Preliminary research – ESE

By Emiel Visser and Joris Bol

HAN RGBbaby

DOCUMENT ORGANIZATIONAL DETAILS:

Document Type:	
Skills Consultant:	Paul van Wegen
Project Client:	Johan Korten
Education:	Embedded Systems Engineering
Institute:	School of Engineering and Automotive, Hogeschool Arnhem-Nijmegen
Report Date:	18-12-2023
Report Version:	v1.0
Project Duration:	
Education Term:	
Students:	2108100 (Joris Bol), 2111740 (Emiel Visser)

Version management

Version:	**Date:**	**Change:**	**Name:**
V1.0	5-9-2023	Creation document and C1.	J. Bol
V1.0	26-9-2023	Addition of C4 and improvement of C2 and C3.	E. Visser & J. Bol
V1.0	2-10-2023	Spellcheck and added extra info.	E. Visser
V1.0	9-12-2023	Added info.	E. Visser
V1.0	10-12-2023	Added info.	E. Visser
V1.0	12-12-2023	Finished chapter 5	J. Bol
V1.0	13-12-2023	Added decision making matrixes.	E. Visser
V1.0	14-12-2023	Added more decision making matrixes.	E. Visser
V1.0	17-12-2023	Added component explanations.	E. Visser
V1.0	18-12-2023	Added conclution	J. Bol

Table of Contents

1 – Objective and requirements

1.1 – Objective

This document is meant as research for the selection of components for the final product. The form factor and other product specific attributes are not discussed here. The main topics of this sub research are the selection of LED or other light emitting elements and touch sensors.

We will work as follows: First and foremost we need to select a sizeable portion of the components according to the specifications made in the following subsections in this chapter. Secondly, a cost benefit analyse will be made, taking into account the cost, component size, ease of use and feature set of the component. The component that comes out on top will be selected. If the client decides that the aforementioned component is not sufficient, the next best option will be presented.

A physical test will be conducted later in this project, if for some reason the selected component doesn't function according to our expectations, this document will not be updated. The final component shall be mentioned in the product report.

1.2 – Requirements light sources

As previously mentioned, the LEDs or other forms of illumination must meet certain requirements.

1. It must be as flat as possible.
2. It must be as cheap as possible.
3. It must be as simple as possible (functionality mustn't be affected).
4. It must be as energy efficient as possible.
5. It must be a SMD (Surface Mount Device) component.
6. It must be capable of at least RGB otherwise multiple different sources must be used.
7. It mustn't generate too much heat.

1.3 – Requirements touch sensors

As previously mentioned, the touch sensors must meet certain requirements.

1. It must be as flat as possible.
2. It must be as cheap as possible.
3. It must be as simple as possible (functionality mustn't be affected).
4. It must be as energy efficient as possible.
5. It must be a SMD (Surface Mount Device) component.
6. It must be able to sense pressure through the silicone.
7. Ideally it disconnects when pressed.

2 – Light sources

Type:	Description:
Link:	19-337/R6GHBHC-A01/2T
Disadvantages:	Large surface area
Advantages:	Small, inexpensive
Price:	€0,13

Type:	Description:
Link:	WUE 150141M1731
Disadvantages:	Quite tall, expensive.
Advantages:	Bright, small surface area.
Price:	€0,46

Type:	Description:
Link:	KAA-3528RGBS-11
Disadvantages:	Quite tall, expensive.
Advantages:	Bright, small surface area.
Price:	€1,05

Type:	Description:
Link:	RND 135-00252
Disadvantages:	Quite tall, only available in large quantities.
Advantages:	Bright, small surface area, cheap.
Price:	€0,04

Type:	Description:
Link:	ILPL-K501-RGBW-SK105-01
Disadvantages:	Large surface area, expensive.
Advantages:	Bright, very thin.
Price:	€1,24

Type:	**Description:**
Link:	SMD-LX0707RGB-TR
Disadvantages:	Relatively expensive.
Advantages:	Very small, doesn't use much IO, low current consumption.
Price:	€0,66

Type:	**Description:**
Link:	SMLP34RGBN1W3
Disadvantages:	Expensive.
Advantages:	Very thin, very small, doesn't use much IO.
Price:	€1,26

Type:	**Description:**
Link:	IN-P55QDTRGBW
Disadvantages:	Large surface area, relatively expensive.
Advantages:	Thin, bright.
Price:	€0,65

Type:	**Description:**
Link:	CLQ6B-TKW-S1L1R1H1TBB7935CC3
Disadvantages:	Large surface area, relatively expensive.
Advantages:	Comes with white element, thin.
Price:	€0,85

Type:	**Description:**
Link:	SK6812 (3535)
Disadvantages:	Still relatively large.
Advantages:	Doesn't use much IO, low current consumption.
Price:	€0,49

Type:	**Description:**
Link:	WS2812b (5050)
Disadvantages:	Large surface area.
Advantages:	Doesn't use much IO, low current consumption.
Price:	€0,49

3 – Touch sensors

Type:	Description:
Link:	TL3342F260QG
Disadvantages:	Relatively expensive, small actuation zone.
Advantages:	Thin, relatively high actuation force.
Price:	€0,61

Type:	Description:
Link:	TS17-48-05-YE-260-SMT-TR
Disadvantages:	-
Advantages:	Super thin, large actuation zone, cheap, relatively high actuation force.
Price:	€0,26

Type:	Description:
Link:	MPR121QR2
Disadvantages:	Discontinued, relatively difficult to integrate.
Advantages:	Small, relatively easy to control, lots of channels, stable.
Price:	?

