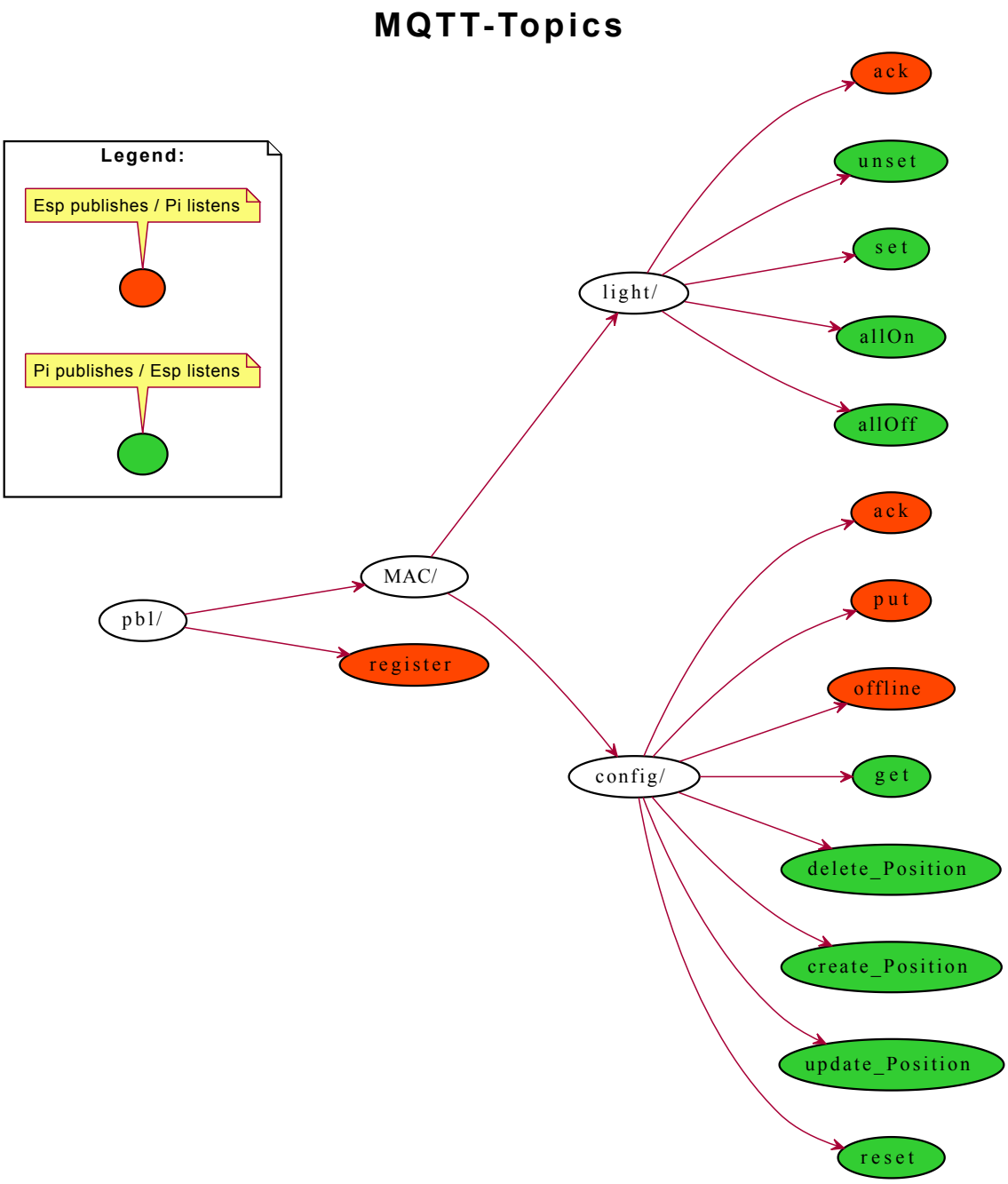
Documentation and Diagrams

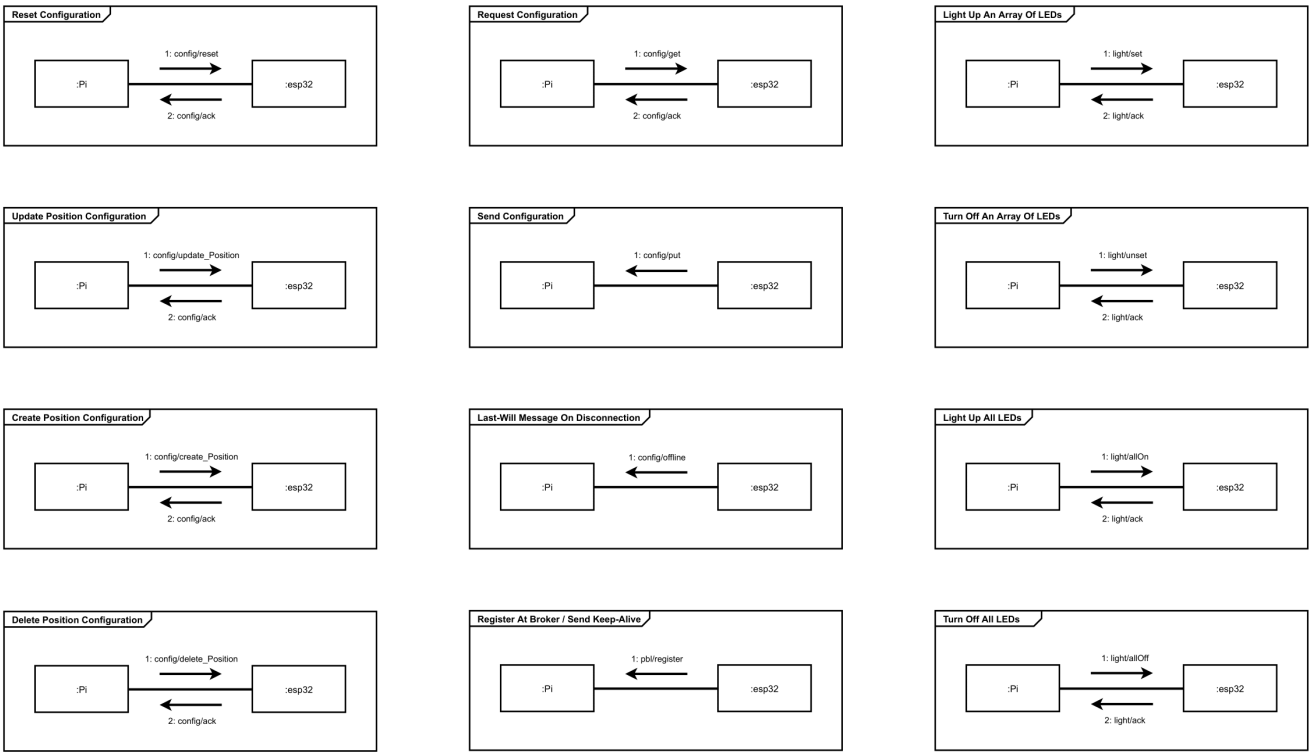


	Technische Hochschule Mittelhessen - THM University of Applied Sciences
Repository	https://git.thm.de/softwaretechnik-projekt-pick-by-light-system-wise21_22/pbl-embedded-system/wiki/
UML-Tool	PlantUML v1.2022.0
Model version	1.0
Creation date	09.12.2021
Creator	Joel Silas Zelenak and David Lotz
Mail	joel.zelenak@mni.thm.de
Last change	25.01.2022

