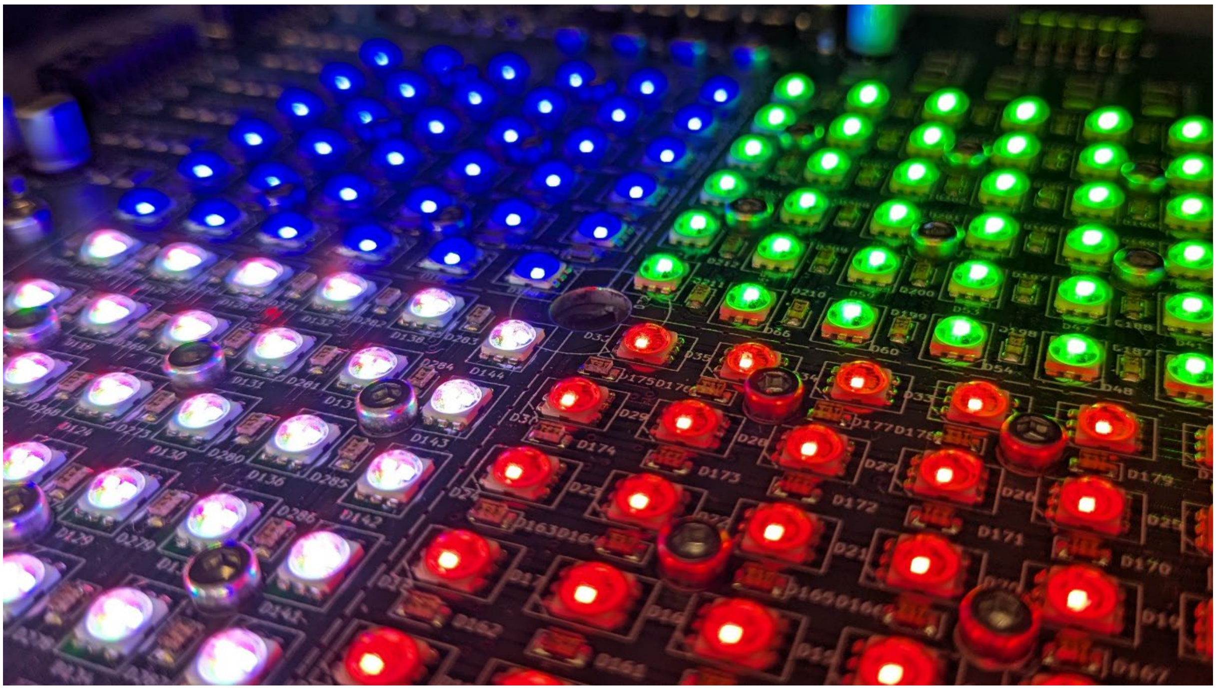
RGB LED Module
for Olfactory Assay Arena rev:C
Model 608 revC V1.0



OPERATOR'S MANUAL

Version A
May 2025

This manual refers to the **RGB LED Module for the Olfactory Assay Arena, Revision C**, as currently provided (as of April 2025, version 608-C-1) by the **Neuroscience Electronics Lab at the Institute of Zoology, University of Cologne**.

Contacts:

- mehrdad.ghanbari@uni-koeln.de (Mehrdad Ghanbari)

The following commands are intended for use with a serial terminal. It is important to ensure that the interface baud rate is set to **115200 bits per second**.

This version of the optogenetic arena uses **RGB LEDs**. Therefore, it is necessary to adapt the corresponding **Arduino scripts** to enable precise control of the LEDs. This includes modifications to the Arduino commands and configurations, as well as appropriate adjustments in **serial terminal communication tools**.

Important Information

- The Model 608 revC V1.0 and its associated parts are not intended for experiments on humans and must not be used for that purpose.Use on humans is strictly prohibited, unless specifically marked and insulated probes designed for such applications are used.This device is intended for laboratory use only.
- Before powering up the board, please ensure that the TTL/PWM Interface Board is properly connected to the RGB LED Module for the Olfactory Assay Arena, Revision C using the appropriate cables.This prevents the RGB LEDs from turning on unexpectedly without a command or signal.
- Do not touch the boards while they are powered on and under voltage.
    - This may cause damage to components or lead to unintended malfunction.

Maximum Brightness and Power Distribution

The total intensity per channel must not exceed 100 %. This means:

1. **You can run one LED color at 100 % while the others are off,**
2. **or two colors at 50 % each,**
3. **or three colors at approximately 35 % each.**

If all three colors (Red, Green, Blue) are operated at full brightness simultaneously (3 × 100 % = 300 % on one channel), the remaining three channels must not exceed a combined total of 100 % brightness.

Thus, the total output across all four channels must not exceed 400 %.

