Product report – RGBbaby

By Manon Sijben, Marlon Reijlink, Emiel Visser and Joris Bol

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:	16-1-2023
Report Version:	v1.0
Project Duration:	
Education Term:	
Students:	2108100 (Joris Bol), 2111740 (Emiel Visser), 2106573 (Manon Sijben), 2107003 (Marlon Reijlink)

Version management

Version	Date	Change	Name
V1.0	5-9-2023	Creation document.	J. Bol
V1.0	26-9-2023	Addition C4 and introduction. (detailed history)	J. Bol
V1.0	10-10-2023	Changes made to requirements.	J. Bol
V1.0	10-12-2023	Added more info.	E. Visser
V1.0	18-12-2023	Addition C2. (detailed history)	J. Bol
V1.0	18-12-2023	Addition C4. (detailed history)	J. Bol
V1.0	19-12-2023	Addition C5.	E. Visser & J. Bol
V1.0	9-1-2024	Added Appendix B.	E. Visser
V1.0	11-1-2024	Spellcheck, added sources, small modifications.	E. Visser
V1.0	12-1-2024	Spellcheck, added sources, small modifications.	E. Visser
V1.0	16-1-2024	Add conclusion, addition C6, addition summary.	E. Visser & J. Bol

Summary

This project, conducted in collaboration with Wilhelmina Children’s Hospital for HAN University of Applied Sciences, aims to develop a realistic skin that can change appearances and respond to user input. The primary objective is to contribute to a larger initiative focused on creating a lifelike doll for nurses to practice scenarios involving premature or problematic babies.

To meet the client's specifications, a comprehensive investigation into component selection was carried out. The key focus was on choosing suitable components that could alter the skin colour of the skin and a device capable of sensing when the skin is touched. The research led to the selection of the SMDLX0707RGBTR led as the skin colour altering device and the AT42QT2120-MMHR as the touch sensing device. Two PCBs were developed to mimic the functionality of the final product during this selection process.

In order to ensure a satisfactory end result, SMART-statements were defined to guide the product development. These statements were translated into a set of functional and technical specifications, aligning with the client's vision after consultation.

Following the definition of project conditions, the physical makeup of the product and software functions were outlined, using flow diagrams and schematics. The fabrication phase commenced, utilizing ECAD software for PCB development and incorporating necessary adjustments. Once the design was finalized, PCBs and corresponding components were ordered based on the preliminary research.

Upon receiving all components and PCBs, assembly took place, and the testing phase began. Unfortunately, significant issues with the PCBs emerged, leading to an inability to upload code or generate light with the led strip. Although testing became impossible, a comprehensive testing plan for potential future projects addressing the same problem was established.

While the current PCB is at this point non-functional, the research conducted holds valuable insights for future endeavours. Recommendations to rectify the issues were identified, promising enhancements for the project's utility once implemented. Collaboration with other educational sources facilitated the creation of a realistic skin, establishing a solid foundation for future innovations in this field. Despite the current challenges, the project has laid a constructive groundwork for further developments and improvements.

Table of contents

1 – Introduction

This project is on behalf of the Wilhelmina Children's Hospital (henceforth this will be called: WKZ). WKZ has a department specialising in premature babies or babies with problems, it is with this department we will be working the closest. WKZ wants a doll on which nurses can practise al kind of different situations. A big part of meaningful practise is meaningful feedback, small or big changes in skin hue is a large part of that.

This rapport answers the question: how we went to work to arrive at a solution and get to a satisfying conclusion. For the development process we will be following the design steps as described in the V-model. WKZ does not explicitly define any conditions, which is why this project is entirely defined in collaboration.

This report is made up of seven chapters, whom all individually describe an important aspect of our project. To make it possible for us to reach a conclusion, in chapter 2 (Preliminary research – ESE) there will be a lot of questions asked and answered. In chapter 3 (functional design) the function of our product will be defined in detail. Next in chapter 4 (technical design) there will be a detailed description of all our hard- and software. Chapter 5 (realisation) describes our process and struggles during the project. You can find all the tests that were conducted in chapter 6 and the conclusion in chapter 7. Chapter 7 also contains possible improvements and recommendations.

2 – Preliminary research – ESE

This next section is the preliminary research we did to select the leds and other components for our PCB(s).

2.1 – Objective and requirements

2.1.1 – 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.

2.1.2 – 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.

2.1.3 – 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.

2.2 – Light sources

Type:	Description:
Link:	19-337/R6GHBHC-A01/2T (Mouser, 2023)
Disadvantages:	Large surface area
Advantages:	Small, inexpensive
Price:	€0,13

Type:	Description:
Link:	WUE 150141M1731 (Reichelt, 2023)
Disadvantages:	Quite tall, expensive.
Advantages:	Bright, small surface area.
Price:	€0,46

Type:	Description:
Link:	KAA-3528RGBS-11 (TME, 2023)
Disadvantages:	Quite tall, expensive.
Advantages:	Bright, small surface area.
Price:	€1,05

Type:	Description:
Link:	RND 135-00252 (Distrelec, 2023)
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 (Distrelec, 2023)
Disadvantages:	Large surface area, expensive.
Advantages:	Bright, very thin.
Price:	€1,24

Type:	Description:
Link:	SMD-LX0707RGB-TR (Mouser, 2023)
Disadvantages:	Relatively expensive.
Advantages:	Very small, doesn't use much IO, low current consumption.
Price:	€0,66

Type:	Description:
Link:	SMLP34RGBN1W3 (Mouser, 2023)
Disadvantages:	Expensive.
Advantages:	Very thin, very small, doesn't use much IO.
Price:	€1,26

Type:	Description:
Link:	IN-P55QDTRGBW (Mouser, 2023)
Disadvantages:	Large surface area, relatively expensive.
Advantages:	Thin, bright.
Price:	€0,65

Type:	Description:
Link:	CLQ6B-TKW-S1L1R1H1TBB7935CC3 (Mouser, 2023)
Disadvantages:	Large surface area, relatively expensive.
Advantages:	Comes with white element, thin.
Price:	€0,85

Type:	Description:
Link:	SK6812 (3535) (Mouser, 2023)
Disadvantages:	Still relatively large.
Advantages:	Doesn't use much IO, low current consumption.
Price:	€0,49

Type:	Description:
Link:	WS2812b (5050) (TME, 2023)
Disadvantages:	Large surface area.
Advantages:	Doesn't use much IO, low current consumption.
Price:	€0,49

2.3 – Touch sensors

Type:	**Description:**
Link:	TL3342F260QG (Mouser, 2023)
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 (Mouser, 2023)
Disadvantages:	-
Advantages:	Super thin, large actuation zone, cheap, relatively high actuation force.
Price:	€0,26

Type:	**Description:**
Link:	MPR121QR2 (DigiKey, 2023)
Disadvantages:	Discontinued, relatively difficult to integrate.
Advantages:	Small, relatively easy to control, lots of channels, stable.
Price:	?

Type:	**Description:**
Link:	AT42QT2120 (Mouser, 2023)
Disadvantages:	Relatively difficult to integrate.
Advantages:	Small, relatively easy to control, lots of channels, stable.
Price:	€1,59

2.4 – 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.

2.4.1 – 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

Table 1 Light sources

2.4.2 – 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

Table 2 Touch sensors

2.4.3 – 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

Table 3 Other components

2.4.3.1 – 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.

2.4.3.2 – 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.

2.4.3.3 – 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.

2.4.3.4 – 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 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.

2.4.3.5 – 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.

2.5 – 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.

2.5.1 – 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.

