Extra Feature

Moodlight

I. Krajinovic, L. Müggler

Lecturer: M. Nussberger

ZHAW School of Engineering ET18

Winterthur, 06.06.2020

Table of Content

1. Overview

This documentation describes the specifications of the Moodlight's extra feature "Temperature-mode". It includes hardware, software report and test results of this extra feature.

The temperature mode adapts colour spectrum as a function of the ambient temperature. This mode can only be selected when operating with the smartphone app.

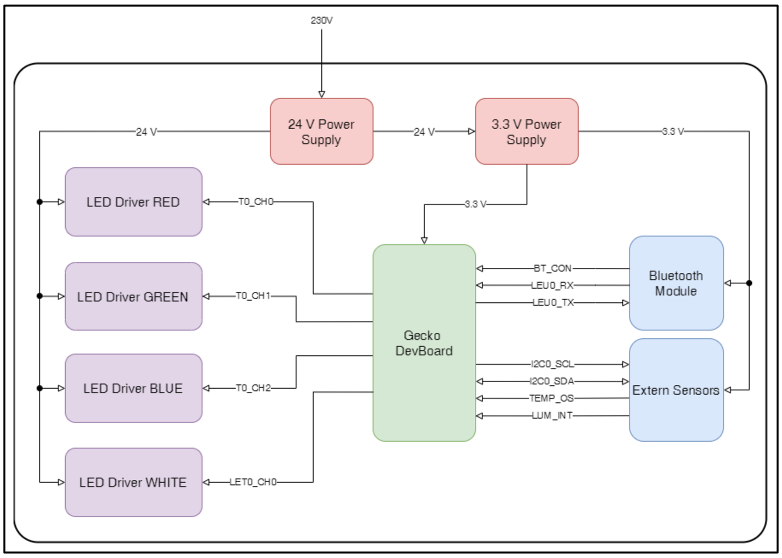


Figure 1: Block diagram of the Moodlight

2. Specifications

This mode adapts the colour spectrum as a function of the ambient temperature. On a user specified low temperature limit, the Moodlight shines completely blue. The warmer it gets, the greater the red component and the lower the blue component. At a user specified high temperature limit the Moodlight shines completely red.

The temperature of the Moodlight is displayed in the smartphone app and is updated each second. For security reasons a notification on the app appears when the temperature passes a defined overtemperature limit of 50°C. As long as the temperature is above this limit, the app does not send any data. The mode gets changed to the *RGBW* and the LEDs are getting switched off.

3. Hardware

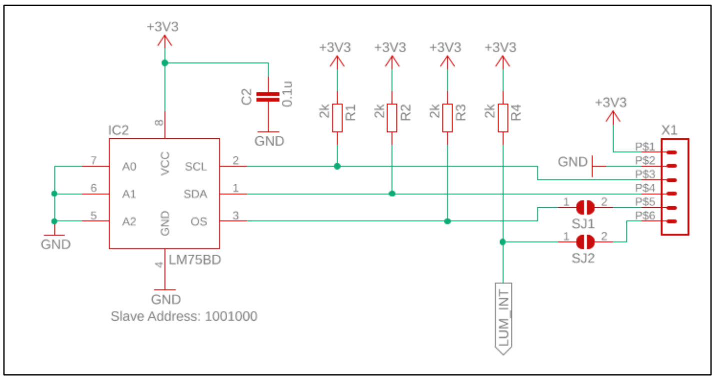


Figure 2: Temperature Sensor

The temperature sensor *LM75BD* is connected to the µC (EFM32G290F128 Gecko DevBoard) via I^2C interface. The device address is partially adaptable with the pins A0, A1 and A2. For a working I^2C communication external pull-up resistors between $1.5\ k\Omega$ and $10\ k\Omega$ are needed. The I^2C operates in standard mode which means frequencies up to $100\ kHz$.

