Mqtt packets v0.2

# structures

We can think about having a welcome packet to each device. Each devices have a private topic to subscribe:

Sub/MacAddress

After subscribing to the private topic, in the welcome packet, the devices knows in which building, floor, .., is mounted and what device is. The benefit of this proposal over the previous one is that the device location can, and its puzzle type can be changed easily. On the other hand, in this solution, the code for all the puzzles must be flashed on the microcontroller.

# Messages:

All the messages must be published in JSON format.

Each message has the following fields.

Timestamp: as the devices are not connected to the internet, timestamp is sent by the server at the beginning.

MessageType: int

Determine kind of message  
The possible types:

1. Reset: reset the devices.
2. Hint: request hint from the manager
3. Data: a structure based on puzzle type information.
4. Status: a packet containing status information

## Reset

1. Type: indicating this is a reset message.
2. Timestamp: to check fresh messages if necessary.
3. ?

## Status

Status packet is reported every 2 minutes to indicate that the device is active and healthiness of the devices that are connected to (like RFID reader, …?) and the status of relays and the mag locks.

1. Type: indicating this is a reset message.
2. Timestamp: to check fresh messages if necessary.
3. Id: The id of the device that is determined by the dip switch.
4. statusData:

lastTimestamp: int

inRfid: bool

outRfid: bool

maglock1: bool

maglock2: bool

k1: bool

k2: bool

lcd?

## Add subscription topic:

1. Type: indicating this is a reset message.
2. Timestamp: to check fresh messages if necessary.
3. Topic: The topic that needs to be subscribed.

## Add publish topic:

1. Type: indicating this is a reset message.
2. Timestamp: to check fresh messages if necessary.
3. Topic: The topic that needs to be published into.

## Remove subscription topic:

1. Type: indicating this is a reset message.
2. Timestamp: to check fresh messages if necessary.
3. Topic: The topic that needs to be removed from subscription list.

## Remove subscription topic:

1. Type: indicating this is a reset message.
2. Timestamp: to check fresh messages if necessary.
3. Topic: The topic that needs to be removed from publish list.

## Hint

1. Type: indicating hint message.
2. Timestamp: to check fresh messages if necessary.
3. puzzleNo: The number of puzzle that is needed for help. (Maybe there are numerous puzzles on single device).
4. ?

## RFID:

1. Type: indicating rfid message.
2. Timestamps: to check fresh messages if necessary.
3. Rfid: tag of the read rfid.
4. Side: true for indoor and false for the outdoor.

## Action:

1. Type: indicating action message.
2. Timestamps: to check fresh messages if necessary.
3. BuiltInLED: the status of built-in led.
4. validDoors: if the door must be controlled.
5. Doors: true for opening the doors and false for closing it.
6. validInLcd: if the messages on next fields must be printed on the indoor LCD.
7. inLcd0: message that must be printed on the indoor lcd row0.
8. inLcd1: message that must be printed on the indoor lcd row1.
9. validOutLcd: if the messages on next fields must be printed on the outdoor LCD.
10. outLcd0: message that must be printed on the outdoor lcd row0.
11. outLcd1: message that must be printed on the outdoor lcd row1.
12. ValidServos: if the motors must be controlled.
13. Servos: an array of the servo motors angle.

## Status request:

1. Type: indicating status request message.
2. Timestamps: to check fresh messages if necessary.

## buildInLED

1. Type: controlling built-in led on the PI PICO.
2. Timestamps: to check fresh messages if necessary.
3. Status: true for turning on and false for turning off the built-in LED.