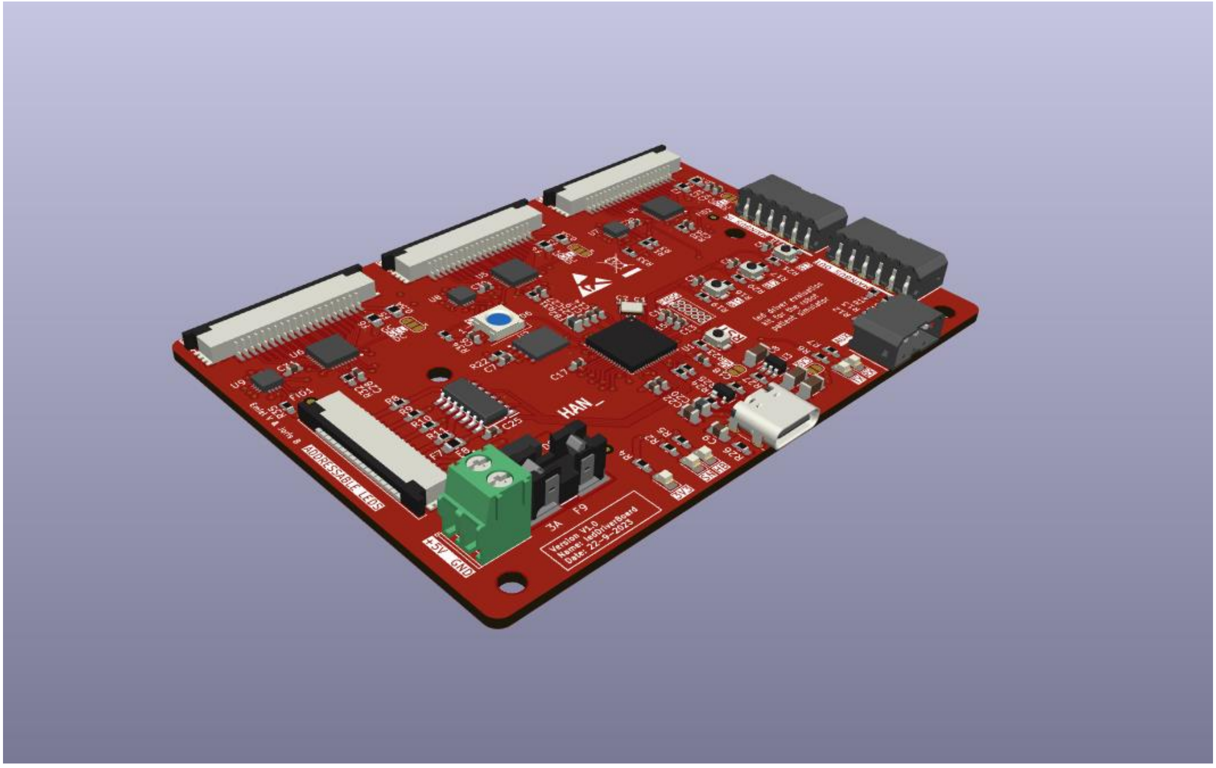


Figure 1 LedDriverBoardV2

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.

2.5.2 – 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

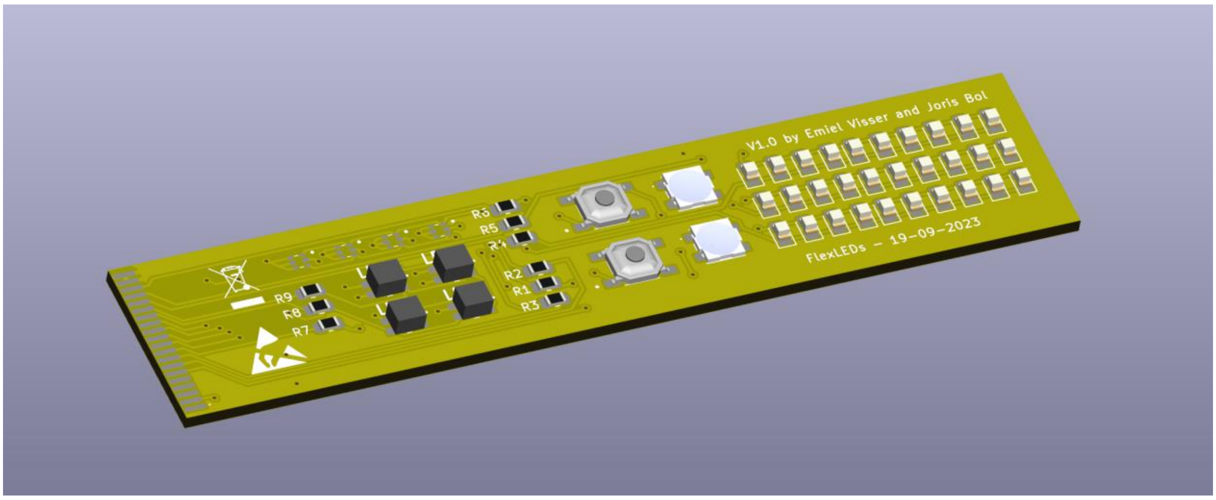


Figure 2 flexLeds

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

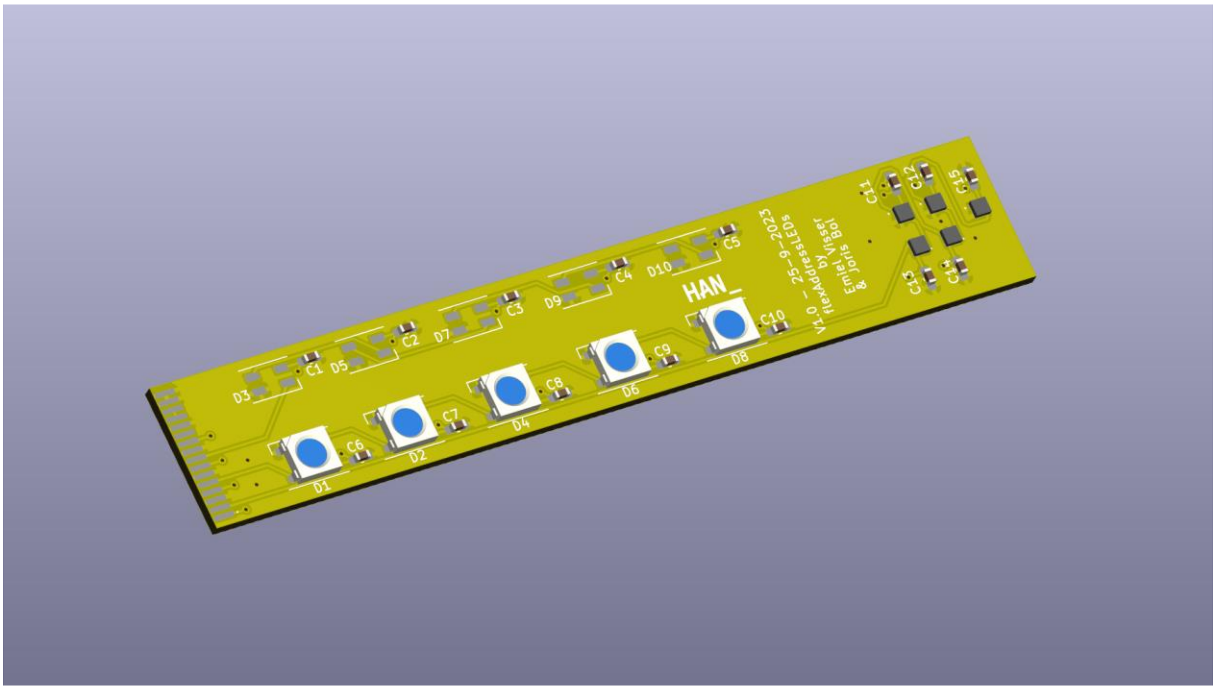


Figure 3 flexAddressLeds

2.5.4 – 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.