The IR LED operates independently from the RGB LEDs and can be freely set between 0 and 100 % brightness without affecting this limitation.

Installationshinweis

Installing the **Model 608 revC V1.0** in a laboratory setting is very straightforward. It can be operated on almost any standard lab bench.

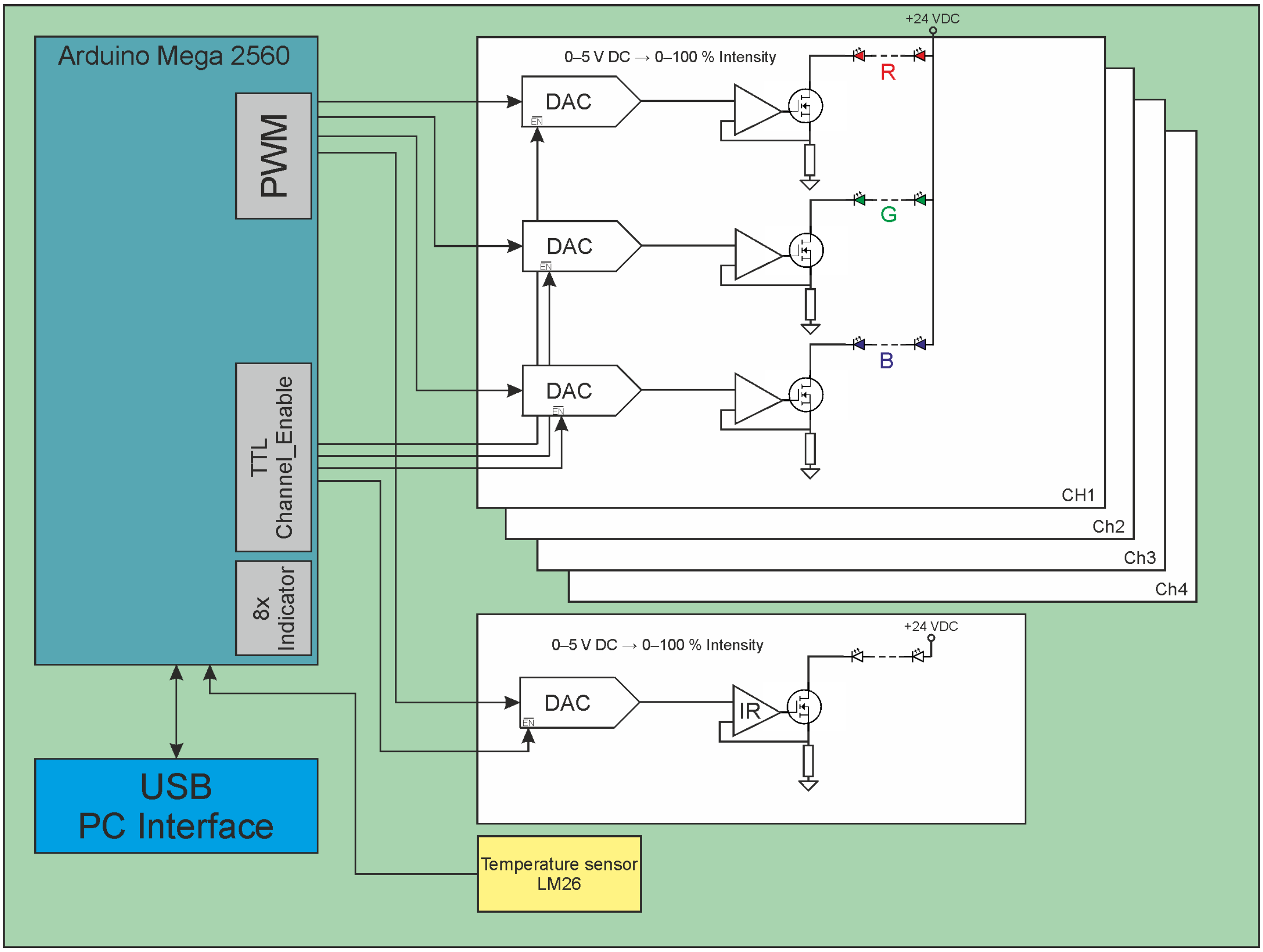
The **"RGB LED Module for the Olfactory Assay Arena, Revision C"** board features two power input jacks:

- **Ø 2.5 mm (inner)** for **24 VDC / 5 A**
- **Ø 2.1 mm (inner)** for **9 VDC / 0.6 A**

The appropriate power supplies are included.
The Arduino is powered using the included **9 VDC power supply**.

⚠ **Make sure** the power supply matches the voltage requirements of the connected boards **before plugging it in**.
⚠ **Do not block** the ventilation side of the board to prevent overheating.

