it changes (significantly)EESTech Challenge 2019 - Tasks 5-6

Tasks and study material

## Task 5.1 - MqttStatus

**Goal: Report status messages periodically and publish to MQTT.**

**Score: 30**

Send the JSON-formatted status content periodically (once every 30s) to an MQTT topic. It's the same content that is in the HttpBasicStatusJson task.

Topic: /teams/teamX/devices/<chipId>/status

Payload: the same as the contents of http://<device IP>/status?format=json

## Task 5.2 - ControllablePeriods

**Goal: Make the temper1554589542ature, humidity and light measurement periods controllable.**

**Score: 20**

To set the temperature and humidity measurement period to every 30s, use this:

Topic: /teams/teamX/devices/<chipId>/commands/set\_property

Payload: dhtMeasurementPeriodMs=30000

To set the light measurement period to every 5s, use this:

Topic: /teams/teamX/devices/<chipId>/commands/set\_property

Payload: lightMeasurementPeriodMs=5000

As a result, the corresponding measurements and events will have a different frequency.

Also, add these values to the status JSON content, like this:

{

"team":"teamX",

…

"properties":{

"dhtMeasurementPeriodMs":30000,

"lightMeasurementPeriodMs":5000

}

}

Serve this new content over HTTP for /status?format=json and send it in the /status MQTT messages.

## Hardware - NeoPixel / WS2812B

See <https://sites.google.com/view/eestech-2019-zurich/study-material#h.p_HOsPw0tmoX_v>

## Task 6.1 - NeoPixelBasic

**Goal: Make a red, a green and and a blue pixel circling the LED ring.**

**Score: 15**

Create an animation which puts 3 pixels on the LED ring, they should be "uniformly" distributed, always at a 120 degree angle, same distance (8 LEDs) between them.

One pixel should be red, one green, one blue.

The three pixels should constantly go around at a speed of about 0.3-1 full rotations per second.

It is important that there should be **no "glitch"** at the edges, **the animation should be smooth**.

**Evaluation:**

If you're ready, call for one of the organizers to visually inspect the animation.

## Task 6.2 - NeoPixelHttp

**Goal: Build a NeoPixel-based analog clock that is controllable over HTTP.**

**Score: 15**

Implement the following API:

http://<device IP>/neoclock?h=13&m=25

h: hours

m: minutes

The "clock" should have two hands:

* **red pixel**: hour hand
* **green blinking pixel**: minute hand

The minute hand (green pixel) should cover the hour hand.

The clock should work in the 00:00 to 23:59 range.

It doesn't have to handle invalid times.

The clock doesn't have to advance.

The hour hand has to be able to point at all the 24 pixels (not just 12 of them).

For example at the following times:

* 1:00
* 1:30
* 2:00

the hour hand should point at three different pixels (next to each other).

**Evaluation:**

If you're ready, call for one of the organizers to evaluate using a combination of sending requests and visual inspection.

## Task 6.3 - NeoPixelMqtt

**Goal: Control the NeoPixel ring using MQTT commands**

**Score: 20**

The device should subscribe to the following topic:

Topic: /teams/teamX/devices/<chipId>/commands/neopixel

<chipId>: is the HEX chipId that is reported under /status?format=json.

Implement the following API

1. Payload: clear → **Special command.** Clears all LEDs.
2. Payload: set all:#101010 → **Sets all LEDs** to R=16, G=16, B=16
3. Payload: set 0:#200000 → **Set specific LED.** Sets LED #0 to R=32, G=0, B=0
4. Payload: set 23:#200000 → **Set specific LED.** Sets LED #23 to R=32, G=0, B=0
5. Payload: set 0-12:#FFFFFF → **Set range of LEDs.** Sets LEDs #0-#12 (half circle) to full white
6. Payload: set all:#000000;1:#101010;2-3:#302010;8-10:#001000 → **Composition.** resets all to off first and turns on #1, #2-#3 and #8-#10 LEDs to different colors

**Evaluation:**

If you're ready, call for one of the organizers to evaluate using a combination of sending requests and visual inspection.