Type:	Description:
Link:	AT42QT2120
Disadvantages:	Relatively difficult to integrate.
Advantages:	Small, relatively easy to control, lots of channels, stable.
Price:	€1,59

4 – Preselection

We have seen that there are a lot of different LEDs and sensors that we could use. To see which ones *could* be the best for our use case we make a preselection by putting the aforementioned parts in decision making matrixes.

4.1 – Light sources

For the non-addressable led strip we choose the 19-337C_RSBHGHC-A01_2T and the WUE_150141M173100 LEDs in combination with standard R, G and B 0603 LEDs. We decide to go with these LEDs because they were either relatively cheap and available or very small.

Type:	Size:	Brightness:	Price:	Controllability single led:	Controllability multiple leds:		Score:
19-337/R6GHBHC-A01/2T	6	3	3	5	2		19
WUE 150141M1731	1	3	3	5	2		14
KAA-3528RGBS-11	1	3	3	5	2		14
RND 135-00252	1	4	3	5	2		15
ILPL-K501-RGBW-SK105-01	2	5	1	5	2		15
SMD-LX0707RGB-TR	6	4	2	3	4		19
SMLP34RGBN1W3	1	3	1	5	2		12
IN-P55QDTRGBW	1	5	3	5	2		16
CLQ6B-TKW-S1L1R1H1TBB7935CC3	1	6	2	5	2		16
SK6812 (3535)	3	4	3	3	4		17
WS2812b (5050)	2	5	3	3	4		17

Size	Higher is better
Brightness	Higher is better
Price	Higher is better
Controllability single led	Higher is better

4.2 – Touch sensors

Because we wanted to have a good balance between buttons and capacitive touch sensors we chose to put all but the MPR12QR2 IC on our PCBs because this IC has been deprecated since 2015. An alternative is the AT42QT2120. We have tested the TL3342F260QG and TS17-48-05-YE-260-SMT-TR buttons on our test flex PCB and the AT42QT2120 on our final flex PCB. We decided to split up the physical buttons and capacitive buttons because we wanted to do more research on capacitive touch sensors. But we couldn't wait any longer with the ordering of the PCBs.

Type:	Size:	Actuation force:	Price:	Controllability:		Score:
TL3342F260QG	3	2	4	4		13
TS17-48-05-YE-260-SMT-TR	4	1	5	4		14
MPR121QR2	5	5	4	3		17
AT42QT2120	5	5	2	3		15

Size	Higher is better
Actuation force	Higher is better
Price	Higher is better
Controllability	Higher is better

4.3 – Other components

To drive the LEDs (addressable and non-addressable) we needed some extra hardware to drive them properly. In this section you will find which ones we choose and why.

Function:	Type:	Size:	Feature set:	Solderability:	Footprint layout:	Price:	Controllability:		Score:
IO multiplexer	PCAL9722HNMP	4	6	2	4	4	3		23
IO multiplexer	PCA8574D	1	4	5	6	5	6		27
IO multiplexer	PCA8575BS118	5	6	3	4	3	6		27
IO multiplexer	PCA9548ABS118	5	3	3	5	1	6		23
IO multiplexer	PCA9670BS118	6	4	4	5	5	3		27
IO multiplexer	PCA9536DR	2	2	5	5	6	6		26
PWM driver	TLC5940RHBR	4	5	4	4	4	6		27
PWM driver	TLC5947RHBR	4	6	4	6	0	6		26
PWM driver	TLC59401RHBR	4	5	4	5	3	5		26
PWM driver	PCA9685BS118	5	5	4	4	2	6		26
PWM driver	TLC59711PWPR	1	4	5	5	1	6		22
Flash	25LP064A-JLLE	4	6	4	3	2	6		25
Flash	IS25WQ040-JNLE	2	2	5	3	6	6		24
Flash	IS25LP032D-JLLE	4	4	4	3	3	6		24
Flash	IS25WP032D-JKLE	4	5	4	3	4	6		26
Flash	IS25LP080DJULETR	5	3	3	3	5	6		25
Flash	IS25LP064D-JBLA3	2	6	5	3	1	6		23
Flash	IS25LP064D-JKLE	4	6	4	3	2	6		25
Flash	IS25LP040EJYLETR	5	3	3	3	5	6		25
Flash	AT25QL641-UUE-T	6	6	1	4	2	6		25
Flash	S25FL116K	2	3	5	3	-	6		19
Microcontroller	ATSAMD21J18A-AU	3	6	5	4	3	2		23
Microcontroller	ATSAMD21J18A-MU	5	6	3	4	4	2		24
Microcontroller	ATSAMD21J18A-CU	6	6	0	2	2	2		18
Microcontroller	ATSAMD21G18A-AU	4	4	5	4	4	5		26
Microcontroller	ATSAMD21G18A-MU	6	4	3	4	5	5		27
Buffer (as levelshifter)	SN74HCT125DR	2	5	5	4	5	6		27
Buffer (as levelshifter)	SN74HCT125RGYR	5	5	4	4	4	6		28

