School of Engineerin

Moodlight Concept Report

Date	09 April 2018
Major	electrical engineering
Subject	electronic project 2 Spring semester 2018
Lecturer	Andreas Ehrensperger
Students	Roman Frick Alen Kamenjas

1 Table of contents

2 Abstract

This document represents the concept of our moodlight. The individual circuit parts are explained in more detail. It also handles the software concept of the microcontroller and the concept of the android application.

During the 6th semester of studying electrical engineering at the ZHAW, the students develop a project which is called “Moodlight”. In the previous semester, we developed a moodlight prototype. This included two different kinds of circuits to control the power led and the software for the microcontroller. The final version of the moodlight will only take one kind of circuit to control the led. The moodlight will be controlled by the microcontroller EFM32, which comes from silicon labs. The microcontroller receives advices via Bluetooth. For this, an android application app will be programmed. Via the android application, the desired colour can be set or the special function can be controlled.

In addition to the basic function, the moodlight should also be able to carry out additional functions. As an example, the colour intensity should be controlled depending on the ambient brightness. Next we measure the ambient noise via a microphone and after surpassing a certain predefined level, the moodlight should start to flash. By using a temperature sensor, the temperature should be measured and the colour should be able to adapt itself to the current temperature. The current ambient temperature will then be displayed on the smartphone via the android application.

3 Requirements and overall concept

For the moodlight project, a lot of components are required. The project does not only include electrical components but also requires the case of the moodlight with a light diffusor.

The first step was to develop the schematics and to carry out some experiment set-ups with breadboards. After that, we designed the PCB and handed in our files. While the PCB was being produced externally, we started to program the android application and the microprocessor software.

The following picture describes the moodlight's in- and outputs and provides an overview of how the different parts work together.

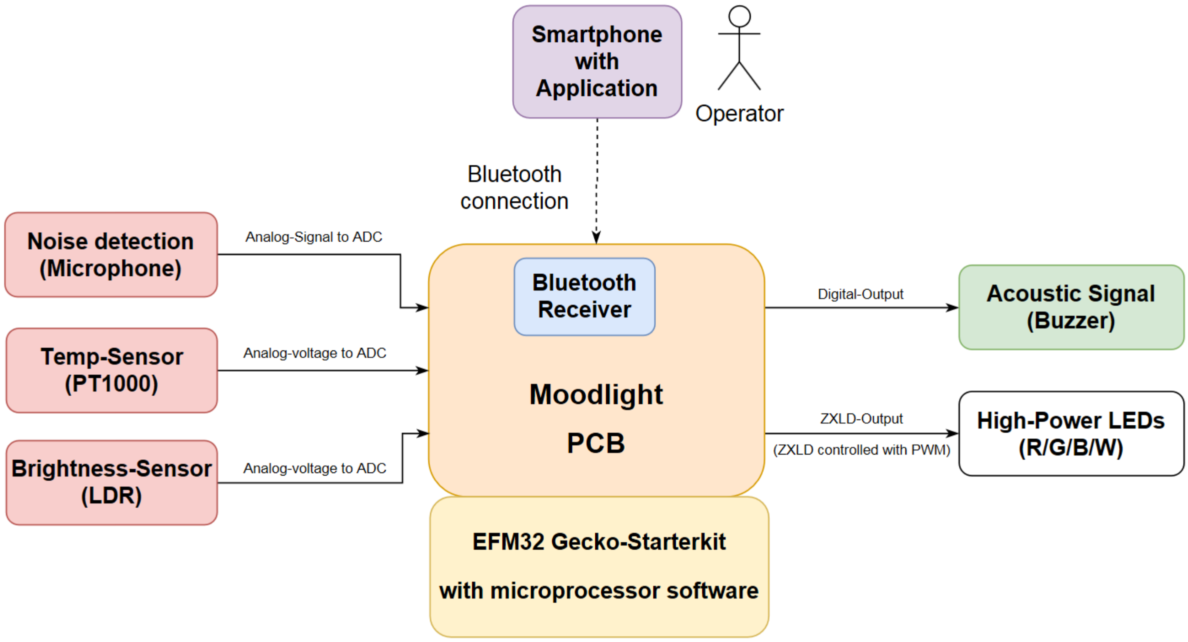

Figure 1: moodlight concept

4 Hardware concept

Figure 2 shows a rough overview of the concept of the moodlight. Below the figure, the individual circuits are explained in more detail.