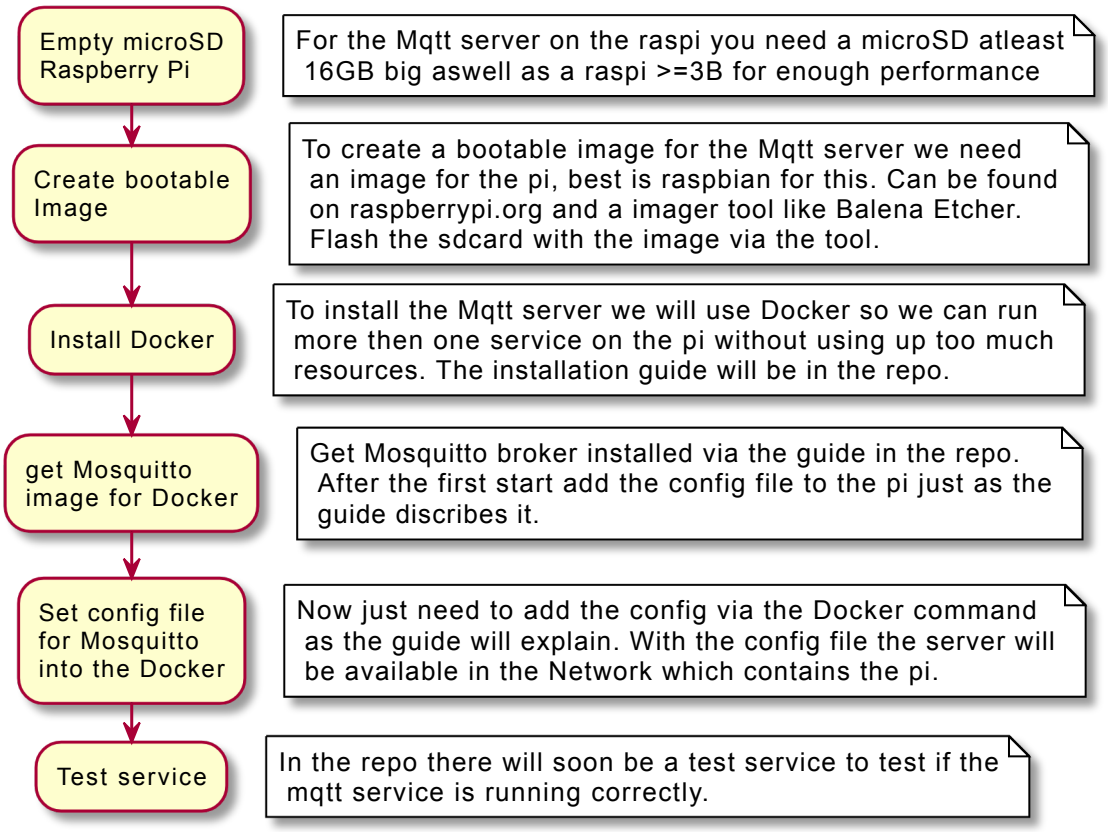
Topics	Payload	Description
.../config/reset	[Byte AckID]	Reset the LED configuration on esp32 (clear its permanent memory)
.../config/update_Position	[Byte AckID, Byte Position, Byte LED1, Byte LED2, Byte LED3, ...]	Update the LED configuration of a position
.../config/create_Position	[Byte AckID, Byte Position, Byte LED1, Byte LED2, Byte LED3, ...]	Create a LED configuration of a position
.../config/delete_Position	[Byte AckID, Byte Position]	Delete the LED configuration of a position
.../config/get	[Byte AckID]	Tell the esp32 to send its current configuration
.../config/put	[Byte PositionID, Byte LED1, Byte LED2, ...]	Send current configuration. PositionID is the ID of the position described in this message. The esp32 publishes as many messages as there are configured positions.
.../config/ack	[Byte AckID]	Acknowledge a received and executed "config" message
.../config/offline	[NULL]	LastWillTopic of every ESP32 where a null byte array is sent from the Broker when a disgraceful disconnection from an ESP32 is detected.
.../light/allOn	[Byte AckID, Byte R, Byte G, Byte B]	Turn on all LEDs
.../light/allOff	[Byte AckID]	Turn off all LEDs
.../light/set	[Byte AckID, Byte LED1, Byte LED2, ..., Byte R, Byte G, Byte B]	Turn on a specific array of LEDs
.../light/unset	[Byte AckID, Byte LED1, Byte LED2, ...,]	Turn off a specific array of LEDs
.../light/ack	[Byte AckID]	Acknowledge a received and executed "light" message
pbl/register	[11x Byte Mac]	Register a ESP32 on the Pi with its MAC address. ESP32 sends periodically his MAC address as keep-alive message.