Size	Higher is better
Feature set	Higher is better
Solderability	Higher is better
Footprint layout	Higher is better
Price	Higher is better
Controlability	Higher is better

4.3.1 – IO multiplexer

To detect when a user has pressed the skin we wanted to add buttons or other pressure sensitive devices on the flexible PCBs. To register these button presses we decided to use I2C enabled IO multiplexer.

We ended up going with the PCA9670BS because the package is very small, it had all the features we wanted, It was relatively easy to solder and the pinout was very convenient to route out to the pins on to the connector. We also found that the price was pretty low in comparison with to the other multiplexers we selected. However there was one issue we did not pay attention to: It doesn't have a library.

4.3.2 – PWM led driver

To actually drive the non-addressable LEDs we went with a specialized PWM led driver. We settled on the TLC59401RHBR led driver. This driver is small, it has a lot of features, it's pretty easy to solder, it's easy to route and it's relatively cheap.

There is, however one small problem which we found out later. This chip also doesn't have a premade library! This is why the TLC5940RHBR scores higher. Note: We have seen people use the TLC5940 library in combination with the TLC59401RHBR. We haven't been able to verify this though.

4.3.3 – Flash memory

Because we're using neopixels we're going to need more flash than usual. The microcontroller we're using has 256k of memory though so it is probably enough. But we wanted to have the option to have more if required.

We ended up choosing the IS25LP032D-JLLE because this chip is easy to solder and has enough storage. If you find the storage size inadequate then you can replace this chip with a different chip because the pinout is more or less standardized. Note: Check the footprint very carefully because the sizes might differ. E.g. the footprint on the PCB is WSON but there are two WSON sizes 6x5mm and 8x6mm.

4.3.4 – Microcontroller

To control the LEDs and communicate with the chips and other modules in the baby doll we will need a microcontroller.

For the test PCB we ended up going with the ATSAMD21J18A-MU because the this chip has a lot of IO but the overall footprint is relatively small.

However there isn't really a good Arduino board file available for this chip. This means you will either have to use the Atmel Xplained Xpro bootloader (quite limited) or the Adafruit feather M0 express bootloader (not made for this chip, but will work. Extra IO can't be used). You can also make a custom Arduino board file but this is quite difficult. We tried using embedded C at first but this proved too time consuming. However it might be a good option for you.

4.3.5 – Level shifter

To translate the 3.3V output signal to a voltage the neopixel LEDs require we need to use a level shifter. For our development board we went with the SN74HCT125DR. This chip was used in an Adafruit development board which we found adequate for our purpose.

The only downside of this chip is that it's quite big. This is why we went with the SN74HCT125RGYR for the ledDriverBoardV3. Same chip, different footprint.

5 – Selection tests

To accurately select the correct light sources and touch sensors we decided that we are going to make a test set up consisting of a test board with all the possible functionalities and flexible test strips with the LEDs and buttons.

5.1 – Driver board

The driver board has 3 20-pin FCC connectors of which the middle eight pins are capable of input. There is a 4th FCC connector specifically for addressable LEDs.