Block Diagra



The following components are included in the delivery:

1. **Board 1:** RGB LED Module for the Olfactory Assay Arena, Revision C
2. **Board 2:** TTL/PWM Interface Board
3. **2x plug-in power supplies** (9 VDC, 0.66 A) – one for the Arduino, one for the 9V input of Board 1
4. **Bench power supply** (24 VDC, 5 A) – for the 24V input of Board 1
5. **RGB PCB SUPPORT**, Revision C
6. **Extruded heat sink** (150 × 160 × 50 mm)
7. **Rectangular cable sets** (for connecting the boards)
8. **4x indicator IR-LED (VSMY1850X01), 850 nm, with approx. 60 cm lead wire, with the following pin assignment:**
    1. Red (+) and Brown (–)
    2. White (+) and Black (–)
    3. Blue (+) and Green (–)
    4. Violet (+) and Grey (–)

Since the new version of the arena is equipped with **RGB LEDs**, the new board requires **multiple PWM signal outputs** compared to the previous version.

For **each channel**, there are **three PWM signals** used to control the intensity of the **red, green, and blue LEDs (RGB)**, and **three TTL signals** to **activate or deactivate** the corresponding RGB LEDs.

The **IR LEDs** receive **one shared PWM signal and one TTL signal** across **all channels**.

The following commands are used to control the **analog and digital outputs** responsible for managing the board via Arduino.
The commands are divided into **three main groups**.

Note: The **capitalization of the commands** is important and must be **strictly followed**.

Control of Analog and Digital Outputs via Arduino

The following commands allow control of the analog and digital outputs of the system, specifically for the **"RGB LED Module for the Olfactory Assay Arena, Revision C"** via Arduino.
These functions provide **flexible and precise control of light output** and digital signals for a wide range of laboratory applications.

The commands are divided into **five main categories**:

1. Changing PWM Frequency for RGB and IR LEDs

The LEDs are **not driven directly by the PWM signal**. Instead, the PWM is **filtered before reaching the LEDs**, resulting in a smoothed signal that ensures stable light output.

Changing the PWM frequency is typically **not necessary** and has **no effect on visible light behavior**. However, the option exists to **adjust the frequency** in case **interference issues arise** in specific setups or environments.

- All **RGB LEDs** share a **common frequency**.
- The **IR LED** can be controlled with a **separate frequency**.

Default settings:

- **RGB LEDs:** 15.625 Hz
- **IR LED:** 7.812 Hz

Table 1: Commands for Frequency Selection of the IR LED

Mode	frequency	**command**
1	62.5 kHz	FREQ_ir_62500
2	7.81 kHz	FREQ_ir_7812
3	976 Hz	FREQ_ir_976

Table 2: Commands for Frequency Selection of the RGB LED

Mode	frequency	**command**
1	62.5 kHz	FREQ_rgb_62500
2	31.25 kHz	FREQ_rgb_31250
3	15.625 Hz	FREQ_rgb_15625
4	7.81 kHz	FREQ_rgb_7812
5	3.906 kHz	FREQ_rgb_3906
6	1.953 kHz	FREQ_rgb_1953
7	976 Hz	FREQ_rgb_976
8	488 Hz	FREQ_rgb_488

Example:

The following command changes the PWM frequency for the IR LED to 62,500 Hz:
Sent: FREQ_ir_62500
The Arduino replies: IR LED frequency set to: 62500

For an invalid command, the Arduino replies, for example:

Sent: FREQ_rgb_6500

The Arduino replies: Ungültige RGB Frequenz

Note: Every time the frequency (for RGB or IR LED) is changed, **all duty cycle values are automatically reset to 0%**.
This is done to **avoid errors in intensity calculation** and within the Arduino script.
After changing the frequency, you must **redefine the duty cycle values for each LED**.

*As previously mentioned, **changing the frequency is usually not necessary**, since the PWM signal is filtered and does not affect the actual light output.*

2. Adjusting the PWM Duty Cycle (Light Intensity)

Although the LEDs are not driven directly by the PWM signal, but by a **filtered version**, the **duty cycle** remains the key parameter for controlling **light intensity**.

The duty cycle defines the **percentage of time the PWM signal stays HIGH** during each cycle (e.g., 50%).
After filtering, this pulsed signal becomes a **smoothed output voltage** that is **proportional to the brightness**.

- A **higher duty cycle** results in a **higher average voltage** and thus **brighter light**.
- A **lower duty cycle** produces a **lower average voltage** and dimmer light.

This enables **precise brightness control** without visible flickering.

