BLE MQTT Gateway

the idea here is to have an esp32 based gateway which detects BLE Beacons and reports them to an MQTT server for presence detection to open doors, turn on lighs etc.

On the software side i decided to go with the OpenMQTTGateway

After Reading some git hub issues and blog posts I decided to trop the esp32 and go for an esp8266 combined with an HM10 module. for the esp8266 i will be using a wemos d1 mini. the nice thing is, that the HM10 module, if purchased on an SZ-040 (or AT-09) board fits right underneath the esp8266. combine some of the sockets that come with the wemos d1 mini and bend some pins on the ZS-040 board and you won't have to use a single wire.

here is how it connects:

pin on ZS-040	pin on D1 mini
RX	D6
TX	D7
VCC	5V
GND	G

HM-10 Module

If you've purchased your HM-10 modules for less than $2 most likely you where sold a knock-off, as was I. This means that you most probably need to flash the HM-10 firmware first, otherwise the whole AT-Commands won't work and you can't use it for this project.

in order to do so, i had do solder some wires onto my HM-10. There are different ways to flash the firmware onto one of these copies. They all involve a software called CCLoader and most of them require an arduino of some sorts to connect to the HM-10. Now here comes an issue: with most HM-10 modules on the market, you will need to use some Logic Level Converter to convert from the 5V logic level of the arduino to the 3.3V logic level of the HM-10 board. On the back of my module it says “Power:3.6V-6V” and “Level:3.3V” which means i can feed it with 5V as supply power, but the logic levels are only 3.3V compatible, so DON'T connect it directly to your arduino one for example!

since i did not have some logic level converters at hand and neither did I have an arduino that uses 3.3V logic levels, i needed another option. There is a port of CCLoader to the raspberry pi. the raspberry pi uses 3.3v logic levels, i had an old one laying around and most importantly it makes for a nice integrated system with no dependencies on any other computer. so if you hae an old raspberry you can't use for other stuff this is the perfect application to put on there.. like this whenever you have an additional HM-10 to flash, you take your raspberry, connect it to the HM-10 and flash it.. no long preparations after the first one is done :)

here is how i wired the three pins that i had to solder onto the HM-10 .. forget holding them in place, the flashing process takes about 10 minutes (with the checking enabled that is).

sorry, this will have to follow later on.. basically i connected the DD,DC and RESET pins to the raspberry GPIO14,15 and 18.

for the power supply i connected the +5V pin of the traspberry to the AT-9 board's VCC and the ground of the raspberry also to the AT-9's ground.

i then downloaded the latest version of CCLoader-RPi to my raspberry and compiled it:

```
gcc src/CCLoader.c -std=c99 -lm -o CCLoader
```

now for the firmware you can't download the original HM-10 firmware form the manufacturers webpage because they don't include the bootloader which we also need for our module. but there is a version out there in the web which contains also that. google for `CC2541hm10v540.bin` to find a copy. there should be one available on this site.

now copy that binary into the same folder as your compiled CCLoader and run the software. it should look like that:

```
pi@raspberrypi:~/CCLoader-RPi-master $ sudo ./CCLoader --RESET=14 --DD=15 --
DC=18 CC2541hm10v540.bin
RESET: pin 14
DD: pin 15
DC: pin 18
Firmware loaded, 256 kB
Chip ID = 0x41
Erasing chip...
Chip erased.
XOSC has stabilized.
Flashing firmware... (511 blocks written, 99% done)
Flashing has completed successfully.
```

as i said, the flashing took an estimated 10 minutes, it shows you a progress where you will see the first blocks reported after a few seconds.

Flashing

with the hardware in place let's get the software side done: first I setup_arduino_ide_for_esp8266. Now download the latest release of OpenMQTTGateway and extract it. Rename the folder to `OpenMQTTGateway` (remove the version) so that you can open it in the arduino IDE.

open the `OpenMQTTGateway.ino` file in your prepared arduino gateway and go through the `User_config.h` settings for your wifi, passwords and mqtt settings. In the module selecton comment out all the modules and remove the comment from the ZgatewayBT module. Next go through the settings in `config_BT.h`.

now Copy the contents of `OpenMQTTGateway/lib` to `<sketchbook folder>/libraries` in your arduino ide folder

NOTE make sure you have SPIFF enabled in the board settings of the arduino IDE. it seems OMG wants to use that to save config changes

the web-interface to configure the wifi and mqtt settings seems to only be available in access point mode.