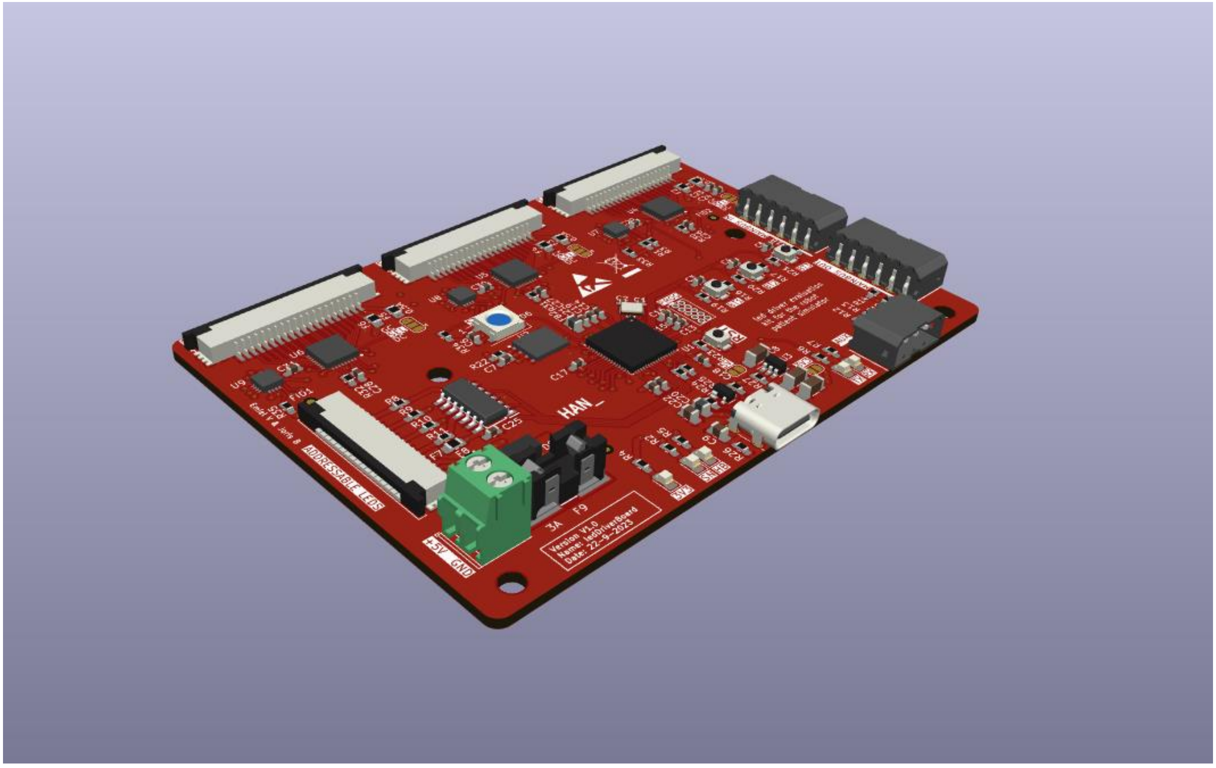

Figure 1 LedDriverBoard

This board makes use of the SAMD21J18A as microcontroller and in combination with led drivers and IO expanders the board gains a lot of functionality:

Digital interfacing		
Capability:	**Input/output:**	**Use:**
3x 12-pin PWM	Output	Used to drive the normal LEDs.
3x 8-pin GPIO	Input/output	Used to drive the normal LEDs and touch sense.
4x channels for addressable LEDs.	Output	Used to drive the addressable LEDs.
debug	Input/output	Used to debug and flash the board.
Main bus in	Input	Used as the input from the mainboard.
Main bus out	Output	Passthrough.
Auxiliary	Input/output	Possibility for an extra I2C sensor.

Led functions	
Colour:	**State:**
RGB led: Off	System is functioning normally.
RGB led: White	The PCB went into hard fault.
RGB led: Red	There is no connection between the PCB and the mainboard.
RGB led: Green	INIT finished successfully.
RGB led: Blue	The PCB is in test mode.
Heartbeat led	blinks indicating that the board is still alive.
Sens led:	Turns on when a press is detected.
3V3 led:	Is on when there is power.
Serial leds:	Connected to the TX and RX pins.

Button functions	
Button:	**Function:**
RST	Resets the microcontroller.
BT1	Can be mapped to any function.
BT2	Can be mapped to any function.
BT3	Can be mapped to any function.

5.2 – FlexLeds

The flexleds board are flexible PCBs that connect to the led driver. The two test strips that are made have the following components:

FlexLeds	
Type:	**Link:**
Button	TL3342F260QG
Button	TS17-48-05-YE-260-SMT-TR
Led	SMLD12EN1WT86
Led	19-337/R6GHBHC-A01/2T
Led	WUE 150141M1731