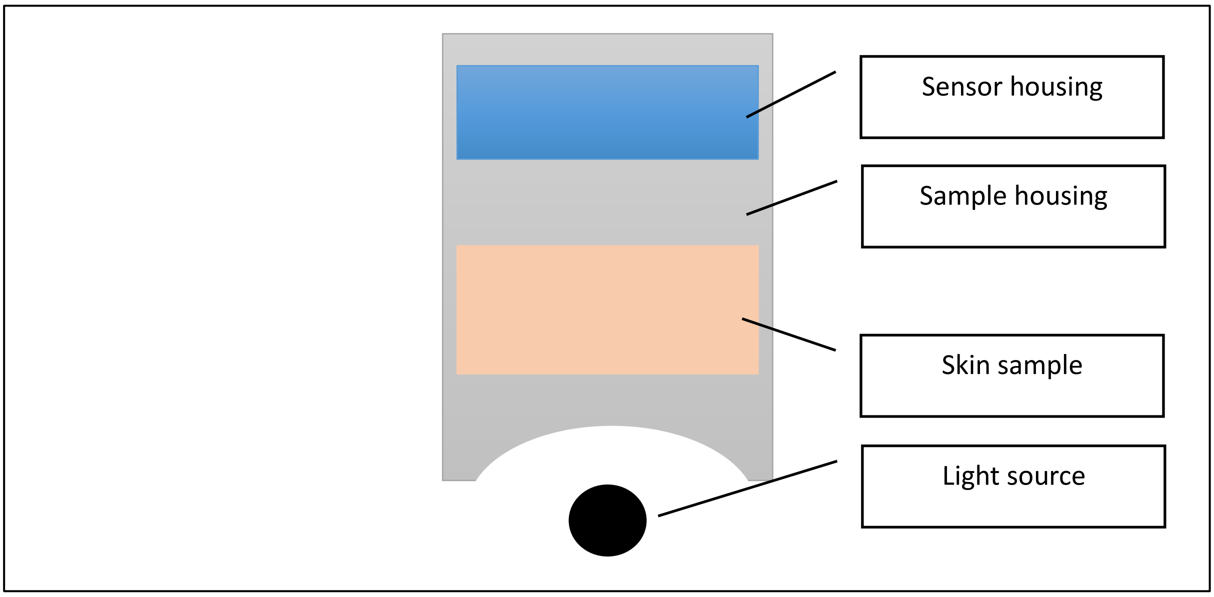


Figure 4 Schematic view testbench

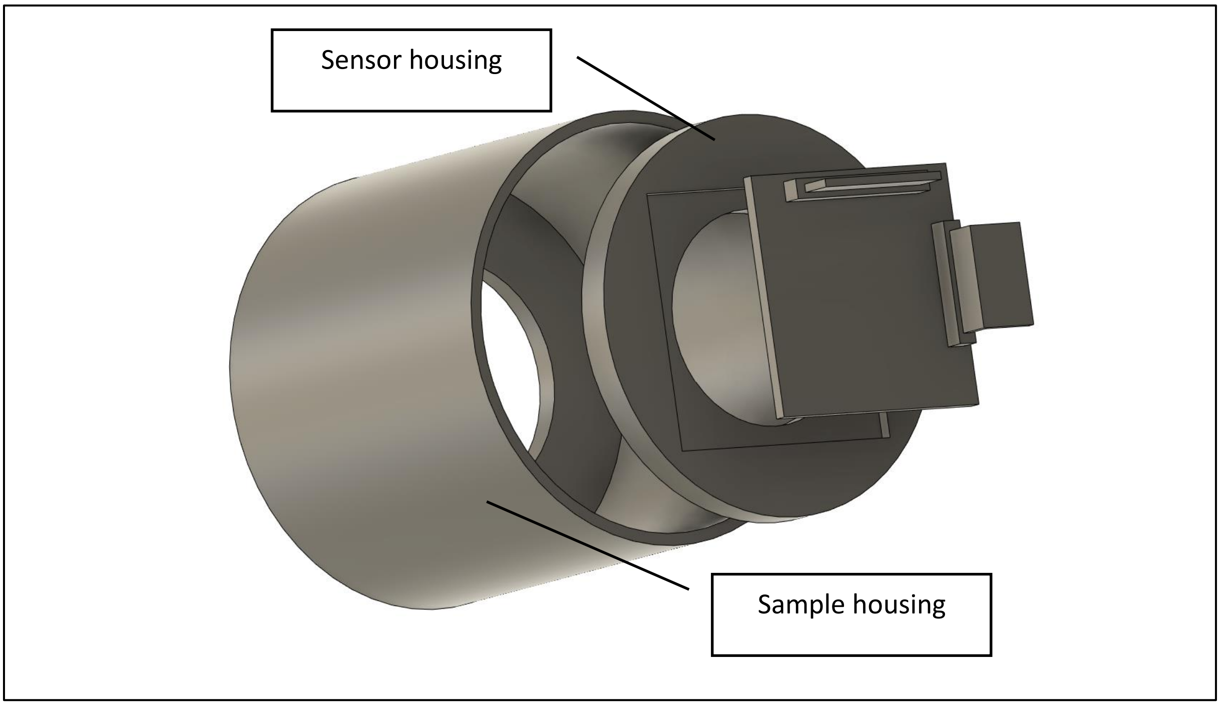


Figure 5 3D view testbench

The reason that we need to use this enclosure is to ensure 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.

2.5.5 – 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.

2.5.4.1 – 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).

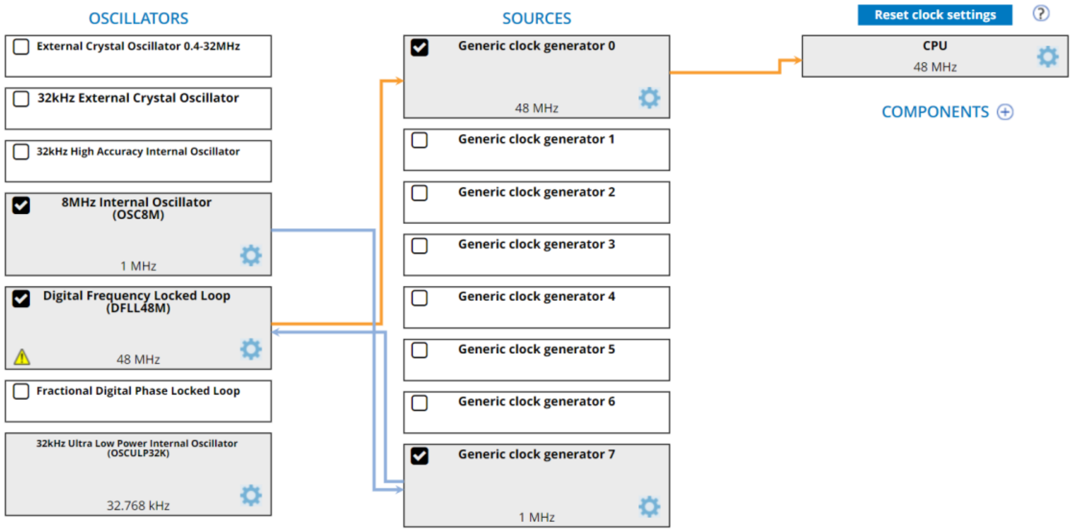


Figure 6 Internal clock configuration of the samd21j18a

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).

2.5.4.2 – 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.

2.5.4.3 – Assembling the spectrometer:

Assembling the spectrometer was a job without too many complications. The project that we followed did not mention which 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.

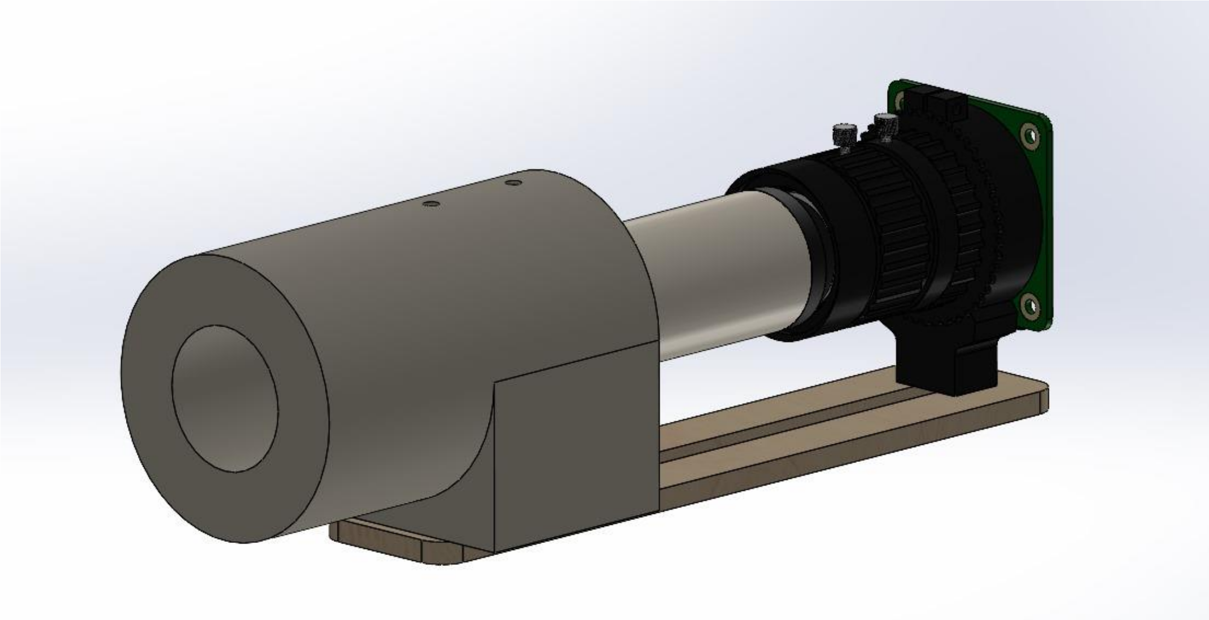


Figure 7 Spectrometer stand.

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

2.5.4.4 – 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 (Wright, 2023)

2.5.4.5 – 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.

2.6 – Conclusion

Concluding our research, we want to achieve / know which 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.

2.6.1 – 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 connector, the IO extenders, and PWM chips.

We also made some changes to existing parts of the chip, which 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.

2.6.2 – 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 neopixels, 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.

2.6.3 – 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.

3 – Functional design

3.1 – Functional design

The primary function of our product is to introduce a variation in skin colours. The following list of functional requirements has been defined in consultation with our client. This list consists of three columns: The requirement number, prioritization based on the MoSCoW method and formulation of the requirement according to the SMART method.