Table 3: Command for Writing an Intensity Value for an RGB LED & IR LED

command	function
PWM_red_ch1_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the red LED of channel 1
PWM_red_ch2_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the red LED of channel 2
PWM_red_ch3_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the red LED of channel 3
PWM_red_ch4_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the red LED of channel 4
PWM_green_ch1_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the green LED of channel 1
PWM_green_ch2_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the green LED of channel 2
PWM_green_ch3_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the green LED of channel 3
PWM_green_ch4_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the green LED of channel 4
PWM_blue_ch1_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the blue LED of channel 1
PWM_ blue_ch2_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the blue LED of channel 2
PWM_ blue_ch3_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the blue LED of channel 3
PWM_ blue_ch4_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the blue LED of channel 4
PWM_ ir_led_00 … 100	Writes a value from 0–100 (% duty cycle) to adjust the IR LED

Example:
Sent: PWM_red_ch3_37
→ Sets the red LED of channel 3 to 75% intensity.

3. Activating/Deactivating Individual LEDs

- Each **red, green, and blue LED per channel** can be **controlled individually** (on/off).
- **IR LEDs** are **switched collectively** for all channels.

The third command group is used to **enable or disable** the RGB LEDs and the IR LED.
With the corresponding commands, individual RGB LEDs and the IR LED can be **turned on or off**.
When enabled, the LED operates using the **previously set PWM value**.

Table 4: command to enable or disable the RGB LEDs and the IR LED

command	function
EN_red_ch1_1	Enable LED red Chanel 1
EN_red_ch1_0	Disable LED red Chanel 1
EN_red_ch1_1	Enable LED red Chanel 2
EN_red_ch1_0	Disable LED red Chanel 2
EN_red_ch1_1	Enable LED red Chanel 3
EN_red_ch1_0	Disable LED red Chanel 3
EN_red_ch1_1	Enable LED red Chanel 4
EN_red_ch1_0	Disable LED red Chanel 4
EN_ green _ch1_1	Enable LED green Chanel 1
EN_ green _ch1_0	Disable LED green Chanel 1
EN_ green _ch1_1	Enable LED green Chanel 2
EN_ green _ch1_0	Disable LED green Chanel 2
EN_ green _ch1_1	Enable LED green Chanel 3
EN_ green _ch1_0	Disable LED green Chanel 3
EN_ green _ch1_1	Enable LED green Chanel 4
EN_ green _ch1_0	Disable LED green Chanel 4
EN_blue_ch1_1	Enable LED blue Chanel 1
EN_ blue _ch1_0	Disable LED blue Chanel 1
EN_ blue _ch1_1	Enable LED blue Chanel 2
EN_ blue _ch1_0	Disable LED blue Chanel 2
EN_ blue _ch1_1	Enable LED blue Chanel 3
EN_ blue _ch1_0	Disable LED blue Chanel 3
EN_ blue _ch1_1	Enable LED blue Chanel 4
EN_ blue _ch1_0	Disable LED blue Chanel 4
EN_ir_led_1	Enable LED red Chanel IR
EN_ir_led_2	Disable LED red Chanel IR

4. Control of the 8 Digital Outputs

The board features a total of eight digital outputs that can be used flexibly as status indicators, control lines, or trigger signals.

- The outputs are divided into two groups: **Group A** and **Group B**, each containing four outputs.
- Each output can be individually set to **HIGH** or **LOW**.
- The outputs are visible directly on the board and can be controlled via software.
- Although labeled as “indicators” on the board, they can be used for various other purposes as well.

This setup simplifies seamless integration of the board with other devices and systems in the laboratory environment.

The fourth command group is used to activate or deactivate the eight digital outputs.

Using the corresponding commands, each individual digital output can be selectively switched on or off.

Table 5: command to enable or disable Digital Outputs

command	function
IND_1_1	Enable indicator 1 (Ind A-1)
IND_1_0	Disable indicator 1 (Ind A-1)
IND_2_1	Enable indicator 2 (Ind B-1)
IND_2_0	Disable indicator 2 (Ind B-1)
IND_3_1	Enable indicator 3 (Ind A-2)
IND_3_0	Disable indicator 3 (Ind A-2)
IND_4_1	Enable indicator 4 (Ind B-2)
IND_4_0	Disable indicator 4 (Ind B-2)
IND_5_1	Enable indicator 5 (Ind A-3)
IND_5_0	Disable indicator 5 (Ind A-3)
IND_6_1	Enable indicator 6 (Ind B-3)
IND_6_0	Disable indicator 6 (Ind B-3)
IND_7_1	Enable indicator 7 (Ind A-4)
IND_7_0	Disable indicator 7 (Ind A-4)
IND_8_1	Enable indicator 8 (Ind B-4)
IND_8_0	Disable indicator 8 (Ind B-4)

5. Reading the board temperature:

The built-in temperature sensor is connected to analog pin A0 on the Arduino. You can read the current board temperature using the following command:

Sent: TEMP

When this command is sent to the Arduino via the serial port, the Arduino responds with the current temperature in degrees Celsius.

Note:

The temperature sensor has no other function besides measuring temperature. However, the measured value can be used in the software – for example, to automatically turn off the LED at a certain temperature. This helps reduce the risk of damaging the board due to overheating.