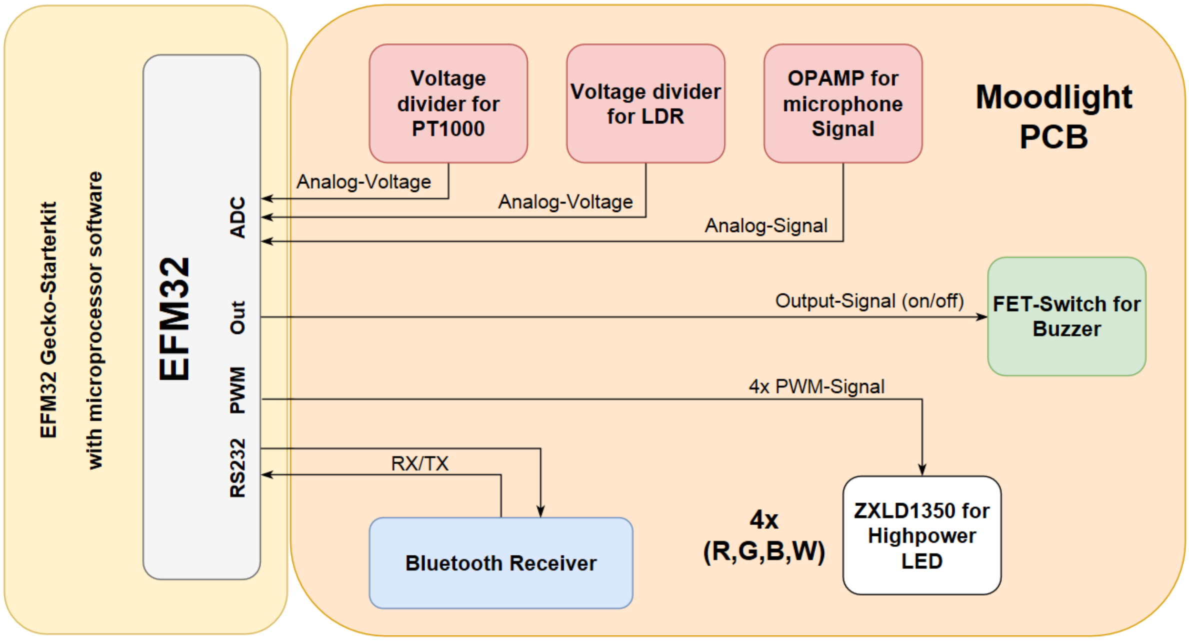

Figure 2: hardware concept

4.1 Temperature and LDR sensor

To measure the analog voltage from PT1000 and LDR, we designed two simple voltage dividers. Both sensors change its resistor depending on the brightness or the temperature. This generates a variable voltage which will be measured with ADC-Inputs from EFM32.