µC Pin	LM75BD Pin	Signal Name
PD7	2	SCL
PD6	1	SDA
PD0	3	OS (Overtemperature Shutdown)

Table 1: Connection to Gecko DevBoard

3.1. Used components

Component	Value	Properties
IC2	LM75BD	Slave Address: 0x48
R1, R2, R3, R4	2 kΩ	Power Rating: 125 mW
C2	0.1 µF	Voltage Rating DC: 50 V
SJ1, SJ2	Solder Bridge	Not Connected
X1	Molex Micro-Fit Connector	Part Number: 43650-0612
U2	EFM32 Gecko DevBoard	EFM32G290F128 (On separate PCB)

Table 2: Used components for temperature sensor

4. Software

4.1. Specification of the LM75BD

- Temperature range from -55 °C to +125 °C.
- Temperature accuracy
    - ±2 °C from -25 °C to +100 °C
    - ±3 °C from -55 °C to +125 °C
- 11-bit ADC
    - Resolution of 0.125 °C

4.2. Conversion of temperature

The data in the temperature register of the LM75BD is arranged as follows:

Upper Byte								Lower Byte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X
±	Temperature value (temperature)										-	-	-	-	-

Table 3: Structure of temperature register

The highest bit in the upper byte (D10) indicates whether the temperature is positive or negative (D10 = ‘0’ → positive / D10 = ‘1’ → negative). The following ten bits are used for the temperature value. The resolution of the measured temperature is 0.125 °C. Therefore the temperature value has to be divided by 8 to get a corresponding temperature in °Celsius if D10 = ‘0’. If D10 = ‘1’, the two’s complement of the temperature value has to be divided by 8. In this case the following value has to be interpreted as a negative value.

Calculation for temperature value [°C]:

$$tempValue = \begin{cases} \dfrac{temperature}{8} & D10 = 0 \\ \dfrac{temperature - 1024}{8} & D10 = 1 \end{cases}$$

4.3. Temperature measurement

The microcontroller does four measurements at equally spaced intervals of 250 ms. This time intervals are implemented with timer interrupts. Each second the average value of the temperature gets calculated with the measurements. After having this temperature value, the next step is to check if it's exceeding a value of 50 °C what would indicate over-temperature. If this is the case the LEDs get switched off. Additionally the "overTempFlag" gets set and the user is getting informed via the app that the device overheats. As long as the temperature is above 45 °C the Moodlight can't be used (hysteresis). Which means the Moodlight ignores all command which are sent by the app. Despite overheating, the device temperature gets still transmitted to the app.

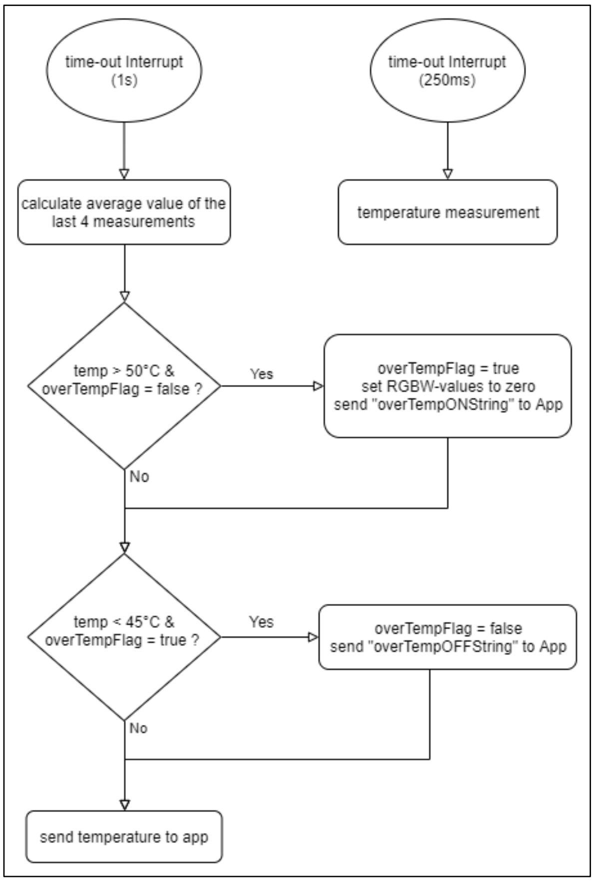


Figure 3: Flowchart for the temperature measurement

4.4. Light dependent colour spectrum

The extra feature allows to display the temperature into a colour scheme. This means that "low" temperature increases the blue component and a "high" temperature increases the red component of the output colour as shown in Figure 4. When reaching or exceeding the upper temperature limit, the colour spectrum consists only of red component and when reaching or falling below the under temperature limit the colour spectrum consists only of blue component. The colour resolution of the Moodlight is depending on the range of the temperature limits. Therefore, the greater the difference between the upper and lower temperature limit, the less visible is a temperature change of 1 °C. This resolution can be changed indirectly by changing the limits on the app (see in Figure 4).