PWM Outputs and Control Signals of the Arduino Mega 2560 for Multi-Channel LED Driver Systems

The following list outlines the pin configuration of the Arduino Mega 2560 used in a multi-channel LED and infrared control system. Various PWM-capable pins are assigned to drive infrared LEDs and RGB LED channels. Additional pins are designated to enable individual color channels. Furthermore, several digital outputs are reserved for controlling four indicator units (IND_1A to IND_4B). Each pin's function, including its PWM capability and associated port, is listed in detail below.

Table 6: Pinout Arduino to Board

Pin Arduino	Pin Register	Timmer	Function
4	PG5	OC0B	ir_led
13	PB7	OC1C	red_ch1
12	PB6	OC1B	green_ch1
11	PB5	OC1A	blue_ch1
3	PE5	OC3C	red_ch2
2	PE4	OC3B	green_ch2
5	PE3	OC3A	blue_ch2
8	PH5	OC4C	red_ch3
7	PH4	OC4B	green_ch3
6	PH3	OC4A	blue_ch3
44	PL5	OC5C	red_ch4
45	PL4	OC5B	green_ch4
46	PL3	OC5A	blue_ch4
30	PC7		Enable red_ch1
31	PC6		Enable green_ch1
32	PC5		Enable blue_ch1
33	PC4		Enable red_ch2
34	PC3		Enable green_ch2
35	PC2		Enable blue_ch2
36	PC1		Enable red_ch3
37	PC0		Enable green_ch3

38	PD7		Enable blue_ch3
39	PG2		Enable red_ch4
40	PG1		Enable green_ch4
41	PG0		Enable blue_ch4
42	PL7		Enable IR_LED
22	PA0		IND_1A
23	PA1		IND_1B
24	PA2		IND_2A
25	PA3		IND_2B
26	PA4		IND_3A
27	PA5		IND_3B
28	PA6		IND_4A
29	PA7		IND_4B
A0			Temperature sensor

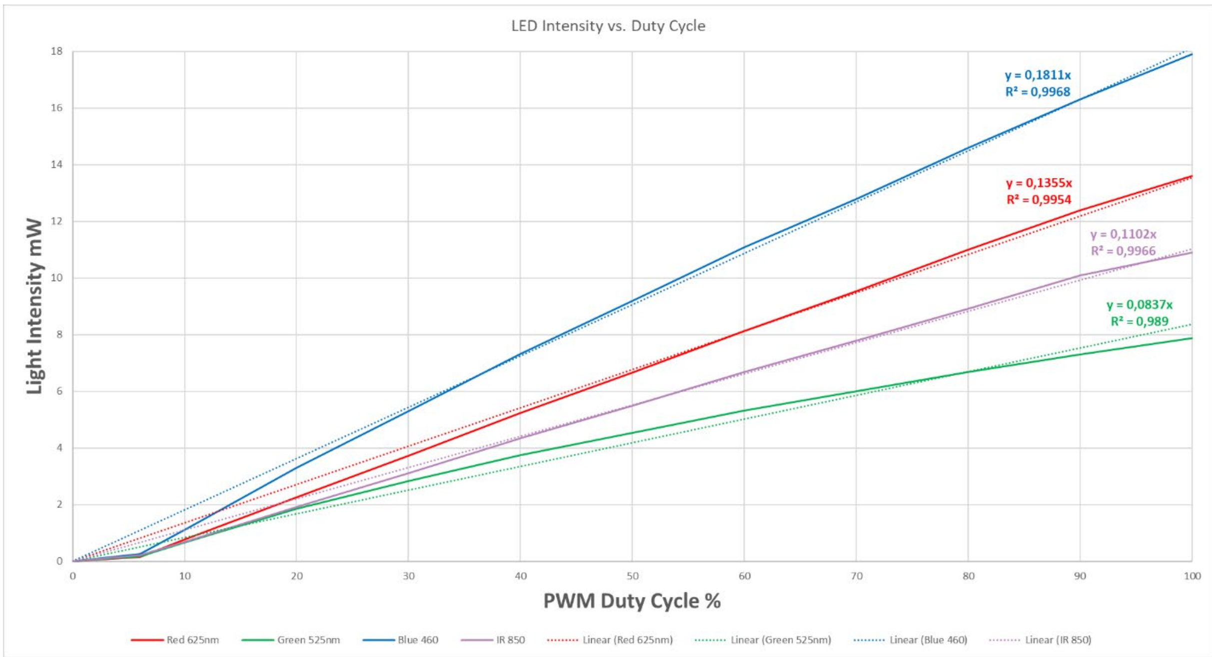
Explanation of the Measurement

The LED intensity was measured using a **photodiode detector of type PH100-Si**, which features a **10 mm aperture**. The detector was positioned on a flat surface (measurement arena) at a fixed **distance of 10 cm from the LED**, ensuring consistent optical coupling. The power calibration of the sensor has an **uncertainty of approximately ±2%**.