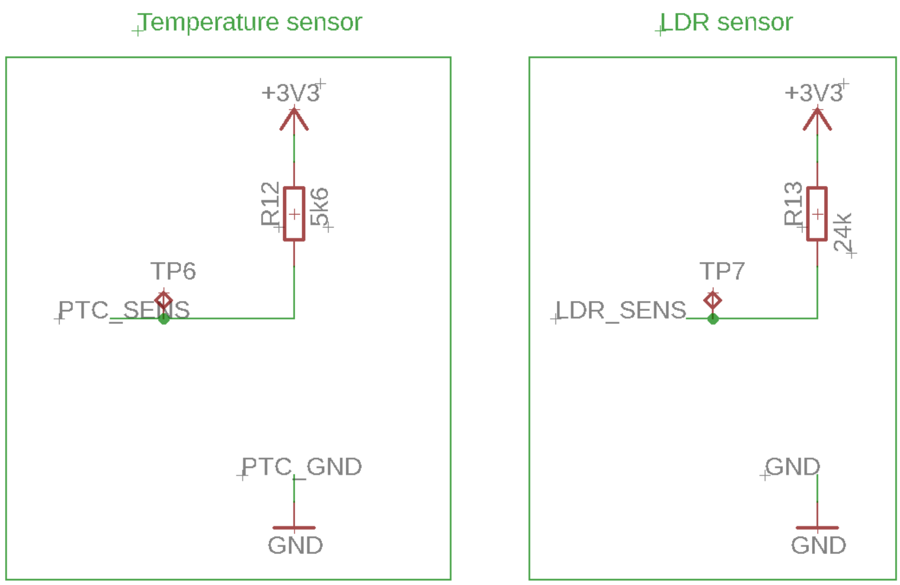

Figure 3: schematic temperature and LDR sensor

4.2 Microphone amplifier

The microphone generates a tiny signal depending on the noise of the ambience; this signal runs through an OPAMP-circuit that will reinforce the signal with a factor of 830. After that, the signal will be smoothed through a RC-lowpass-filter which generates the envelope from the measured signal. An ADC from EFM32 measures the voltage and the outcoming information can be processed by the software.

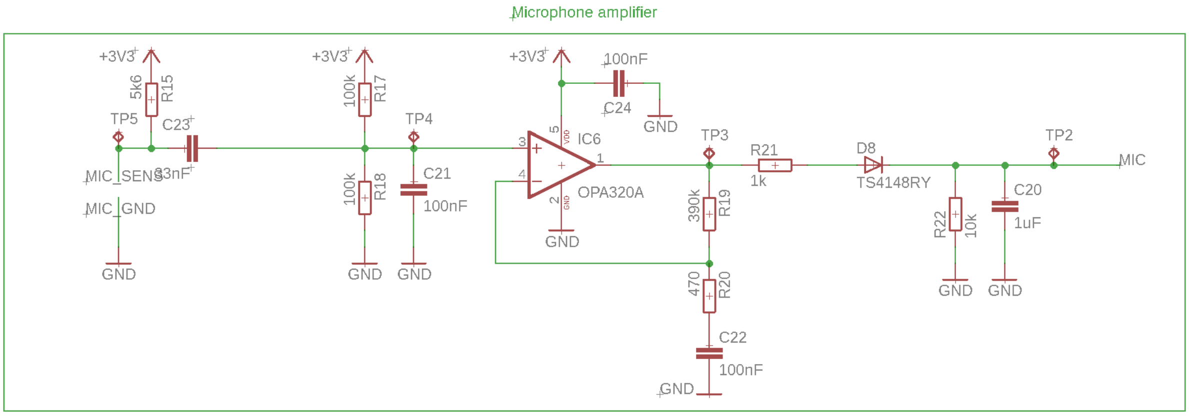

Figure 4: schematic microphone amplifier

4.3 LED control

The high-power LEDs will be controlled by the integrated-circuit ZXLD1350. The ZXLD1350 sets the brightness of the LEDs depending on the duty-cycle from the PWM signal which is generated by the processor. A higher duty-cycle generates a higher current through the LEDs. This circuit is included 4 times, for every colour: red, green, blue and white. With a high capacitor C8 from the schematics below, it is possible to smooth the PWM signal from the microcontroller to an analog DC-voltage, depending on the duty cycle from the PWM signal.

