Assignment: MQTT with button and LED

Goals

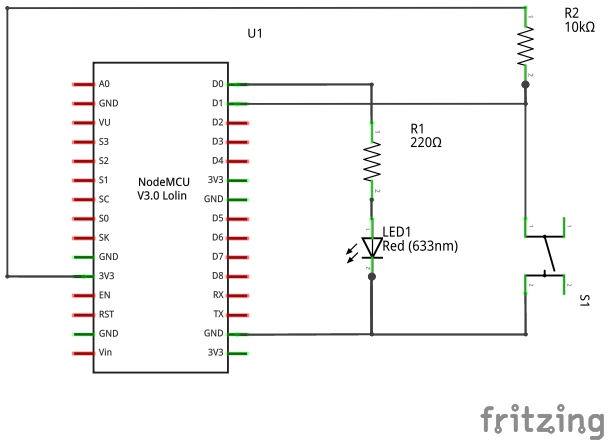
- Get aquainted with MQTT.
- Publish a message when a button is pressed.
- Subscribe to a topic to control the LED

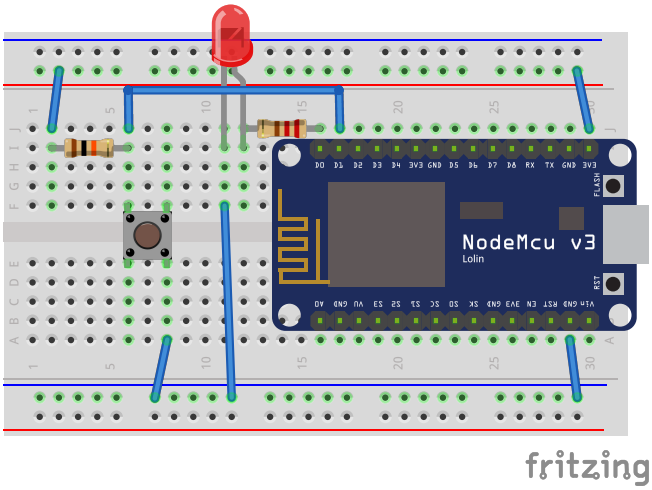
Preparation

- Install the *PubSubClient* library. Use the library manager under *Sketch -> Include library -> Library Manager* to install it.
- Open http://www.hivemq.com/demos/websocket-client/ and subscribe to `ndc/#` or `ndc/$deviceId/#`.

Step 1

Wire up this schematic on the bread board:





Step 2

- Read button's value inside `loop()`. If the button's state changes, print a message.

Note: reading the button in a busy loop is not really a best practice as it uses lots of energy. Instead use the `attachInterrupt`.

Step 3

- Connect to the Wi-Fi network
- Connect to MQTT broker. Use `broker.hivemq.com:1883` as host and port.

See the slides for example code.

Step 4

- Create a subscription for `ndc/$device-id/#` with HiveMQ's web ui: http://www.hivemq.com/demos/websocket-client/
- Publish a message when the button is pressed on the topic `ndc/$device-id/button`

Step 5 (Bonus)

Feel free to be creative here.

The other useful half of MQTT is subscribing to topics and reacting to messages. Here are some example things you can do:

- Subscribe to the topic you're publishing to and change the state of the LED when the button is pressed. This will give you a very complicated and brittle way of toggling a LED.
- In `loop` blink the led. Subscribe to another topic (for example `ndc/$device-id/frequency`), and use the value as a way to control the blink rate.

Step 6 (Strech goal)

Use a last will message to indicate if the device is online or not.

- When connecting, include a last will message with topic `ndc/$device-id/online` and payload `0`.
- After a successfull connection, publish a similar message with the payload `1`. Observe what happens when you unplug the device.