Communication Diagram MQTT

This diagram shows every communication route between the Raspberry Pi and the Espressif ESP32
Every topic **not** starting with "pbl", starts with "pbl/MAC/" (MAC = XX:XX:XX:XX:XX:XX, with X = Hexadecimal Number)

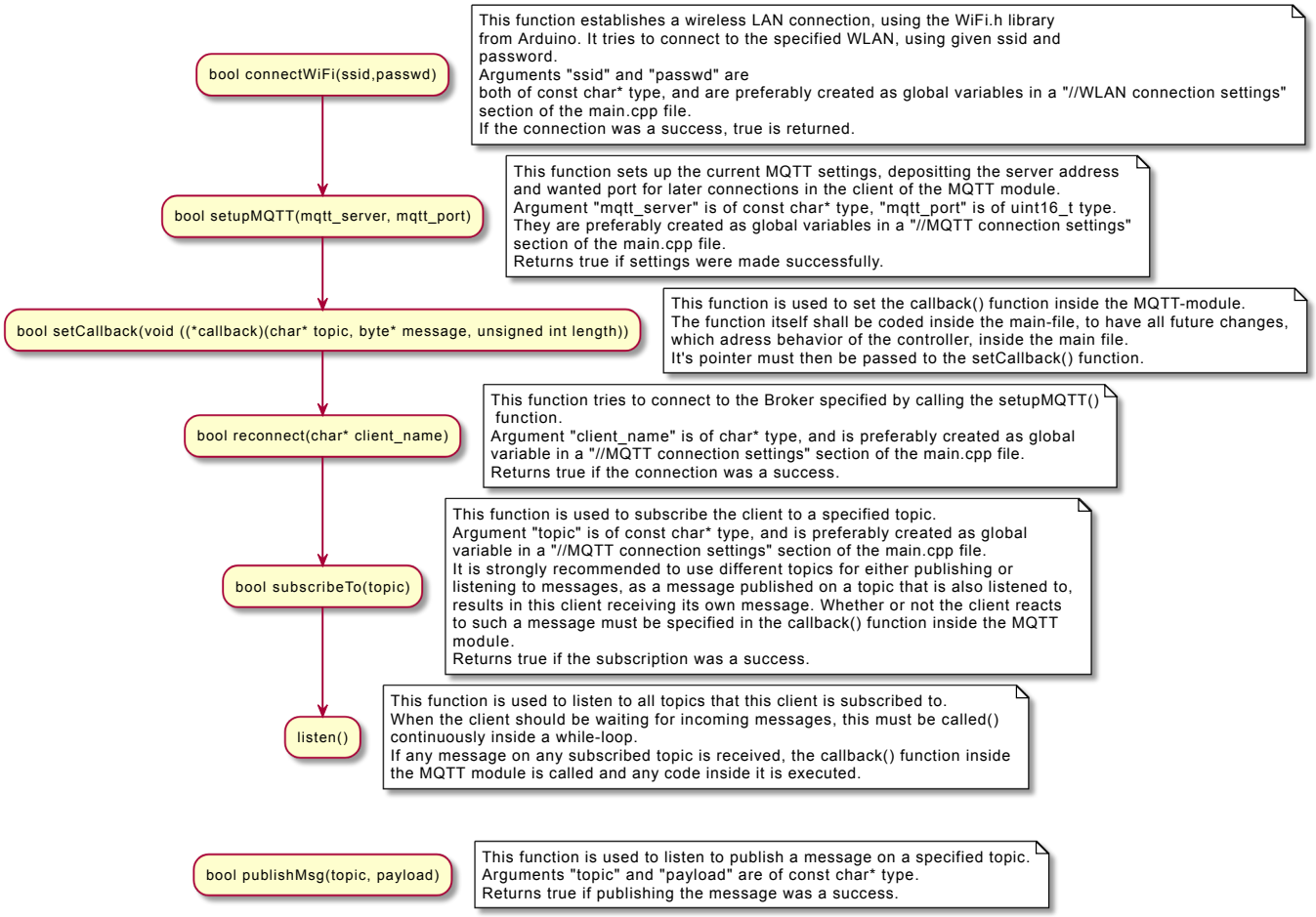


Create Mqtt Server on Pi



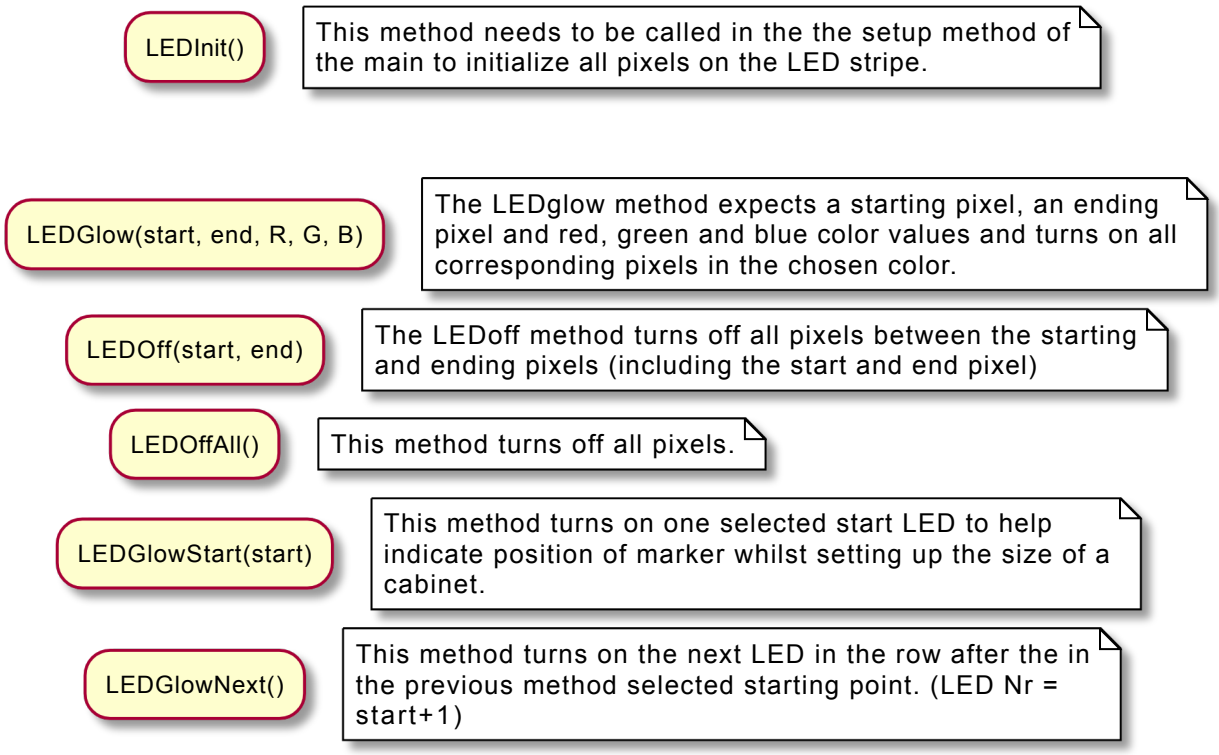
Using the WLAN & MQTT Module on esp32

Include the esp32_wlan.h and esp32_mqtt.h Header-Files before trying to use any functions. Moreover, these modules rely on the PubSubClient Library by Nick O`Leary, so this must be included in your project, too. Sequence of correct function calls + explanation:



Using the (current) LED controls for a LED stripe

This module is designed to be used via Platform.io. Place this header file inside your src directory and use the implemented methods in your code. To use the LED.h header you need to make sure the Adafruit NeoPixel library and the Arduino.h are installed in your project. Make sure the LED Stripe is connected to PIN 2 on your board and you initialize your Pin 2 as a digital output inside your setup method. This library works staticly with one connected LED stripe and you dont need to create any objects whatsoever.


LEDInit()
This method needs to be called in the the setup method of the main to initialize all pixels on the LED stripe.
LEDGlow(start, end, R, G, B)
The LEDglow method expects a starting pixel, an ending pixel and red, green and blue color values and turns on all corresponding pixels in the chosen color.
LEDOff(start, end)
The LEDoff method turns off all pixels between the starting and ending pixels (including the start and end pixel)
LEDOffAll()
This method turns off all pixels.
LEDGlowStart(start)
This method turns on one selected start LED to help indicate position of marker whilst setting up the size of a cabinet.
LEDGlowNext()
This method turns on the next LED in the row after the in the previous method selected starting point. (LED Nr = start+1)