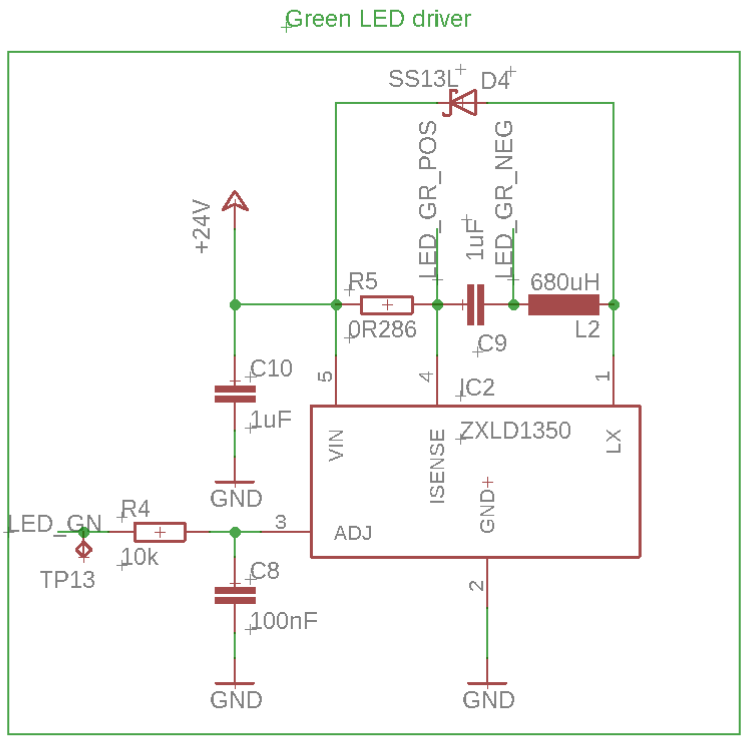

Figure 5: schematic LED driver

4.4 Buzzer control

To turn the buzzer on and off, the moodlight-PCB includes a switching-FET. The output pin-signal of the microcontroller switches the FET on and off.

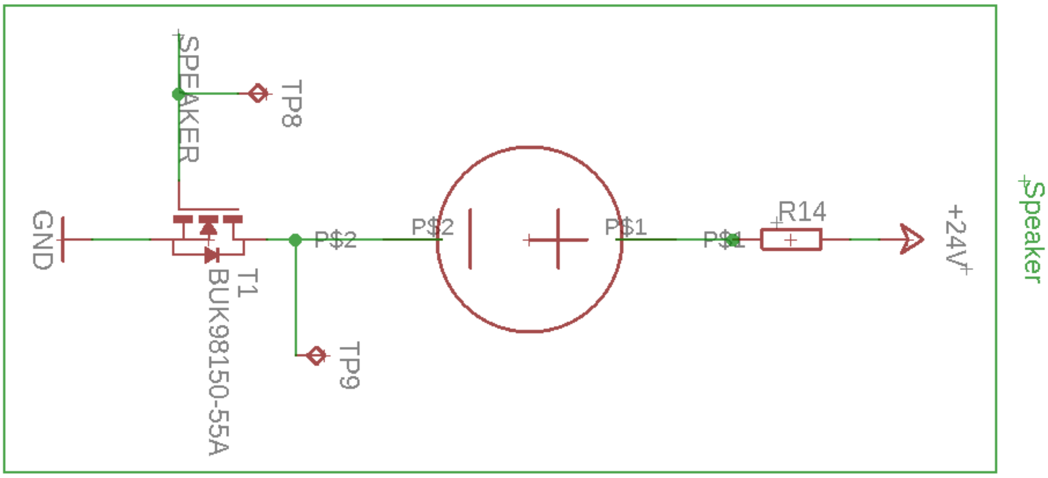

Figure 6: schematic buzzer

4.5 Bluetooth module

To connect the android application on the smartphone with the moodlight, we implemented a bluetooth-receiver in our moodlight-PCB. When the connection has been completed, the smartphone application sends commands over the bluetooth-module RN42 to the processor EFM32 by using UART.