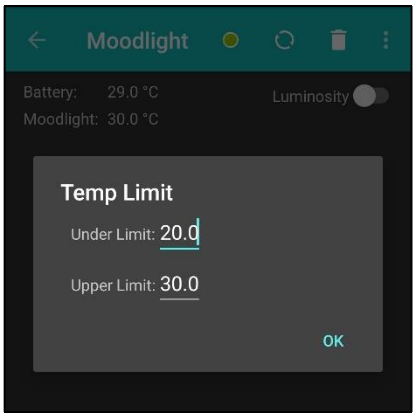


Figure 4: changeable temperature limits on the app

The calculations of the red and blue component are as follows:

$$Resolution = \frac{255}{UpperTempLimit - UnderTempLimit}$$

$$Red\ Component = \begin{cases} 0 & tempValue \leq UnderTempLimit \\ (tempValue - UnderTempLimit) \cdot Resolution & UnderTempLimit < tempValue < UpperTempLimit \\ 255 & tempValue \geq UpperTempLimit \end{cases}$$

$$Blue\ Component = \begin{cases} 255 & tempValue \leq UnderTempLimit \\ (UpperTempLimit - tempValue) \cdot Resolution & UnderTempLimit < tempValue < UpperTempLimit \\ 0 & tempValue \geq UpperTempLimit \end{cases}$$

With the temperature limits of 20 °C and 30 °C and the upper equations, the functions in Figure 5 result.