The measurement was performed with the **board in a cold state**, i.e., before any significant self-heating due to prolonged operation occurred. It should be noted that **rising temperatures can affect the measured values**, as LED output is temperature-dependent.

Table 7: LED Intensity vs. Duty Cycle

	Light Intensity (mW)			
Duty Cycle (%)	Red 625nm	Green 525nm	Blue 460nm	IR 850nm
0	0	0	0	0
6	0,15	0,174	0,266	0,217
10	0,78	0,67	1,12	0,68
20	2,27	1,85	3,3	1,92
30	3,73	2,83	5,31	3,12
40	5,24	3,75	7,33	4,34
50	6,66	4,54	9,19	5,49
60	8,14	5,32	11,1	6,68
70	9,53	6	12,8	7,8
80	11	6,68	14,6	8,93
90	12,4	7,31	16,3	10,1
100	13,6	7,88	17,9	10,9



In addition to intensity measurements, the **emitted wavelengths of the LEDs** were also checked. The measured wavelengths showed slight deviations from the nominal values provided by the manufacturer, as shown in the following table:

LED	Nominal Wavelength	Measured Wavelength
Red	625 nm	630 nm
Green	525 nm	527 nm
Blue	460 nm	458 nm
IR	850 nm	858 nm

These deviations can result from manufacturing tolerances in the LEDs as well as the spectral response characteristics of the detector.