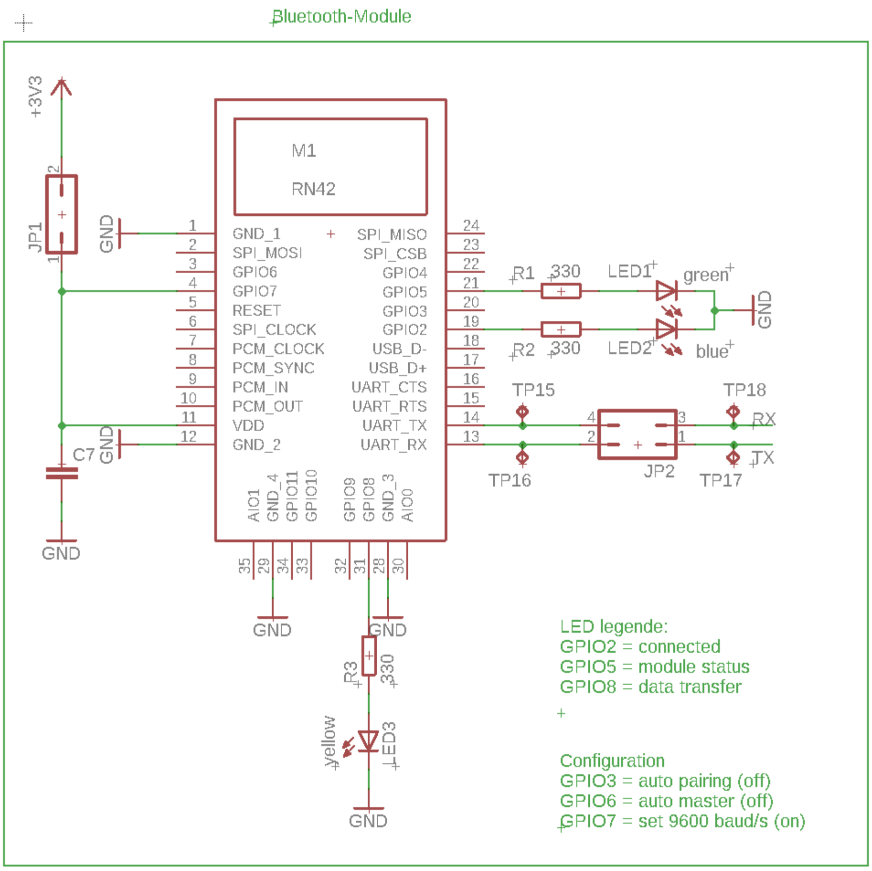

Figure 7: schematic Bluetooth module

4.6 Microcontroller resources and pinout

The moodlight project uses the following resources of the microcontroller:

- Timer0 for PWM 1-3
- LETimer0 for PWM 4
- LEUART0 to communicate with Bluetooth module
- ADC channel CH0, CH6 and CH7 for measuring voltage from PT1000, LDR and microphone
- Output pin PC4 to switch on/off the buzzer

The connections from the Gecko-PCB to the moodlight PCB are listed in the following table.

Pin from Gecko	Name	Application	Remarks
1	GND	GND	
2	VMCU	Supply voltage	
3	PC4	Output pin	Control the buzzer
4	PD0	ADC0_CH0	Control the PTC
6	PD1	TIM0_CC0#3	Control LED green
8	PD2	TIM0_CC1#3	Control LED blue
10	PD3	TIM0_CC2#3	Control LED red
11	PB11	LETIM0_OUT0#1	Control LED white
12	PD4	LEU0_TX#0	Bluetooth TX
14	PD5	LEU0_RX#0	Bluetooth RX
15	PD7	ADC0_CH7	Control LDR
16	PD6	ADC0_CH6	Control microphone
19	GND	GND	

Table 1: pinout gecko

5 Microcontroller SW concept

To implement all features from moodlight, the microcontroller must be able to generate four PWM outputs, measuring 3 different analog-voltages and must be able to communicate with UART.

The following block diagram shows the peripherals of the microcontroller EFM32 which are used in the project moodlight.