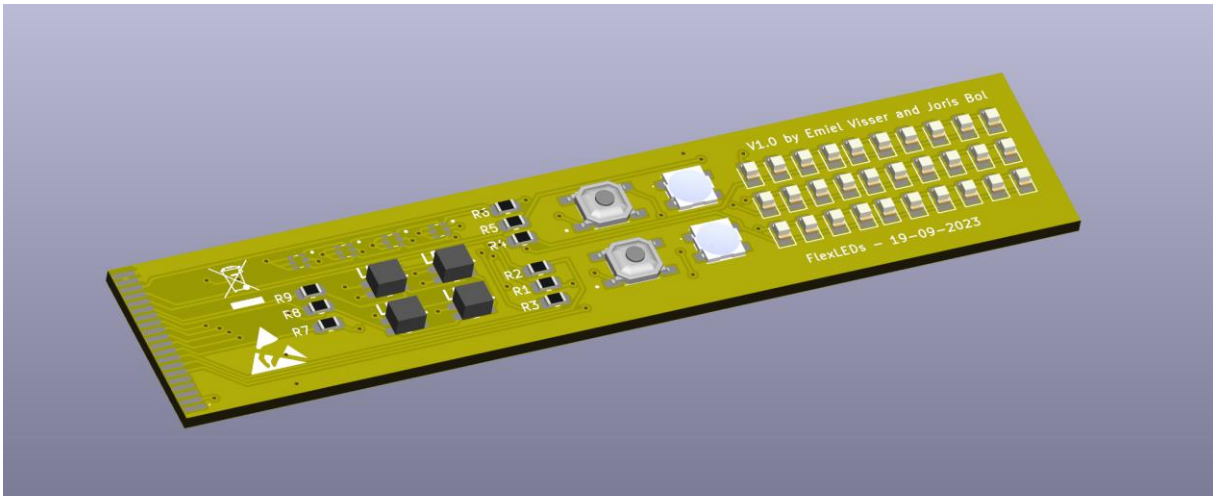

Figure 2 flexLeds

flexAddressLeds	
Type:	**Link:**
Led	[SMD-LX0707RGB-TR]
Led	[SMLP34RGBN1W3]
Led	[SK6812 (3535)]

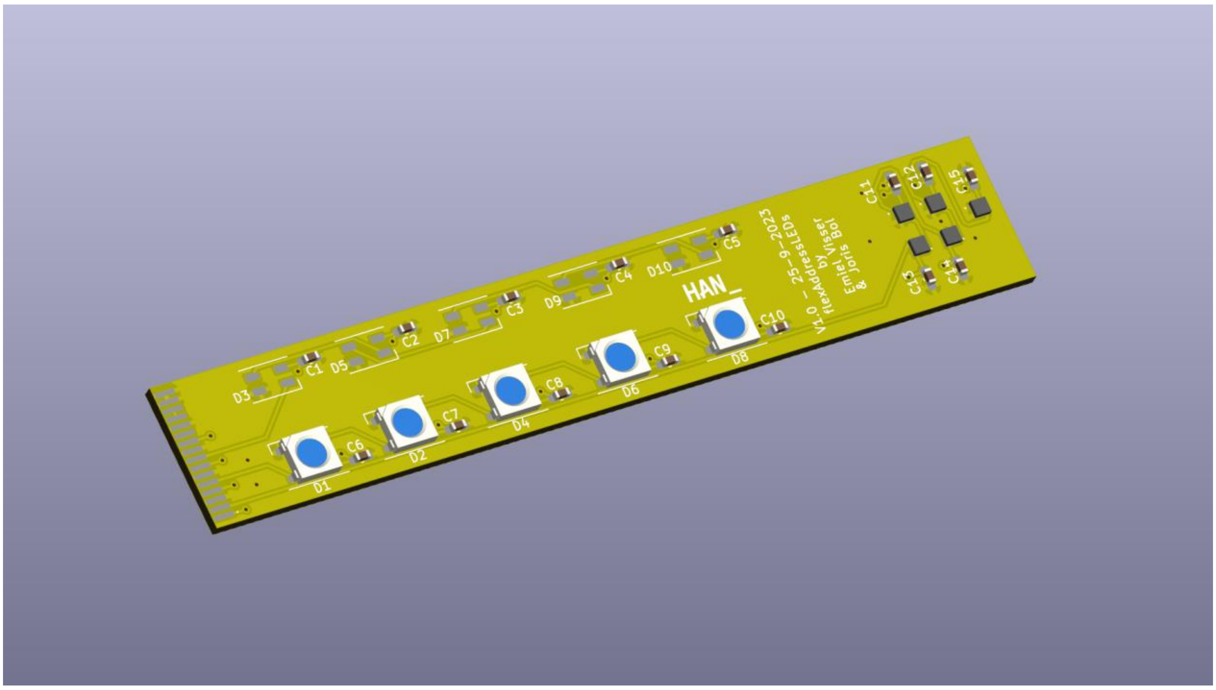

Figure 3 flexAddressLeds

5.4 – Test method

Accurate tests must be done objectively. A way to do this in our case is to measure the light intensity and the light spectrum. This can be done by using a combination of a spectrometer and a lux measuring device.

The way we intent to use these devices is as follows:

The test bench will consist of four components: the sample housing, the sensor housing, the light source, and the medium being tested. These four objects will be layered as illustrated in the figure below.