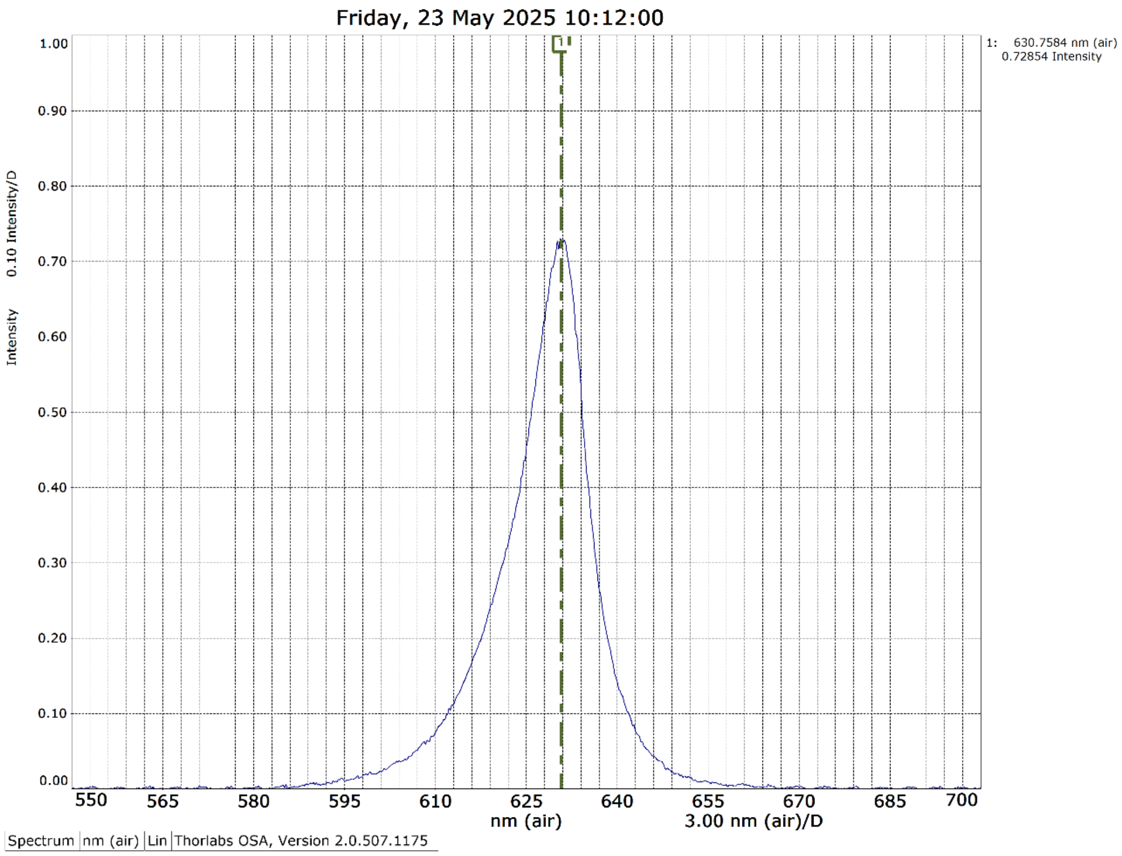


Figure 1: Red LED

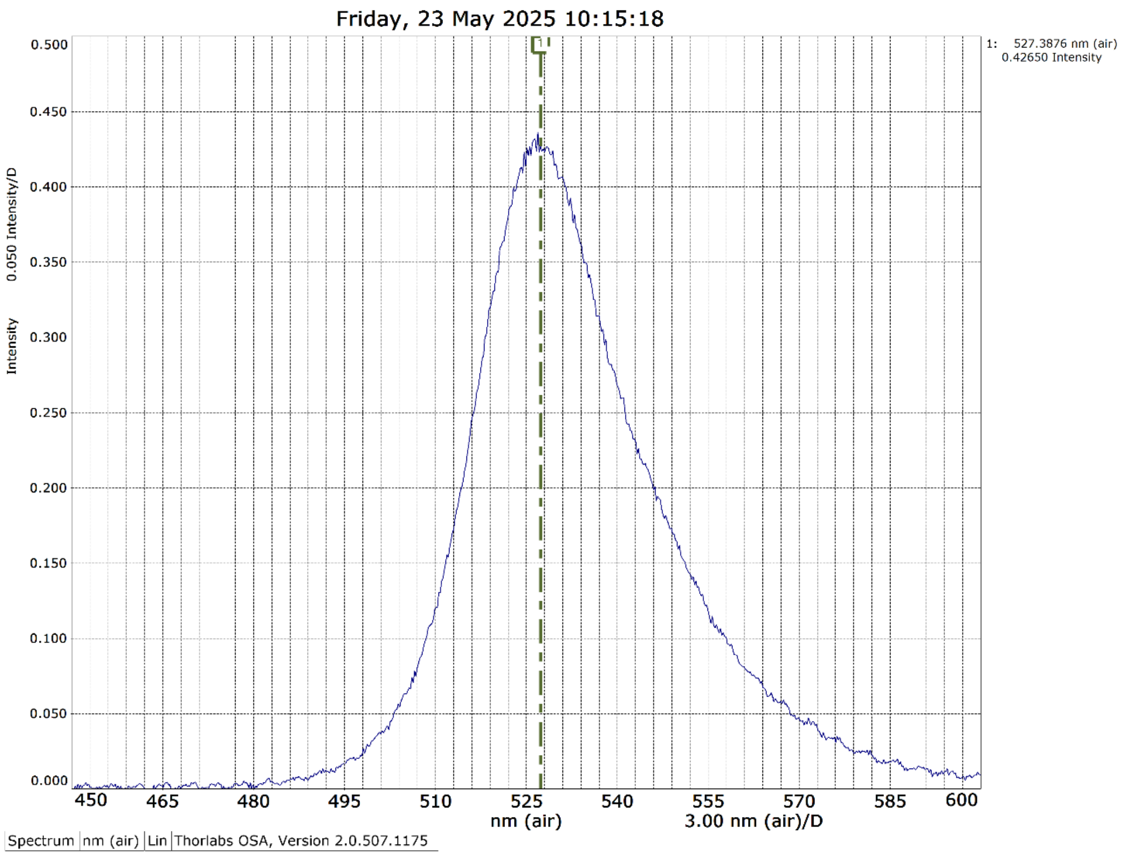


Figure 2: Green LED