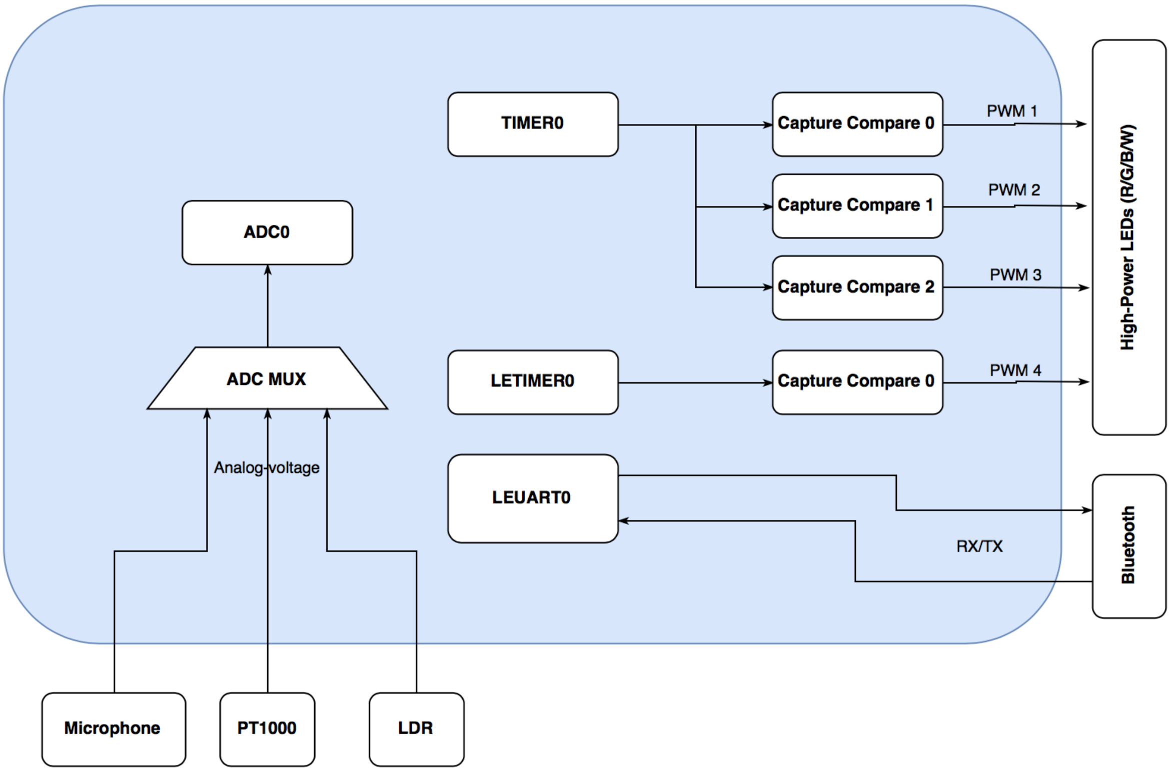

Figure 8: software concept microcontroller

5.1 PWM 1 - 4

The PWM-channels one to three are generated by the TIMER0 with a frequency of 500 Hz. Setting the capture-compare registers zero to two allows to change the duty-cycle from each PWM what generates a different brightness of the LEDs. The PWM channel four is generated with a low-energy timer LETIMER0. This timer generates a PWM with a frequency of 150 Hz. The duty cycle is controlled with the help of capture-compare register 0.

5.2 ADC

To measure analog voltages from sensors, the moodlight uses the ADC0 of the microcontroller. The internal ADC-Mux switches the ADC to the desired pin:

Analog-voltage	Channel from ADC0
LDR	ADC_CH7
Microphone	ADC_CH6
PT1000	ADC_CH0

Table 2: ADC pinout

5.3 UART

The UART communication is implemented by the internal LEUART0 of the microcontroller. The baud rate is 9600 baud per seconds.

The Bluetooth-connection protocol is very simple. To change the state, the three first letters of the command will be compared with the states from state machine. After the command, it is possible to add a value for example to change the brightness of a colour.

Sending the command ***gre**en 140* changes the brightness of the green LED to 140 of 255.

Sending the command ***tem**p* advises the microprocessor to measure and transmit the voltage over the PT1000 to the smartphone which calculates the resulting temperature.

5.4 State machine

The software which runs on the microprocessor is written in C. The project is compiled with the Simplicity-Studio IDE from Silabs. The software implements a state machine with different states, depending on the commands which will be transmitted by the smartphone.

To develop the software, we used the template from ZHAW and adapted it according to our needs. It is possible to control the moodlight directly by using the touch buttons and the touch slider on the Gecko, or by sending commands over Bluetooth from an android smartphone.

There is also the possibility to change or add some states from state machine in the future, but the software concept so far implements the following states:

State	Description
RED	Changing the value from colour red
GREEN	Changing the value from colour green
BLUE	Changing the value from colour blue
WHITE	Changing the value from colour white
MICROPHONE	Measuring noise level from ambience
TEMP	Measuring temperature from ambience
LDR	Measuring brightness from ambience
BUZZER	Sending alarm beeps
IDLE	Idle state after special-functions

Table 3: state of the microcontroller

To jump from state to state, it is possible to use the buttons. To change the value from the LEDs, we can use the touch slider. Another way is to send the command from a smartphone via Bluetooth, which includes the state and the value in only one command.

6 Android SW concept

This chapter includes a description of the android app. It can happen that the final version of the app does not completely match the concept. We used a template from ZHAW which already implemented the basic functions.