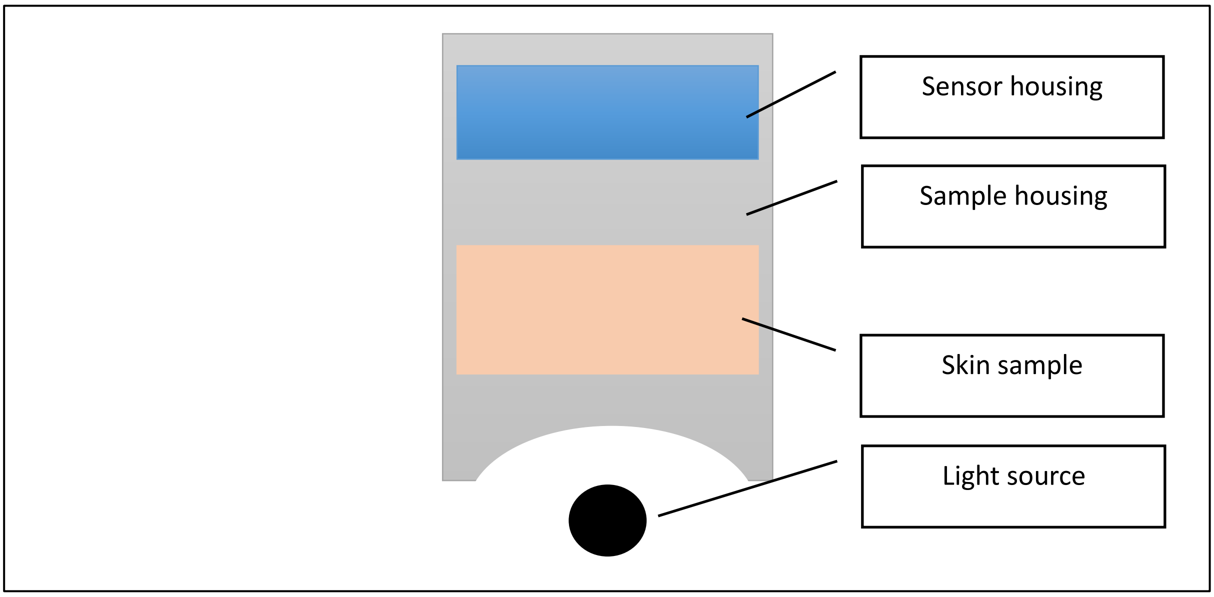

Figure 4 schematic view testbench

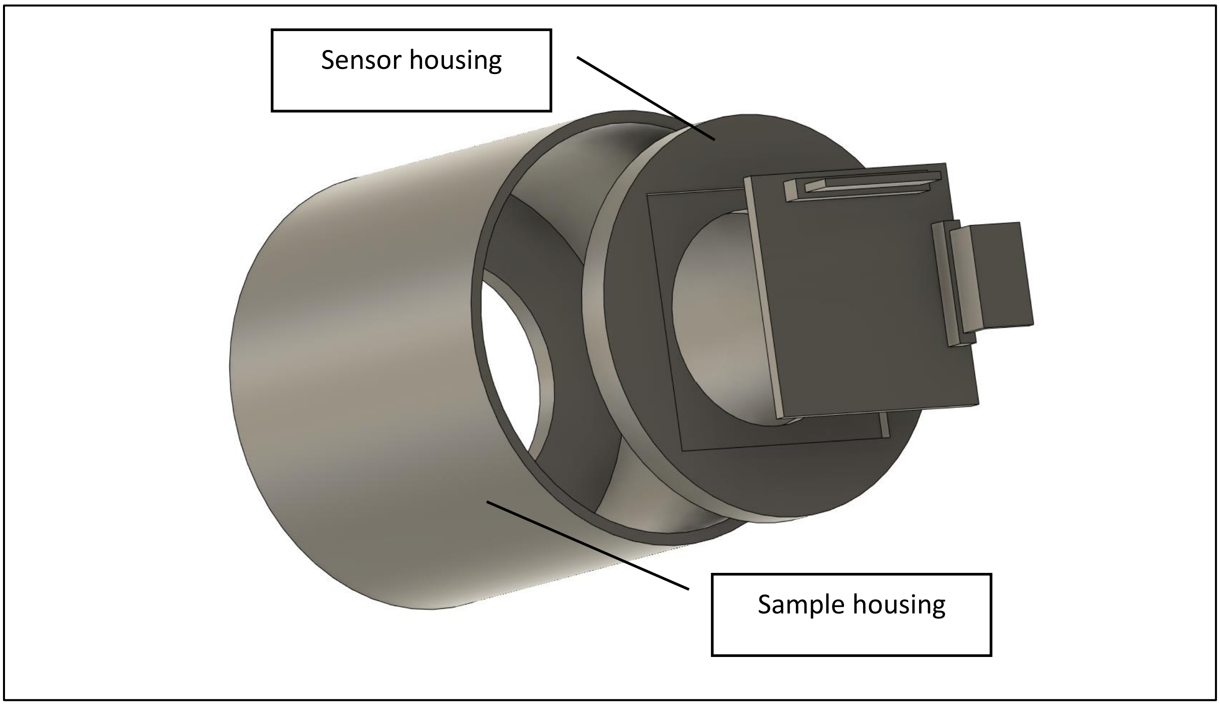

Figure 5 3D view testbench

The reason that we need to use this enclosure is to insure that outside light is not a factor in our tests.

The components that will be used are:

- PiCamera (provided)
- spectrometer (provided)
- RICOH HF16HL-1b (provided by school)
- raspberry pi 4 or higher

For each test, the test will be conducted as follows:

1. A new sample will be put in the housing. These samples are labelled: A1, C2, etc. the samples are also labelled based on the time of production.
2. The sensor housing with the spectrometer will be lowered onto the sample.
3. The sensor and sample housing will be put on a light source.
4. The sensor value will be noted along with the sample id.
5. The sensor housing with the LUX measuring devise will be lowered on the sample.
6. Step 3 and 4 will be repeated.
7. The sample is removed from the housing.