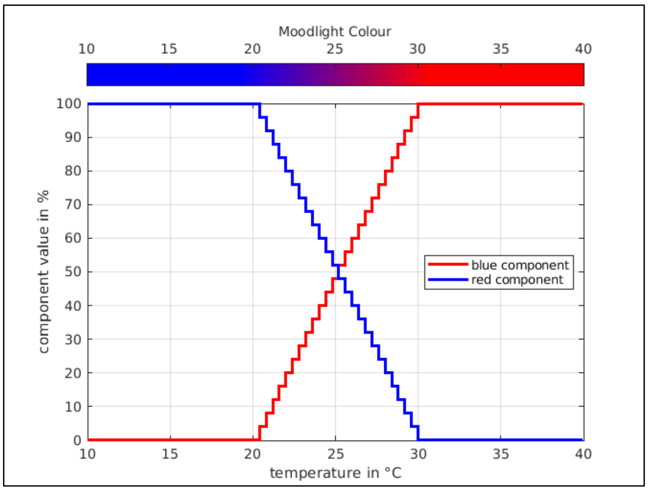


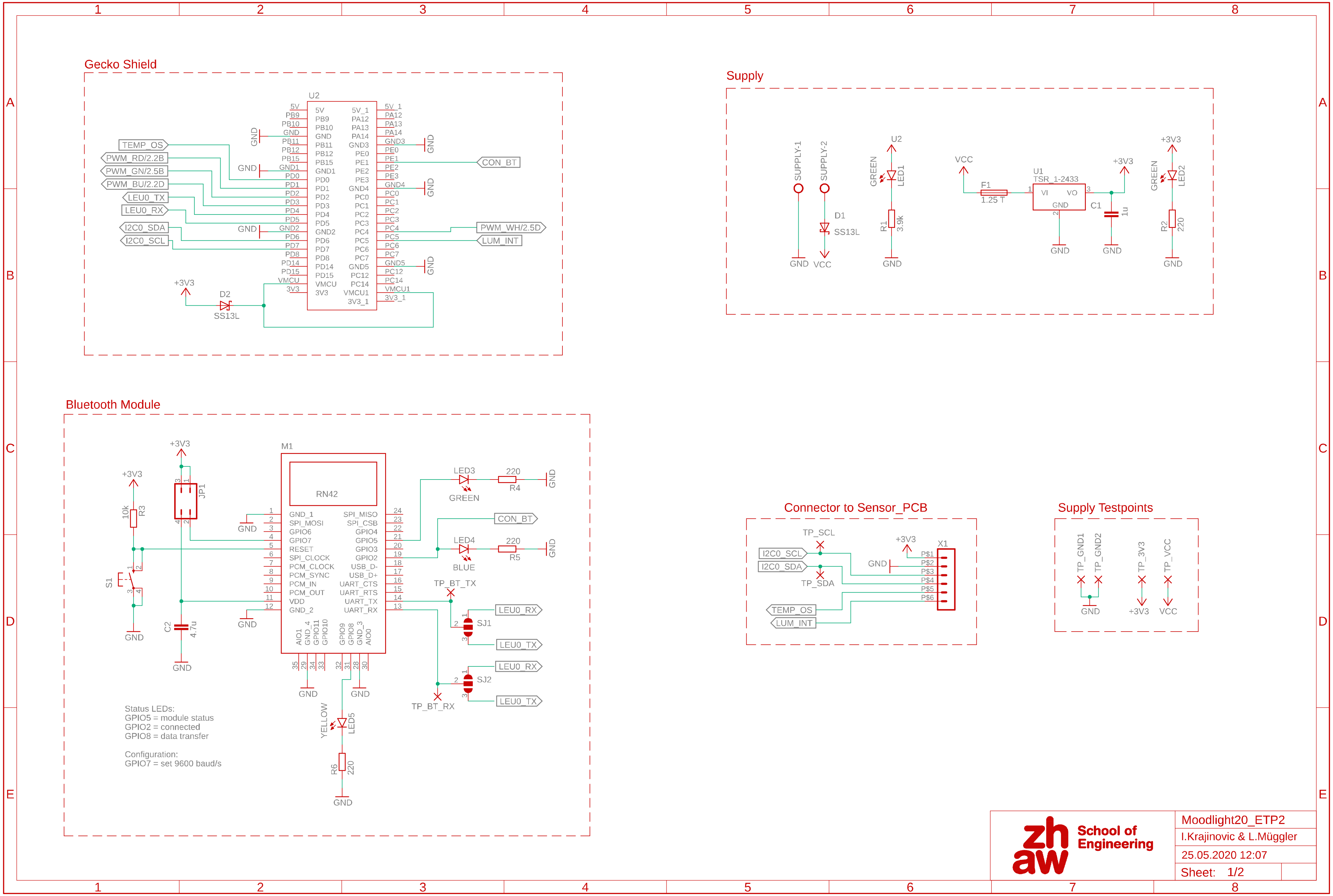
Figure 5: Temperature depending colour spectrum & range dependent step resolution