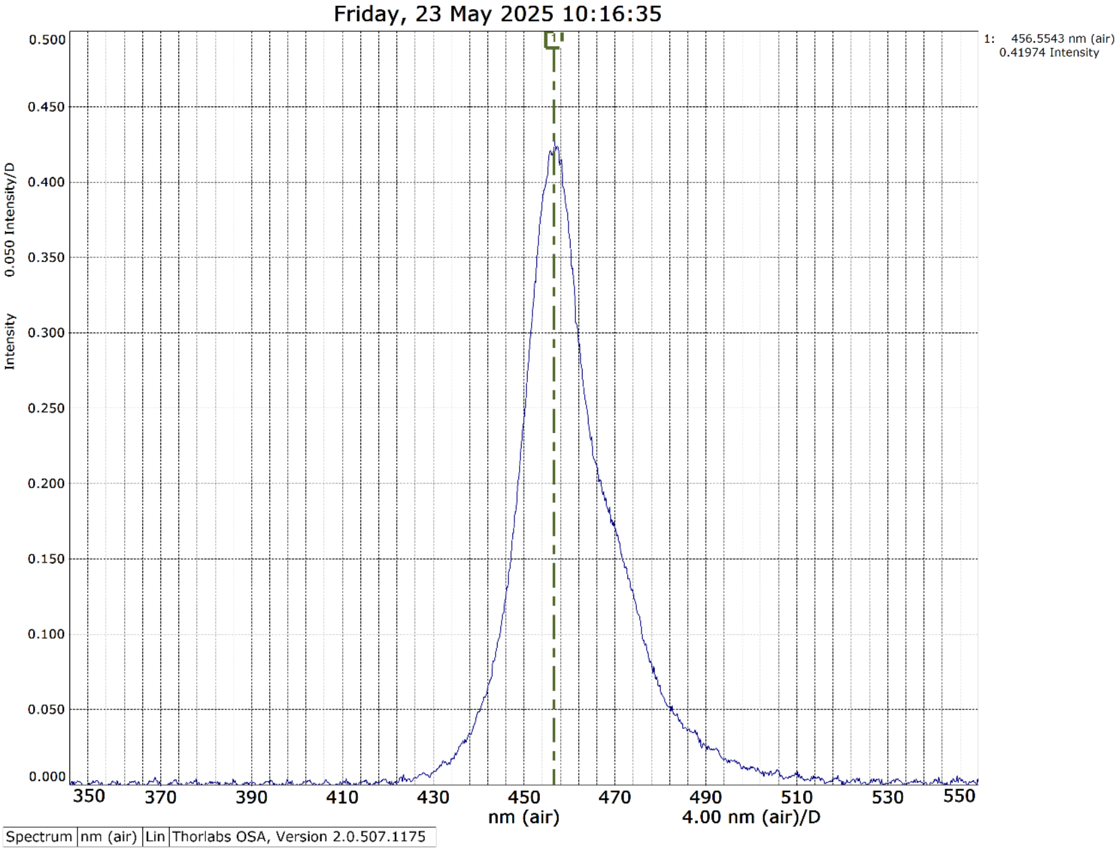


Figure 3: Blue LED

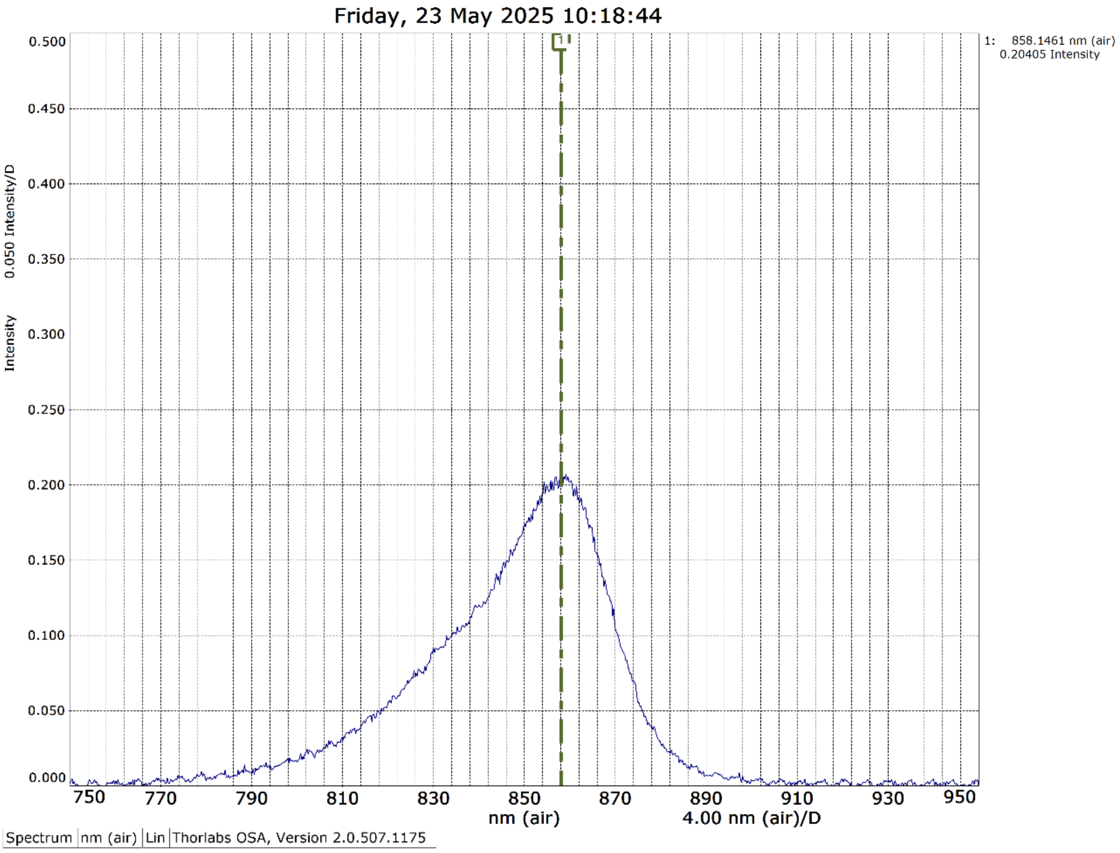


Figure 4: IR LED