The combination of testing and consulting will get us an objective answer on the question what colour the skin must be and what colours it must turn.

5.5 – Tests realisation

To conduct the test we need to first accomplish the following steps:

1. Assemble the test board.
2. Assemble the flexible led strips.
3. Assemble the spectrometer.
4. Have working code for the spectrometer.
5. Have working code for the test board.

Each of the aforementioned points had a lot of complications and compromises, the changes that we made are all discussed in the next subchapters. At the end of this chapter, a new test plan will be established and the coinciding tests will be carried out.

5.4.1 – Assembling the test board

During the assembly end physical testing of the board, three main problems came up: the flash chip (did not line up with its footprint), the diode (that protects the board from reverse plugins) and the crystal.

- The flash chip did not correctly line up with the footprint we had chosen for the test board. At first it seemed that it did line up, but when we plugged in the test board the 3.3V led stayed of, meaning we had a short somewhere in our circuit. After some measurements we concluded that it was the flash chip end consequently removed it. This fixed the short problem.
- The diode we selected for the reverse plugin protection for the external power supply, didn't physically fit on the board, so we bridged it and called it a day.
- The problem regarding the clock had us stumbled the longest. When we first put an Arduino bootloader on the board and noticed that nothing worked, we concluded rather quickly that it was due to the clock. But rather than fix the clock issue we tried to use bare C to program the microcontroller. This worked initially, we got some LEDs to blink. All the uploading however,

went via a Segger directly connected to the chip. The next step to see if our board was capable, is to see if we could get our onboard neopixel to blink. This however, did not go as smooth as we had hoped. To use a neopixel you need to be able to generate a signal that is capable of switching H->L or L->H withing 90ns. We could not get this to work because our board was running on 8KHz instead of the 48MHz it is capable of. To remedy this we tried to link the internal oscillators in such a way that we could generate close to 48MHz (as seen in the figure below).

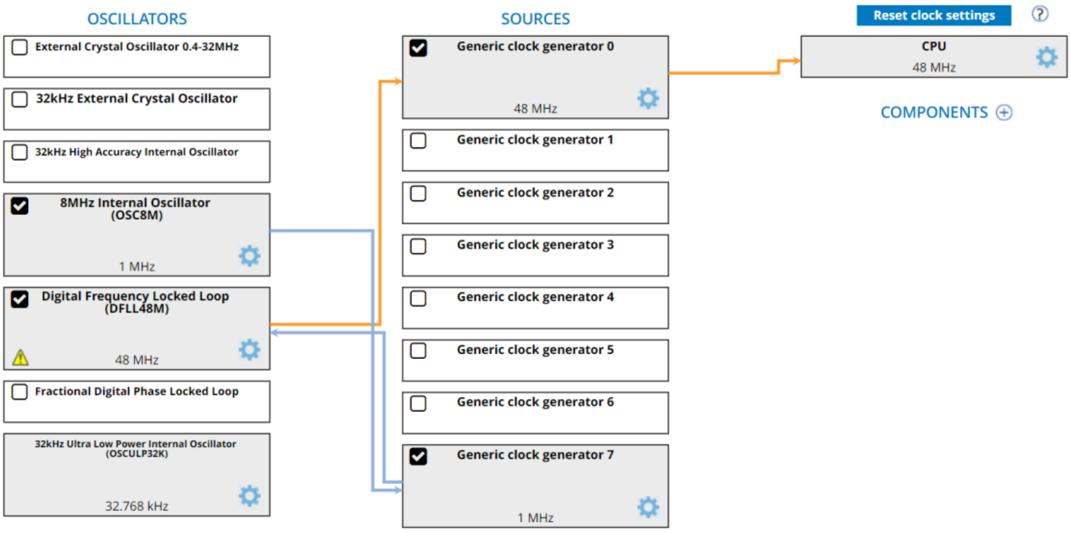

Figure 6 internal clocks of the samd21j18a

We eventually succeeded in this and using bare C we were now capable of generating a signal with a high enough frequency. This frequency – sadly – was not capable of sustaining anything you could call stable, so this also ended in a dead end.
So we decided to combat the problem at its roots, we fixed the oscillator. The original problem was that we did not have the correct load capacitor values (C1 and C2).

5.4.2 – Assembling the flexible led strips

Luckily there weren't a lot of problems regarding the assembly of the flexible led strips. There were however two small inconveniences. Some of the components of the led strip with the non-addressable LEDs came in late, so we couldn't yet assemble them. One of the LEDs on the addressable led strip caused a short underneath by its pads. This however was a quick fix.

5.4.3 – Assembling the spectrometer

Assembling the spectrometer was a job without to much complications. The project that we followed did not mention witch raspberry pi we needed so at first we had a pi 3 and the software did not work on it. Then we tried the pi 4 and that worked fine.