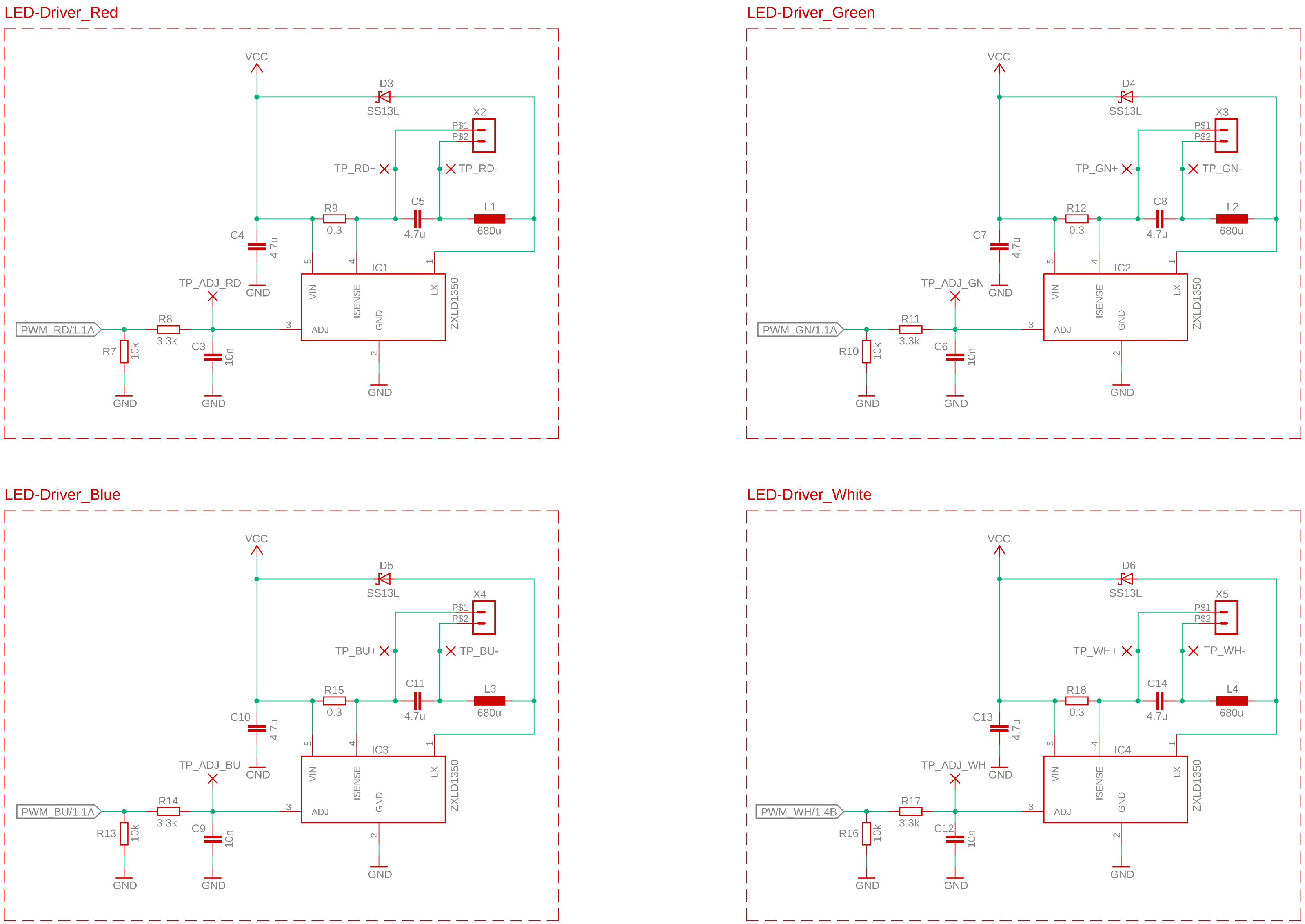
5. Test

To check the reliability of the temperature sensor, the temperature value was compared with a thermometer. Additionally the temperature sensor got heated up and cooled to check if it's reacting correctly to temperature changes. The temperature sensor had to be calibrated to eliminate offset differences. To test the temperature feature, the temperature value was simulated in the software and continuously increased by 0.125 °C. It was observed, if the colour spectrum changes correctly with the increasing temperature and how it reacts at the temperature limits. In addition, the temperature limits were changed via app, and the same procedure was performed as before to test the selectable temperature limits. The overtemperature limit has been set at 30 °C in order to be able to test this case better and more easily. Then the Moodlight was heated up to simulate overtemperature. In this case it was tested if the Moodlight switches off the LED's, if an error message appears on the smartphone and if the hysteresis is functioning properly. The Moodlight successfully passed these tests.

6. Attachments

- Schematic


Gecko Shield
U2
TEMP_OS
PWM_RD/2.2B
PWM_GN/2.5B
PWM_BU/2.2D
LEU0_TX
LEU0_RX
I2C0_SDA
I2C0_SCL
CON_BT
PWM_WH/2.5D
LUM_INT
D2
SS13L
+3V3
GND
Supply
SUPPLY-1
SUPPLY-2
D1
SS13L
GND
VCC
LED1
GREEN
R1
3.9k
U1
TSR_1-2433
F1
1.25 T
C1
1u
LED2
R2
220
+3V3
Bluetooth Module
M1
RN42
JP1
R3
10k
S1
C2
4.7u
LED3
GREEN
R4
220
LED4
BLUE
R5
220
LED5
YELLOW
R6
220
SJ1
SJ2
TP_BT_TX
TP_BT_RX
Status LEDs:
GPIO5 = module status
GPIO2 = connected
GPIO8 = data transfer
Configuration:
GPIO7 = set 9600 baud/s
Connector to Sensor_PCB
X1
TP_SCL
TP_SDA
Supply Testpoints
TP_GND1
TP_GND2
TP_3V3
TP_VCC
School of Engineering
Moodlight20_ETP2
I.Krajinovic & L.Müggler
25.05.2020 12:07
Sheet: 1/2



School of Engineering

Moodlight20_ETP2

I.Krajinovic & L.Müggler

25.05.2020 12:07

Sheet: 2/2