With the MoSCoW method we can determine the importance of each specification. The M stands for *must have*. We must implement this specification in our final product. The S stands for *should have.* denoting features that are highly desirable and expected in the final product. The C stands for *could have*. suggesting optional features that may be considered for inclusion. The W stands for *won't have*. indicating that these features won't be incorporated into the final product due to their perceived difficulty or lack of importance.

The SMART method dictates that each specification must be formulated according to the following guide rules:

- Specific: Goals should be clear and specific.
- Measurable: Goals should have measurable criteria for tracking progress.
- Achievable: Goals should be realistic and attainable.
- Relevant: Goals should be relevant to your overall objectives.
- Time-bound: Goals should have a defined timeframe or deadline.

Using the SMART method helps ensure that goals are well-defined, achievable, and aligned with your larger objectives, increasing the likelihood of success.

Functional specifications		
#	MoSCoW	**Description**
F1	**M**	**The skin should be able to change colour realistically.**
F1.1	**M**	The skin must be able to turn yellow.
F1.2	**M**	The skin must be able to turn red.
F1.2	**M**	The skin must be able to turn dark blue.
F1.3	**M**	The skin must be able to turn pale.
F1.4	**S**	The skin shouldn't emit a lot of light.
F1.5	**S**	The skin should be able to accurately simulate petechiae. **Added on 10-10-23**
F2	**S**	**The skin should feel realistically.**
F2.1	**S**	The skin should not feel sticky.
F2.2	**S**	The flexibility of the skin should match the flexibility of real skin.
F3	**S**	**The skin should look realistically.**
F3.1	**S**	The skin shouldn't enter the uncanny valley*.
F3.2	**S**	The colour of the skin should be realistic.
F4	**C**	**The skin could react to being touched.**
F4.1	**C**	The touched spot could display capillary refill**.
F5	**S**	**The system should work on different skin colours.**
F5.1	**S**	White skin colour.
~~F5.2~~	~~**S**~~	~~Brown skin colour.~~ **Removed on 10-10-23**
~~F5.3~~	~~**S**~~	~~Dark brown colour.~~ **Removed on 10-10-23**
F6	**M**	**We must test our system on different skin colours.**
F6.1	**S**	White skin colour.
F6.2	**S**	~~Brown skin colour.~~ **Removed on 10-10-23**
F6.3	**S**	~~Dark brown colour.~~ **Removed on 10-10-23**

*Used in reference to the phenomenon whereby a computer-generated figure or humanoid robot bearing a near-identical resemblance to a human being arouses a sense of unease or revulsion in the person viewing it.

**The 'capillary refill' time is the time it takes for colour to return to a distal capillary network after pressure has been applied.

3.2 – Technical specifications

The table below presents the technical specifications, adhering to the same formatting principles we applied to the functional specifications table in paragraph 1.

Technical specifications		
#	**Moscow**	**Description**
T1	**M**	**The used microcontroller is the SAMD21~~J18A-M~~. Removed on 4-12-2023**
T1.1	**M**	The microcontroller must be placed on a PCB designed by us.
T2	**M**	**The used silicone is Eurosil 8 or 33 Transparant.**
T2.1	**M**	The silicone must be used to create the skin for the baby.
T2.2	**M**	The silicone must not tear apart.
T3	**M**	**The used pigments come from NEILL's.**
T3.1	**M**	X gram of pigment is used for one litre of silicone.
T3.2	**M**	X gram of flocking material is used for one litre of silicone.
T4	**M**	**The code must be written in CPP.**
T5	**S**	**We should make use of flex PCBs.**
T5.1	**C**	We could use FFC connectors.
T5.2	**C**	We could use x number of buttons.
T5.3	**C**	We could use leds.
T5.3.1	**S**	*We should supply the leds separately with power.*
T6	**M**	**The system must be modular.**
T6.1	**M**	The control PCB must be detachable from the main system.
T6.2	**M**	The flex PCBs must be detachable from the control PCB.
T7	**M**	**The control PCB must be connected to the main I2C bus to the rest of the system.**
T7.1	**M**	There must be two I2C ports. An input and output port.
T7.2	**M**	The standard I2C protocol must be used.
T8	**M**	**The PCBs must be made as compact as possible.**
T9	**M**	**The system must produce as little heat as possible.**
T10	**S**	**The status of the PCB must be made visible to the user.**

3.3 – User interface

A user interface (UI) serves as the medium through which a user interacts with a system or application. Examples of user interfaces include physical screens, on-screen applications, or even a basic indicator like a led, all of which enable users to engage with a machine or software.

In the following paragraph, we will detail the specific types of user interfaces we intend to use in our project, elaborate on the information they will present to the user, and outline our implementation strategy.

3.3.1 – On board interface / heartbeat

We intend to communicate vital information regarding our system's operation using six leds. These leds will serve as intuitive indicators, providing users with a clear understanding of the system's status.

In the following paragraph, we will explain the specific roles of these leds and describe how they can be effectively utilized to monitor the system's condition.

3.3.1.1 – Serial leds:

The Serial leds are designed to provide a visual representation of both the RX (Receive) and TX (Transmit) signals between the microcontroller.

These leds will be placed next to each other. We did this to maintain a good overview of the serial communication status.

3.3.1.2 – Heartbeat led:

As the name suggests, this led blinks like a heart to indicate that the board is still 'alive.' The led will turn on and of every second.

3.3.1.3 – Sense led:

The sense led turns on when one of the buttons is pressed. This is extremely helpful while debugging.

3.3.1.4 – RGB led:

The RGB led indicates the state of the PCB. The distinct colours indicate the following states:

Colour:	State:
Off	Nothing is wrong, PCB is off.
White	The PCB went into hard fault.
Red	There is no connection between the PCB and the main board.
Green	INIT finished successfully.
Blue	The PCB is in test mode.

3.3.1.5 – 3V3 led

The 3V3 led indicates when a 3.3 volt is present. This means that when the board is connected to power, the led turns on. This is not dependent on the chip.

3.4 – User interfacing

The system will interact with the user in a before specified manner, but not only the user will have impact. Our board will be largely controlled by another piece of hardware and software. This interaction is by touch in case of the user and by I2C in case of the other hardware.

A UML use case diagram is shown below, it describes what actions can be taken and how our system will react to them. In our case there are two primary actors, both depicted on the left.

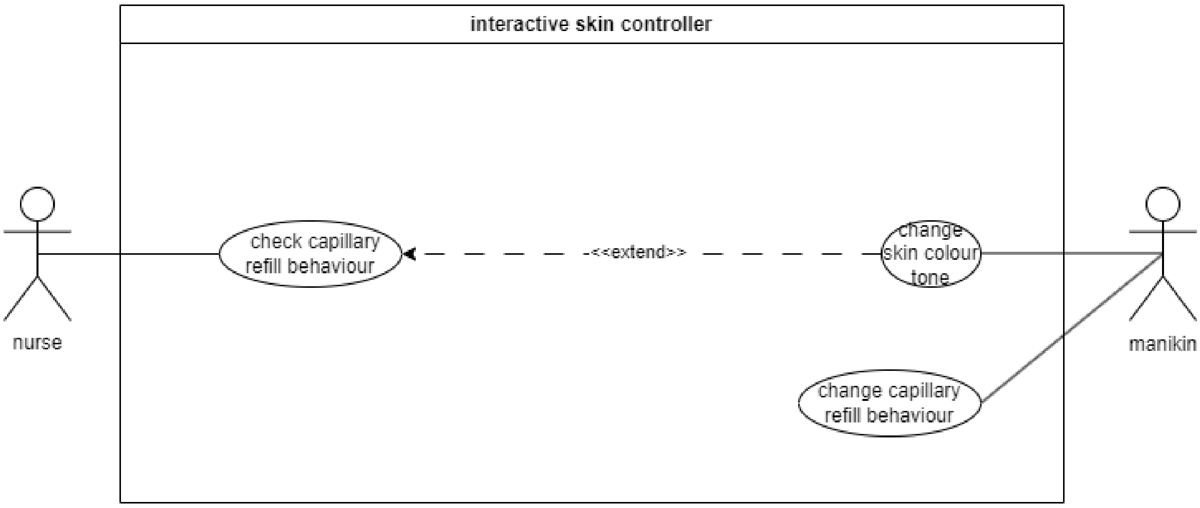


Figure 8 UML use case diagram

4 – Technical design

About the exact reason why we choose these sensors and actuators we would like to point you to our preliminary research.

4.1 – Architecture

In figure 1 a diagram is shown showing how our system will be controlled. The boxes contain one or more items which the board will interface with. The arrows between the boxes and board indicate the direction of the data/energy flow. The protocol/voltage is placed on the arrow.