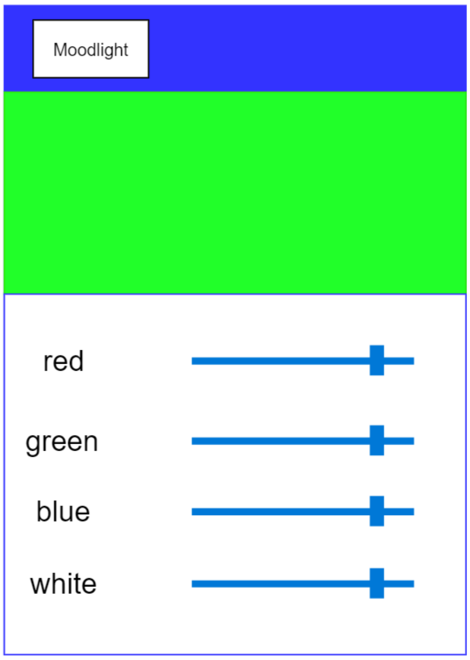

Figure 9: main screen android app

When the app is opened by the user, the smartphone should automatically connect to the Bluetooth module in the moodlight and open the main page shown in Figure 9. Each colour of the moodlight can be controlled individually by the seekbar. The current set value will be shown above the seekbar. In the upper half of the main page, the current set colour will be displayed.

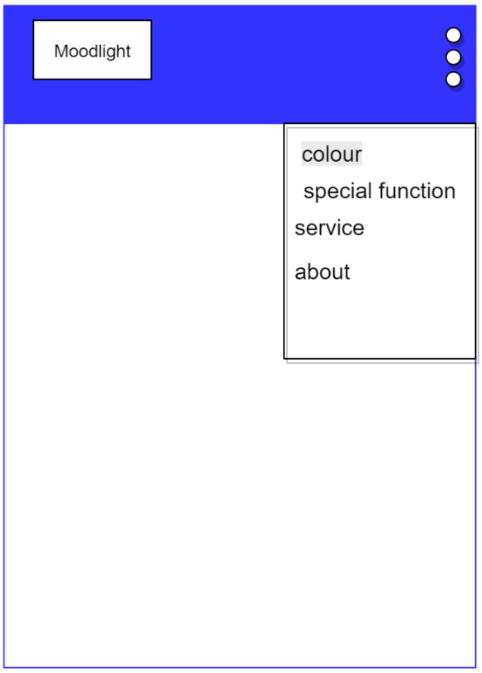

Figure 10: menu android app

Figure 10 shows the sketch of the menu. For the menu, a dropdown menu has been used, which shows all implemented functions.

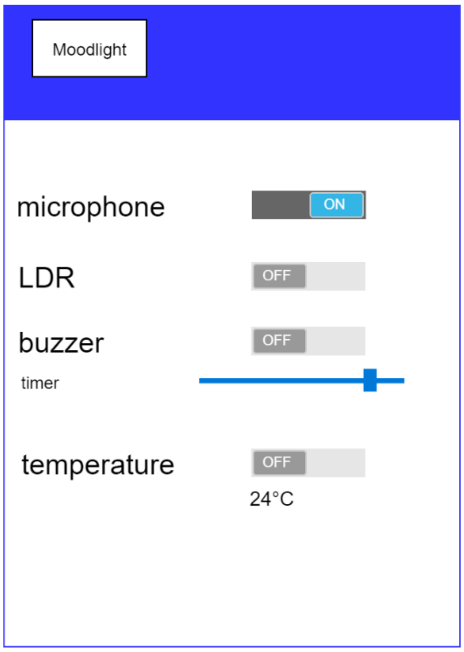

Figure 11: special function screen android app

Fehler! Verweisquelle konnte nicht gefunden werden. shows the special function screen. This part of the app controls all hardware special functions, which are implemented in the moodlight. With a slider button, the special function can be connected or disconnected with the moodlight. For the temperature function, the current outside temperature is still displayed when the function is switched off.

The timer seekbar is to use the buzzer function. By adjusting the seekbar, a waiting time up to 180s can be appointed. When the time runs out, the buzzer begins to whistle.

Figure 12: service screen android app

The service screen, displayed in Figure 12, is for the manual service of the moodlight. Buttons make it possible to connect or disconnect the smartphone with the moodlight. Furthermore, it should be possible to send single commands to the moodlight, for example red 230. This means that the microcontroller adjusts to the menu red and the value 230 will be appointed.

7 Additional features

We decided to implement four additional hardware features in our moodlight. The first feature is a microphone that is able to measure the noise level of the ambience and the moodlight will change the colour as a consequence of the measurement. The moodlight will also be able to measure the brightness of the ambience with a light dependent resistor. Further it is possible to define the outside temperature with a PT100. The last additional feature we are going to implement, is a buzzer. With the buzzer, we can generate some acoustic signals. The following subchapters will explain the additional features in more detail.

