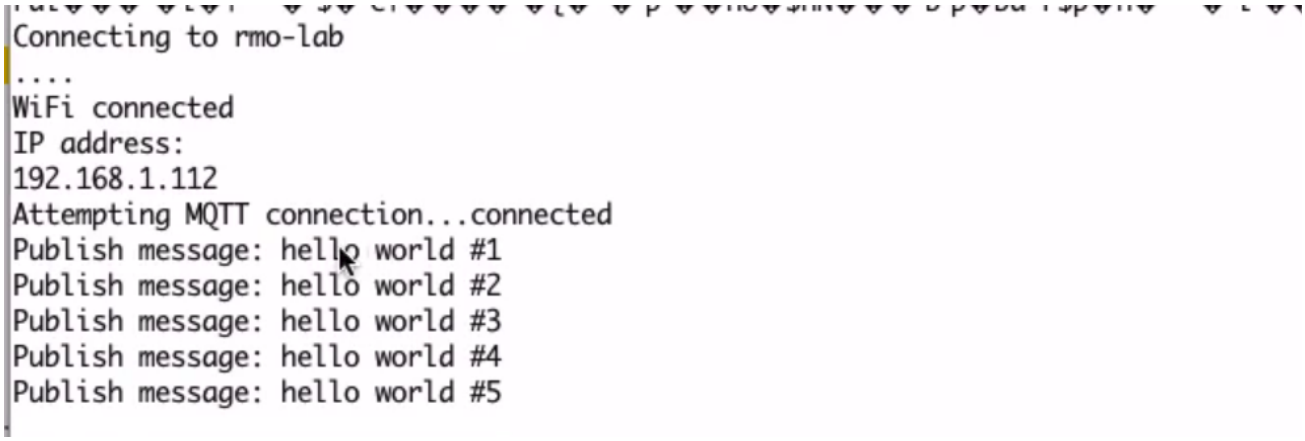
Precisamos buscar em outros lugares do código da biblioteca onde os dados de porta podem ser utilizados. Mais diretamente, o de usuário e senha podem ser encontrados no bloco `reconnect`. Como vemos abaixo:

```
if (client.connect("ESP8266Client")) {
    // restante de código
}
```

Neste ponto, precisamos também informar os dados de usuário e senha que criamos antes.

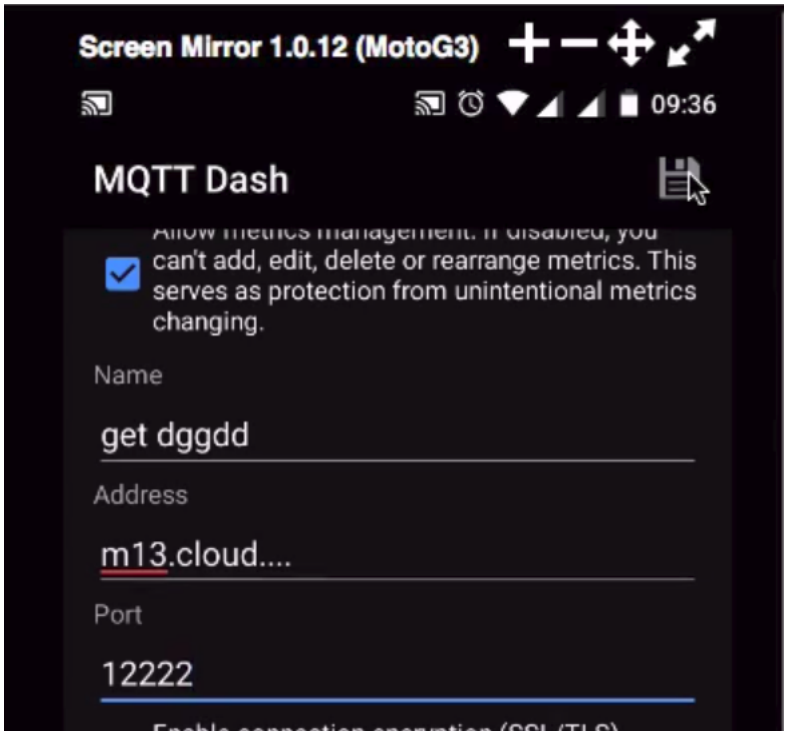
```
if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
    // restante de código
}
```

Estas informações, além de outras sobre o uso da biblioteca *PubSubClient*, podem ser encontradas neste endereço: PubSubClient API Documentation (https://pubsubclient.knolleary.net/api.html). Depois, faremos *upload* do código para a placa para efetivamente testar. Podemos visualizar as mensagens do processo no *console* da serial.

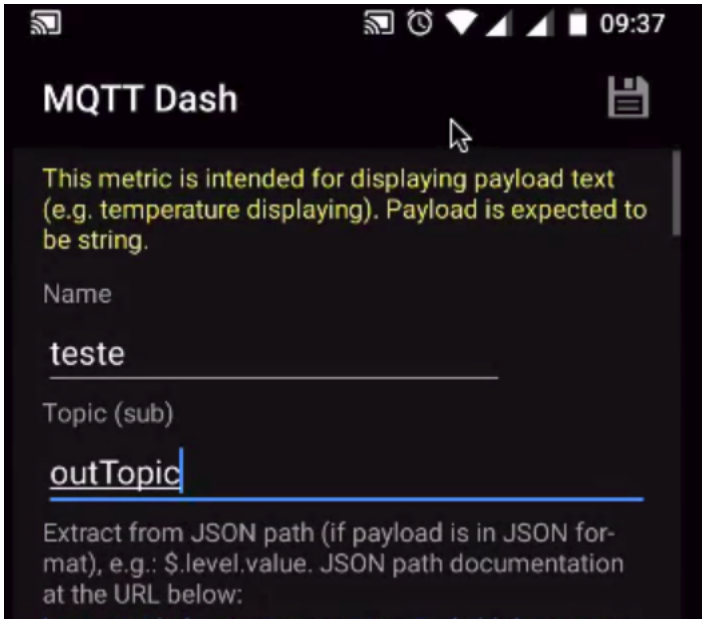


Com o código em execução, usaremos uma aplicação cliente para fazer a leitura das mensagens. Note que no código há uma linha informando qual o tópico que mantém as mensagens, que é o `outTopic`. Para Android, uma das alternativas como cliente é o aplicativo MQTT Dash (https://play.google.com/store/apps/details?id=net.routix.mqttdash&hl=en)

Nele, podemos configurar o serviço (*broker*), com endereço do servidor, nome de usuário e senha da mesma forma que fizemos no código.



Após configurarmos o *broker*, assinaremos um tópico de recebimento de mensagens como ilustra a imagem abaixo:



E por último, visualizaremos as mensagens chegando de forma semelhante ao que vemos na serial.