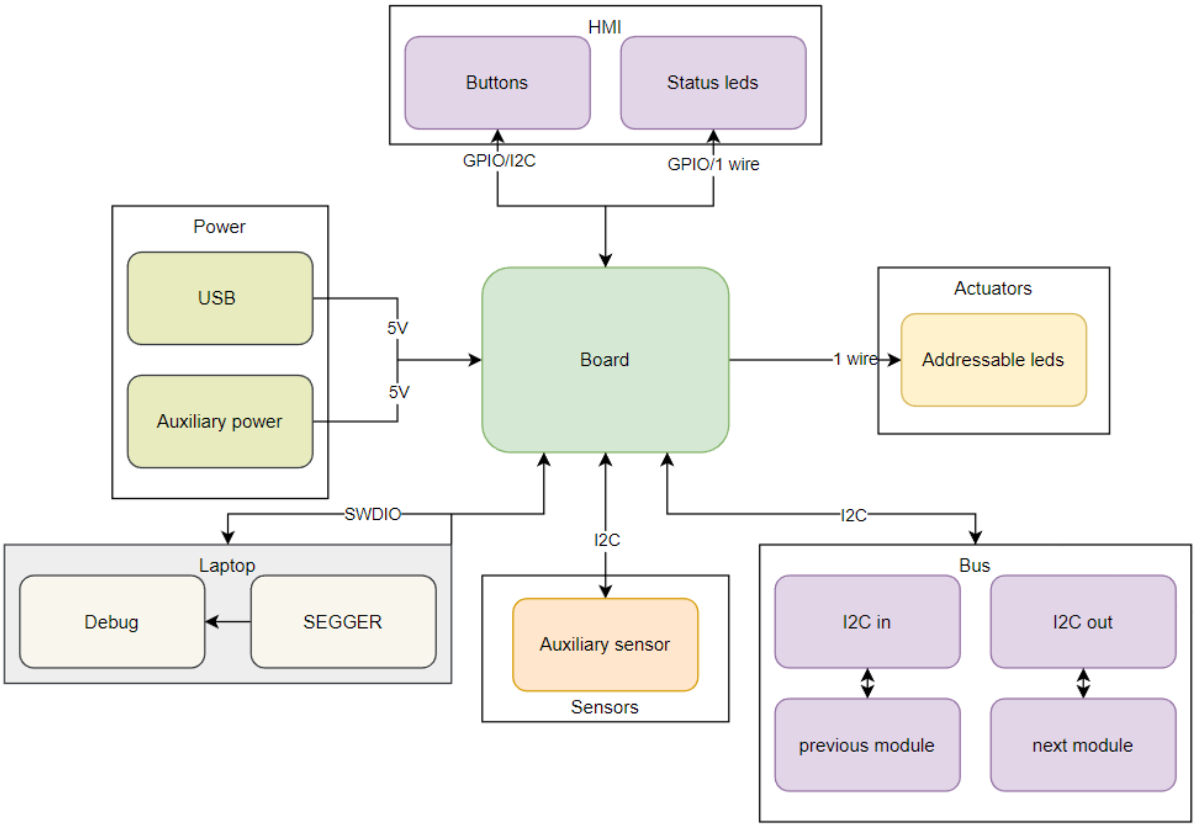


Figure 9 System interconnections

4.2 – Interfaces

In this paragraph all of the interfaces between the board and the items are described. An overview of the interfaces and items can be seen in figure 1.

4.2.1 – Power

In this paragraph the power section of our system will be described.

4.2.1.1 – USB-C:

To deliver power and upload code to our board we have added a UCB type C port to our PCB. We have added two test points so that the USB protocol can be debugged if necessary. The USB termination resistors are placed in the main sheet and are there to reduce ringing on the USB data lines. An ESD protection diode protects the host (e.g. a laptop) from ESD. The USB C port can deliver a maximum of 3A in this configuration.

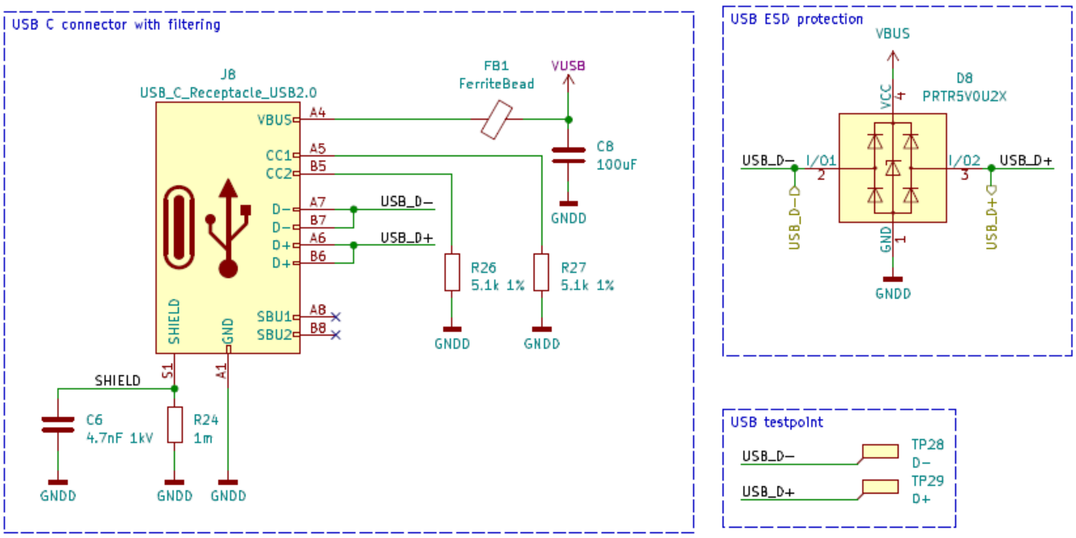
Note: Don't come close to this maximum though because the traces coming out of the port might not be able to handle 3A!

Requirement **supports T7:**

Specification 1.1:

No software scheme for mounting holes and fiducials.

Specification 1.2:



4.1.1.2 – Auxiliary power:
A USB C plug can be quite bulky, so we decided to add another way of powering the system. That's way we added a JST-XH port where 5V power can be fed into the system. We also added a fuse holder. We recommend using a fast blow fuse to protect the system in an overcurrent event.

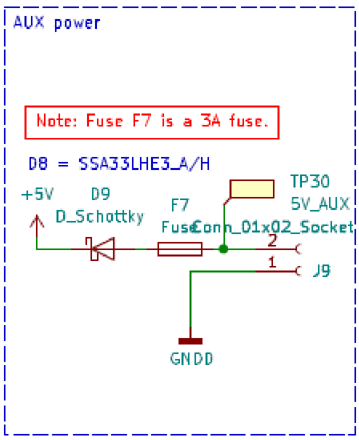
Note: The maximum current for prolonged periods is about 2.5A. For short periods 3A is possible.

Requirement **supports T7:**

Specification 2.1:

No software scheme for aforementioned element.

Specification 2.2:



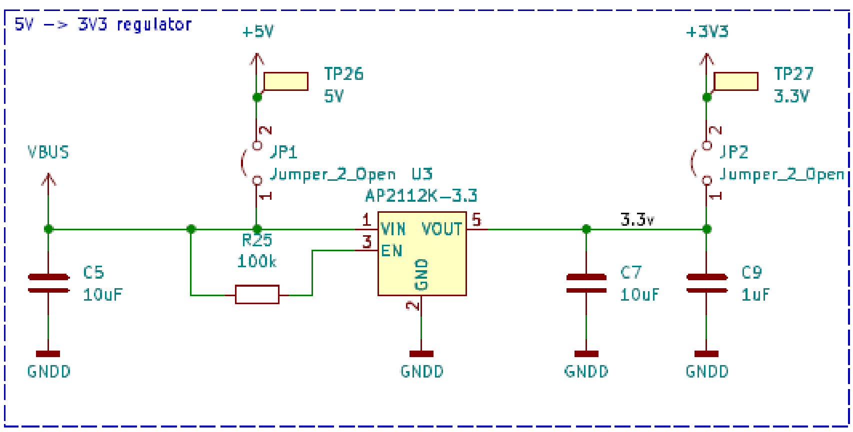
4.1.1.3 – 3.3V regulator:
We have decided to go with an AP2112K type regulator. This regulator drops the auxiliary and/or USB voltage down to 3.3V which is used by the microcontroller and other components. The regulator is decoupled using three decoupling capacitors and will deliver 3.3V when 5V is applied to its input.

Requirement **supports T7:**

Specification 3.1:

No software scheme for aforementioned element.

Specification 3.2:



4.2.2 – Status leds:

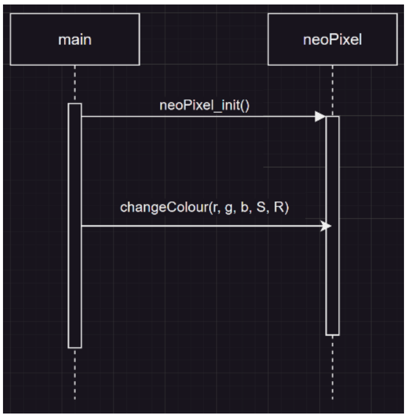
We have integrated multiple leds onto the board to provide users with clear visual indicators for various processes taking place on the controller board. These leds serve as a user-friendly interface, offering real-time feedback on the ongoing activities and status of the system.