7.1 Microphone

The microphone is detecting the noise level of the surrounding ambience. With a RC-lowpass-filter, the ambience signal will be smoothed to an enveloppe. Which feature the microphone will finally realize, is not clear yet. This is part of the software. Some ideas are for example a clap sensor, controlling the moodlight- brightness by the actual noise level or blinking synchronous to a sound, which will be played externally.

7.2 LDR

With a simple voltage divider, we measure the ambience brightness. With this function, we can dim the moodlight if the ambience is getting darker.

7.3 PT1000

The moodlight measures the ambience temperature with a PT1000-sensor. The exact value of the temperature will be transmitted to the smartphone and the moodlight will change its colour depending on the temperature. This could probably be blue for cold outside temperatures and red if the temperature is increasing.

7.4 Buzzer

The DC-buzzer will be set on or off by using an I/O-Pin of the microprocessor. A possible application for the buzzer could be the setting of a timer, where the moodlight is for example going to ring after exactly 3 minutes. Another application for the buzzer would be to send a short beep after receiving a command so that the operator is able to hear that the moodlight has understood the command.

8 Appendix functional test

During the development phase, we carried out some functional tests, because the PCB will be the final version and it will not be possible to make any adjustments later.

The picture below shows the test arrangement of the microphone amplifier. The circuit is built on a breadboard. To check the functionality, we examined the input and output on the scope. When clapping, the peak must be visible at the input. At the output an envelope should appear, as shown in Figure 14. The yellow line shows the output, the blue line the input.

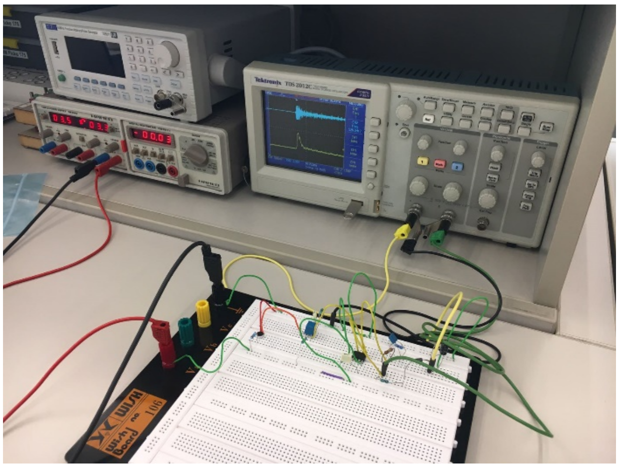

Figure 14: microphone amplifier

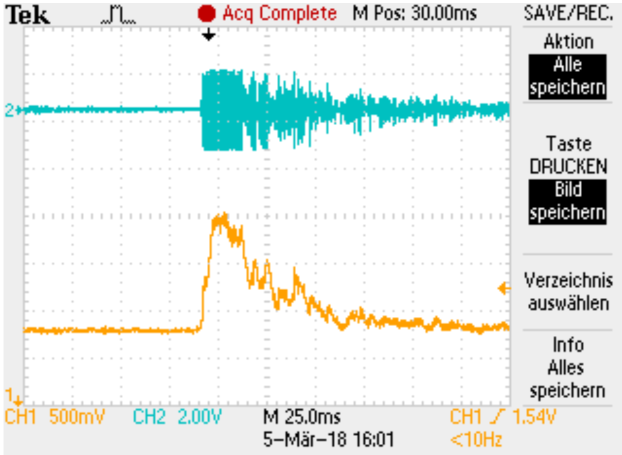

Figure 13: scope measurement of a clap

Device name	Serial number	Remark
EL302RT Triple	NT12.72	Power supply for 3.3V
Tektronix TDS2012C	NT16.66	Scope
Breadboard	-	-

Table 4: test equipment

For this project we needed Bluetooth to control the moodlight. So we decided to put a Bluetooth module on a self-made PCB and connect it to the GECKO development board. To check the connection, we used a smartphone and the android application “Bluetooth terminal”. The blue led on the PCB shows if the smartphone is connected to the Bluetooth module. If the connection stands, it is possible to send commands. When you send <green 123>, the GECKO must display the state <green> and the value <123>, shown in Figure 15.

Figure 15: bluetooth test