Light alarm - Using a react-native client in combination with a rust backend.

1st Maksim Sandybekov
computer science - autonomouse systems)
HTWG Konstanz
Konstaz, Germany
maksim.sandybekov@live.de

2nd Benjamin Bäumler
computer science - autonomouse systems
HTWG Konstanz
Konstanz, Germany
be391bae@htwg-konstanz.de

***Abstract*—**
***Index Terms*—rust, react-native, redux, redux-saga, light, alarm, smart-light**

The main tasks of the server are to communicate with clients and to actually drive a led strip or any other kind of light source.

A. Light Sources

The server can support different kind of light sources, as long as it implements the `LedControls` trait. Only one implementation of the trait can be used at the same time tho. But it is possible that a specific implementation controls multiple hardware lights. At the current time there are only two implementations that can be used as light sources:

- LedStrip
  Controls a 4-pin led strip with a 12v pin and one pin for each color(red, green,blue). This cannot be driven directly by the Raspberry Pi and therefore we need to use a extra curcuit board [1] for it. The circuit board allows us to controll each color seperatly by driving the gate pin of a MOSFET's respectivly. We will use the pigpio Daemon [2] for this, because we need 3 PWM pins for this and currently available gpio libraries for Rust only over up to 2 PWM pins.
  picture of circuit board
- MocLedStrip
  This is used only for testing. It allows us to verify the logic of the led controller without driving any GPIO's of the Raspberry Pi and also allows us to run the server on amd64 architecure for testing purposes.

References

[1] Raspberry Pi & RGB LED-Strips How to control a RGB LED-Strip with a Raspberry Pi By David Ordnung - https://dordnung.de/raspberrypi-ledstrip/
[2] pigpio Daemon - http://abyz.me.uk/rpi/pigpio/pigpiod.html