4.2.2.1 – Status led (neopixel):

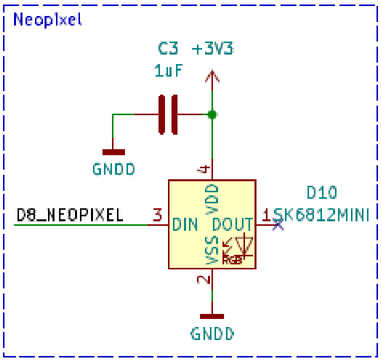
We have opted for the SK6812MINI neopixel led due to its compact package size. Additionally, these leds offer significant IO savings, a crucial factor because we decided to go with the ATSAMD21G18 instead of the original ATSAMD21J18 microcontroller. Another compelling reason for choosing this specific led is its compatibility with the existing protocol we are using for other leds. This not only streamlines our development process but also minimizes the need for additional code, promoting efficiency and consistency across our lighting system.

Requirement: **supports T10**

Specification 4.1:



Specification 4.2:

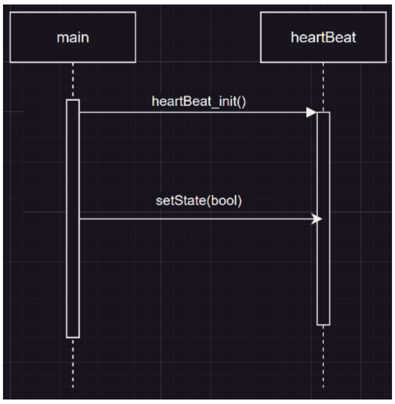


4.2.2.2 – HB led (heartbeat):

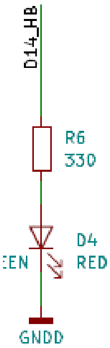
The Heartbeat led blinks at a fixed frequency. In the event that this led stops blinking, it indicates a system crash or that the system is stuck in a loop. Typically, a system reset is the recommended solution to address such issues.

Requirement **supports T10**

Specification 5.1:



Specification 5.2:

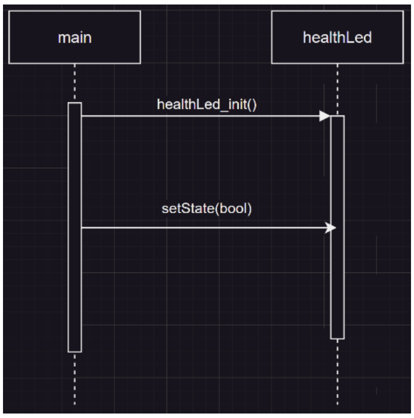


4.2.2.3 – Health led:

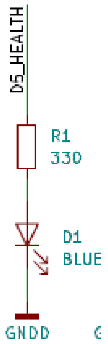
This led will be on when the board is in good health. And will turn of when the board encounters errors that it can't handle properly.

Requirement **supports T10**

Specification 6.1:



Specification 6.2:



4.2.2.4 – Voltage indicator leds:

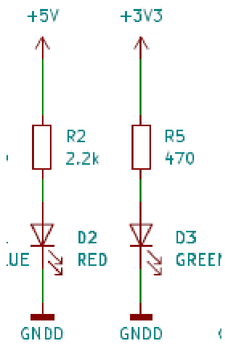
We've added voltage indicator leds on the board for a quick check on the availability of the two supply voltages.

Requirement **supports T10**

Specification 7.1:

No software scheme for aforementioned element.

Specification 7.2:



4.2.2.5 – Receive and transmit leds:

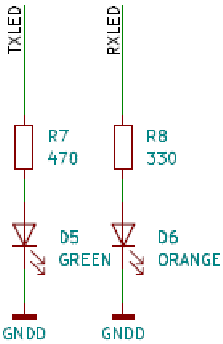
To give the user an indication if the serial bus is working, we have added a TX and RX led.

Requirement **T10**

Specification 8.1:

No software scheme for aforementioned element.

Specification 8.2

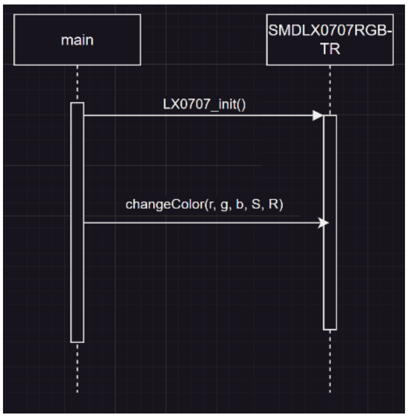


4.2.3 – Main leds

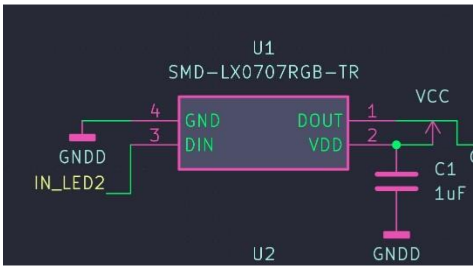
We have found that the SMD-LX0707RGB-TR leds fit our requirements.

Requirement **T5.3**

Specification 9.1:



Specification 9.2



4.2.3.1 – Neopixel output drivers:

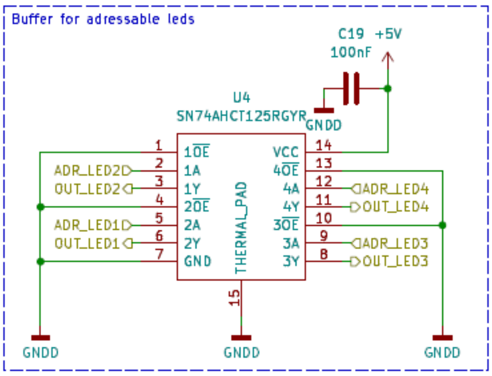
To convert the 3.3V microcontroller output signal to a 5V output signal which results in a more stable signal. We have chosen the SN74AHCT125RGYR buffer IC. In our tests we have used the SN74AHCT125DR. the RGYR version has a smaller package which is why we went with this version for our latest revision. Every in- and output has a test point.

Requirement **supports T5.3.**

Specification 10.1:

No software scheme for aforementioned element.

Specification 10.2:



4.2.3.2 – FFC/FPC flat cable connectors:

To connect the flex PCBs to our control PCB we are using 16pin FFC/FPC connectors. We are using 1A resettable fuses on the 5V and 3.3V power rails. The 62-ohm resistors are there to prevent ringing on the data lines. We have also opted to breakout an I2C bus to pin 11 and 12 on the connector which enables a next group to add a I2C enabled device to be used on the flex PCBs. We also decided to breakout two IO's to pin 8 and 9 on the connector. The user can use the for general purpose applications.

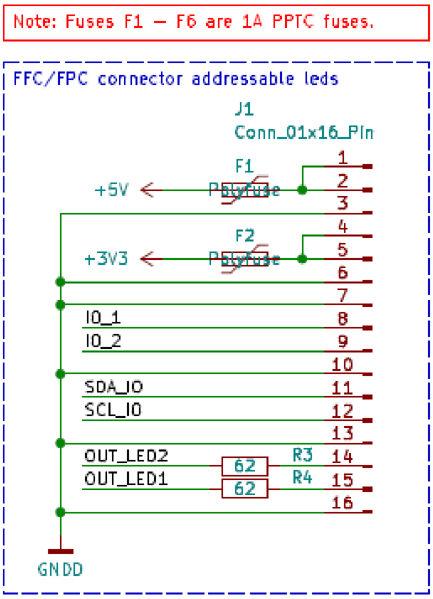
Note: The connector only makes contact on the under side so if you want your parts to be on top of the flexible PCB you need to place the connector on the underside!

Requirement **T5**

Specification 11.1:

No software scheme for aforementioned element.

Specification 11.2:



4.2.4 – Microcontroller

We have chosen the ATSAMD21G18A microcontroller. This microcontroller is used by the Adafruit feather M0 express board which makes integration with Arduino or some other platform a lot easier.