To mount the spectrometer we had to develop a stand, shown in the figure below.

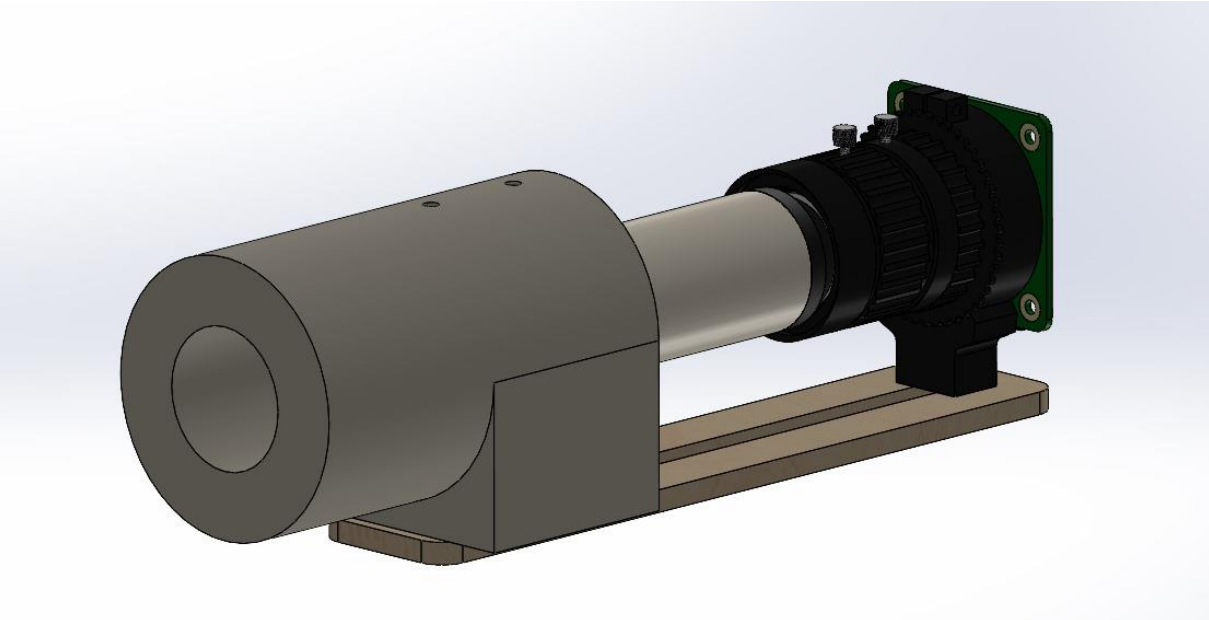

Figure 7 spectrometer stand

We assembled the spectrometer according to a process better described in the RGBbaby manual.

5.4.4 – Spectrometer code

As previously mentioned, we followed a project for the spectrometer. For a more detailed inside, please refer to our manual or go directly to: https://github.com/leswright1977/PySpectrometer2

5.4.5 – Test board code

The test board has the bootloader from the Atmel Xpro on it. This bootloader matches the samd21j18a pinout the best. With Arduino we can now get the LEDs to light up any colour we want. The previous statement only applies to the addressable LEDs. This is because the pins for the PWM driver don't match up with the bootloader. This is however, not an issue because we realised that the final product will always use some kind of addressable LEDs. The code that we used is from Adafruit's neopixel lib. We can use statements like *strip.fill(colour, z, n);* to change the colour of the strip. This simple functionality is enough for the tests we will be conducting.

6 – Conclusion

Concluding our research, we want to achieve / know witch components and methods we are going to use from now on. The things we have discovered and have the perseverance for are:

- What components we are going to use for the driver board.
- What components we are going to use for the flexible led strip.
- How future groups can use our spectrometer.

6.1 – Driver board components

Due to the problems we encountered with assembling and using the test board, we will make the following changes to the newer iterations.

- We are going to use the SAMD21G18 instead of the SAMD21J18. This is due to the problems we have with addressing and using all the pins that we have available, and the fact that we don't need as much IO.
- We dropped the support for non-addressable LEDs. This includes: the 20 pin ffc type connector, the IO extenders, and PWM chips.

We also made some changes to existing parts of the chip, that didn't work properly or weren't sufficient.

- We adjusted the footprint and location of the debug connector, so it fits properly.
- The new board will support more flexible led strips.

6.2 – Flexible led strip components

We were able to select the LEDs that we will be using from now on. These will be the SMDLX0707RGBTR. We made this decision based on the following properties:

- They are small 0.7X0.7 inch
- They use the same data protocol as neo pixels, which is well documented.
- The don't use a lot of power

As for the touch detection. We tried the buttons but they were all not able to deliver the desired result. So we abandoned that way of detecting touch.

6.3 – Using the spectrometer

The spectrometer will be more useful in later stages of this project. Sadly we weren't able to use the spectrometer to its fullest potential. This is why we recommend that future teams will use it. A guide can be found in our RGBbaby manual.