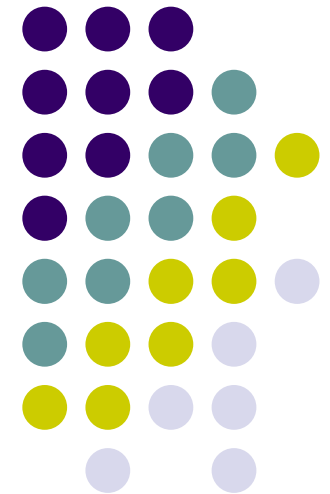


SCM

Sistemas de Comunicação Móvel

3rd Graded Assignment



© Paulo Simões – DEI/FCTUC

Goals



To further explore NodeMCU's WiFi libraries - Promiscuous Mode
To start exploring MQTT

Delivery date:

- Phase 1: October 24, 2021
(report sent until the end of the day,
live demo on October 24 or the next week)
- Phase 2: November 5, 2021
(report sent until the end of the day, live demo in Nov. 7)

Documentation – WiFi library:

- <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>
- Other documentation sources to be discovered by the students

Stealth Presence Tracker

Phase 1: find out when Mr. Bond is @dei



Our personal devices with Wi-Fi often “announce” our presence by having visible network activity using their unique MAC addresses, even when we are not actively using the internet.

This can be used for stealth tracking of our presence in a location

- Using your NodeMCU, build a device that permanently scans all Wi-Fi channels, listening for packets sent by other devices, in order to detect if a specific predefined set of devices (defined by their mac addresses) are present in the area.
- When a device in this list is present, the NodeMCU led should be turned on for 2 seconds, and then the system goes back to scan mode.

Use the serial port to provide further details about the detected device, such as the specific device that triggered the led and the channel where it is present.

Assignment description

Reference Scenario for basic setup



WiFi clients

Stealth
Presence Detector



Assignment description

Recommendations for Phase 1:



Please consider the following notes:

- In order to listen for all devices (even those not associated with a specific SSID) you need to use promiscuous mode.
- Please remember there are several Wi-Fi channels. Even if in promiscuous mode you'll be listening only one channel at a time. You must jump from channel to channel to cover all possible channels.
- The Wi-Fi library you used in previous assignments might not provide explicit support for promiscuous Wi-Fi mode. Feel free to further investigate this and, if necessary, use other libraries.
- In the report, please add a discussion on the possible solutions for a mobile device to avoid being tracked in such a way. Investigate if mobile platforms such as Android or Apple iOS provide some support for this.

Assignment description

Delivery format for Phase 1:



- Short report, including:
 - *Student name*
 - *Optionally, any remarks and comments you may want to provide (e.g. implementation options, found issues, devised solutions, etc.)*
 - *Discussion on how to avoid device tracking using such techniques*
 - *Printed source code*
(please properly comment your source code so I can understand it)
 - *Report to be sent by email until the November 24, 23h59*
 - *Live demo to be provided in October 31 class*
(or the week before, if ready)

Stealth Presence Tracker

Phase 2: Integrate the trackers using MQTT



After completing the first phase, use MQTT to:

- Receive the list of devices to detect from a central application
- Report to the central application whenever a device is detected

Optionally, you may keep using the detector's serial port to provide further details about the detected device, such as the specific device that triggered the led and the channels where it is present.

Notes:

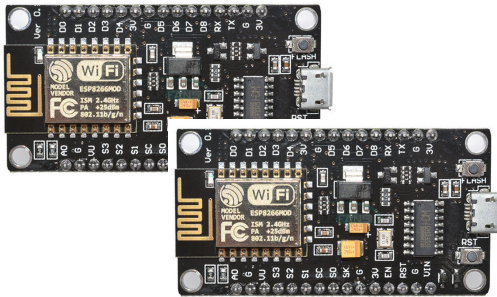
- *You may use any Wi-Fi network for MQTT communications*
(remember the detector doesn't need to be connected to the broker 100% of the time)
- *While @dei, you may use the test MQTT server*
- The “*application for controlling the system*” may be a web browser or a specific client developed in any language and for any operating system chosen by the student.
- There might be more than one detector.
Plan MQTT features considering this.

Assignment description

Reference Scenario for Phase 2



Stealth
Presence Detectors



MQTT Broker



Management
Application



WiFi clients



Assignment description

Delivery format for Phase 2 (November 5th):



- PDF Report, including:
 - *student name*
 - *Description of the developed system, explaining the general architecture, expected behavior, used libraries and opensource software, etc.*
 - *Description of the solutions devised for MQTT communications, including naming of data, data persistence options, etc.*
 - *Any further remarks and comments you may want to provide (e.g., implementation options, found issues, devised solutions, etc.)*
- Zipped source code
(please properly comment your source code so I can understand it)