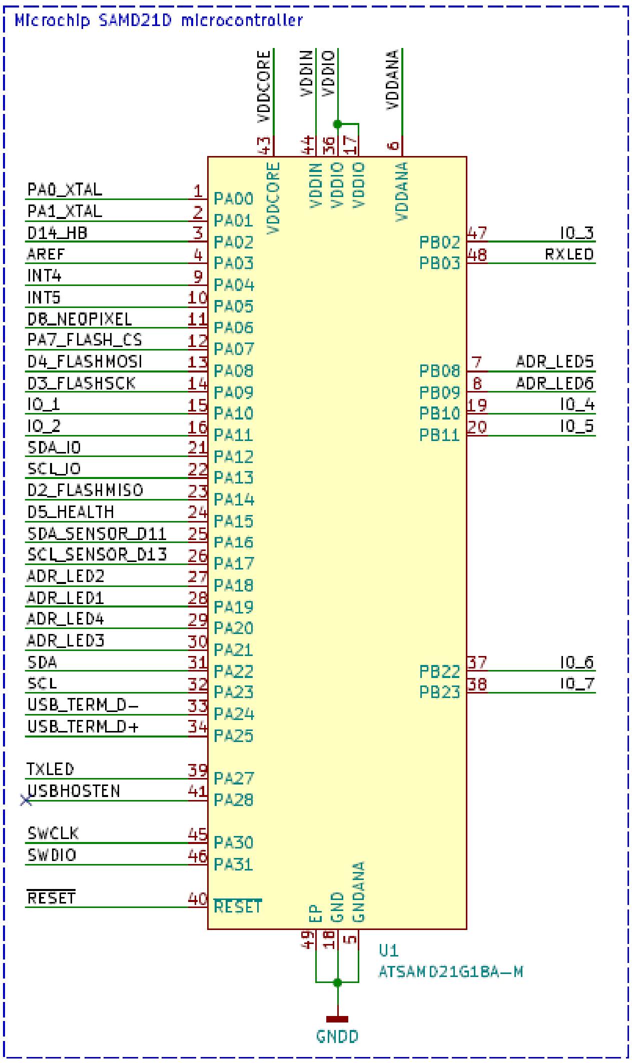
Requirement **T1**

Specification 12.1:

The software for the microcontroller is the combination of the other specified software schematics.

Specification 12.2:

Note: The decoupling capacitors and filtering for the MCU can be found in the *Power* sheet of the schematic.



4.2.4.1 – Clock:

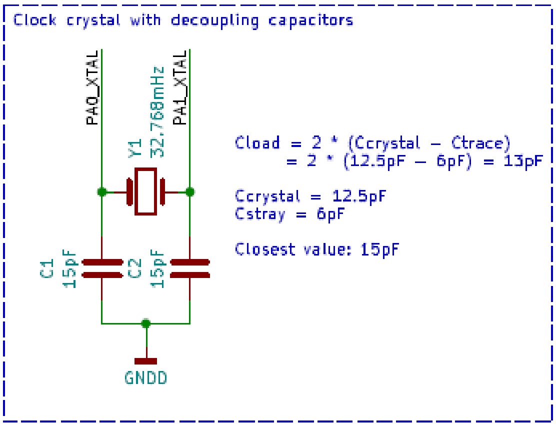
The SAMD21G18A runs on a 32.768mHz clock crystal. C1 and C2 are load capacitors. These are crucial for getting the correct clock speed. If you end up going with a different clock crystal, you will probably need te recalculate the load capacitors. The formula is in the schematic. But this formula is incorrect. For the correct formula have a look at paragraph 5.1.2. You can verify the clock speed at the test points we have placed on the clock pins.

Requirement **supports T1**

Specification 13.1:

No software scheme for aforementioned element.

Specification 13.2:

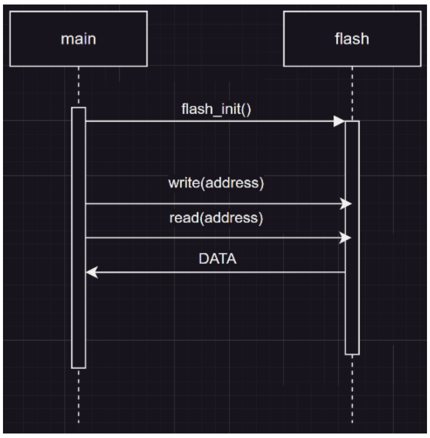


4.2.4.2 – Flash memory:

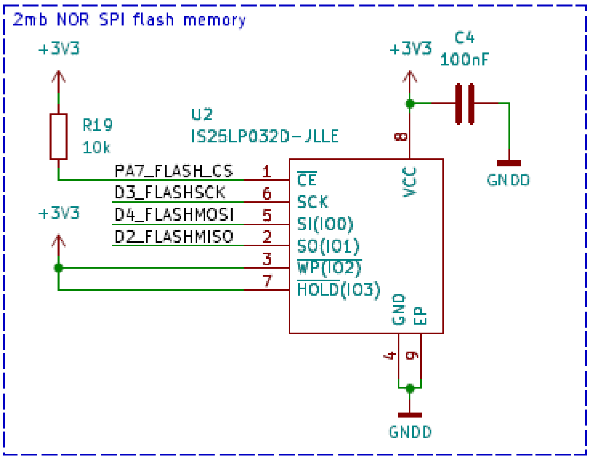
We are using a 64kb flash chip. The chip communicates via SPI. Every communication pin has been connected to a test point making debugging easier.

Requirement **general functionality**

Specification 14.1:



Specification 14.2:



4.2.4.3 – Debug header:

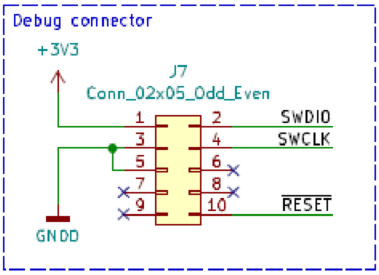
If you want to write software to the chip (e.g. flashing a bootloader) you can use this port with a programmer like the J-LINK mini edu. This port can also be used to debug the processor and peripherals. We have added test points on the reset and data lines just in case.

Requirement **general functionality**

Specification 15.1:

No software functions for debug header.

Specification 15.2:



4.2.4.3 – Reset button:

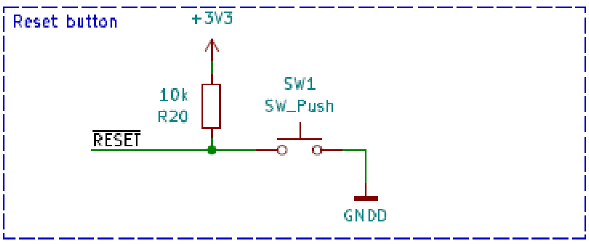
To reset the system, we can use the physical reset button.

Requirement **general functionality**

Specification 16.1:

No software schemes for reset button.

Specification 16.2:

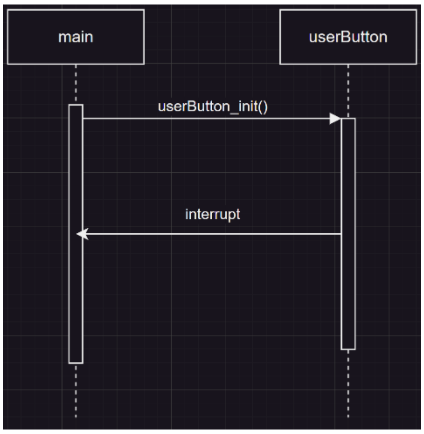


4.2.5 – User button:

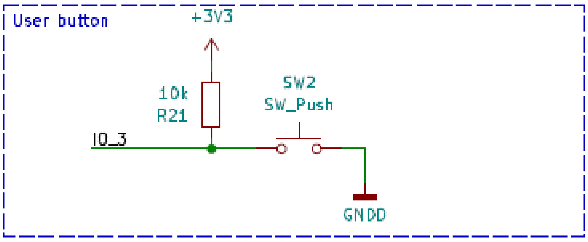
We had one IO left so we routed that to a button. This button also has a test point.

Requirement **general functionality**

Specification 17.1:



Specification 17.2:



4.2.6 – I2C bus components

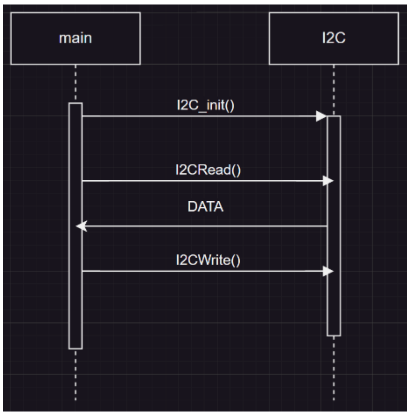
There are three I2C busses available. All I2C busses have test points.

4.2.6.1 – Main bus:

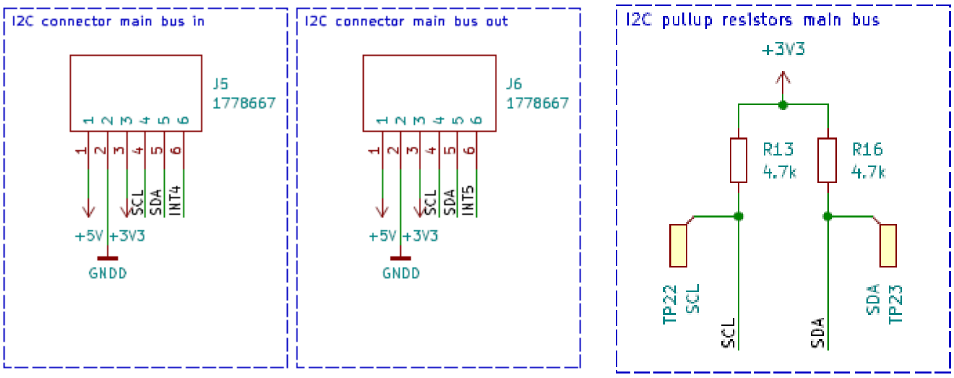
The main bus is the communication bus that communicates with other modules in the baby doll. The main bus has two ports. Main bus in and main bus out. You should connect the output cable of the previous module to the main bus in port on the PCB INT4 is connected to main bus in. The main bus out port connects to the next module. INT5 is connected to this port.

Requirement **T7**

Specification 18.1:



Specification 18.2:



5.2.6.2 – IO bus:

We wanted to maintain the possibility to add other I2C enabled devices to the flex PCBs. This is why we have broken out one I2C bus to every FFC/FPC connector. This bus also has test points.

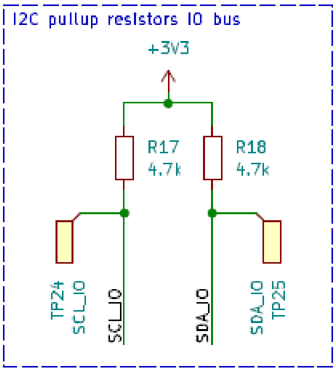
Requirement **supports T6**

Specification 19.1:

See specification 18.1.

Specification 19.2:

Note: I2C output pins can be found in paragraph 4.2.3.2 – FFC/FPC flat cable connectors



4.2.6.3 – Auxiliary bus:

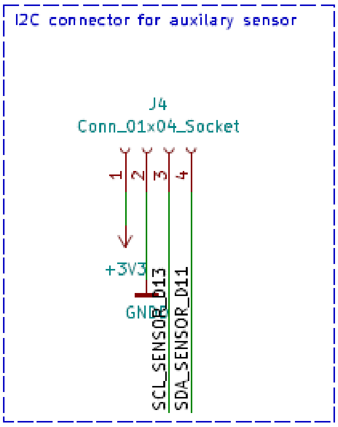
This bus is meant for auxiliary sensors. This bus does not have an interrupt pin connected to it and the output voltage is 3.3V only. We have used a JST-XH connector. We also added more test points tot the data pins.

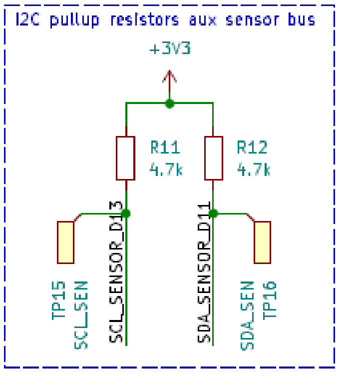
Requirement **supports T6**

Specification 20.1:

See specification 18.1.

Specification 20.2:





4.2.7 – Mounting holes and fiducials:

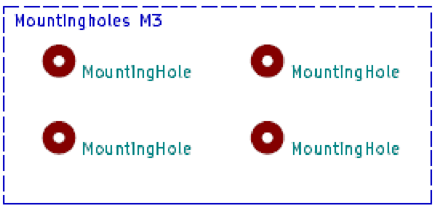
We have added mounting holes to the PCB the hole size is 3.2mm which means a M3 screw can be used to secure the PCB into place. The exposed copper pads near the screw hole are **not** connected to ground. The fiducials are positioning marks for the PCB manufacturer and pick-and-place machine.

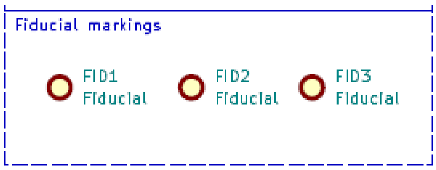
Requirement **general functionality**

Specification 21.1:

No software scheme for mounting holes and fiducials.

Specification 21.2:





4.3 – Software

In this paragraph the software structure is described.

4.3.1 – Main structure

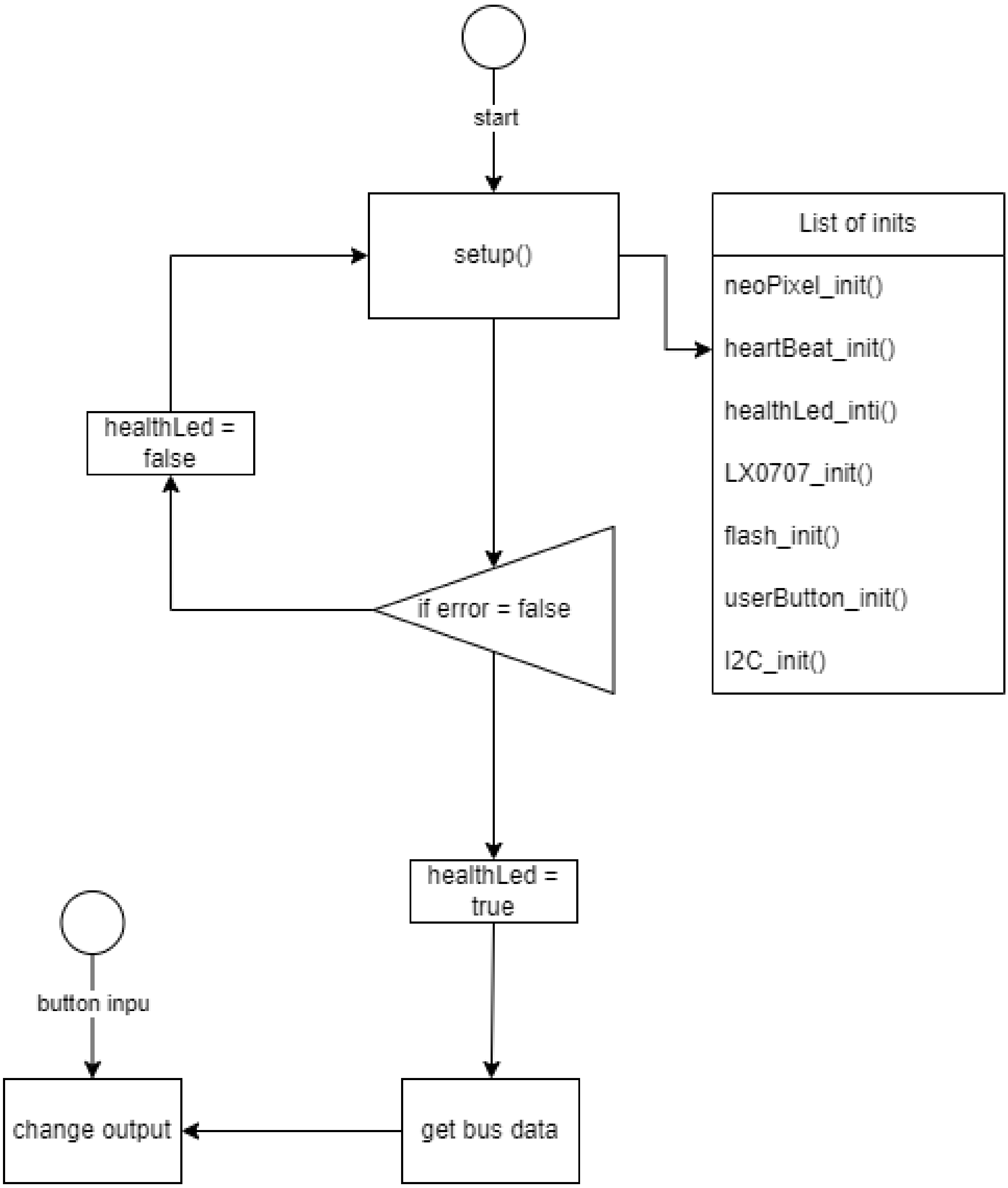


Figure 10 Main software structure

5 – Realization

In this part of the report, we describe the realisation of the project in combination with several design choices we have made to come to our final product.

5.1 – Hardware

In this paragraph we will describe all the hardware with relevant design choices that we have made.

We have used KiCad 7.0.9 to create the PCBs. For the 3D files we used SolidWorks 2022 and Fusion360.

5.1.1 – I/O diagram

The input/output diagram for our final PCB is listed below. In this diagram we have shown which pins are used to control the different functionalities.

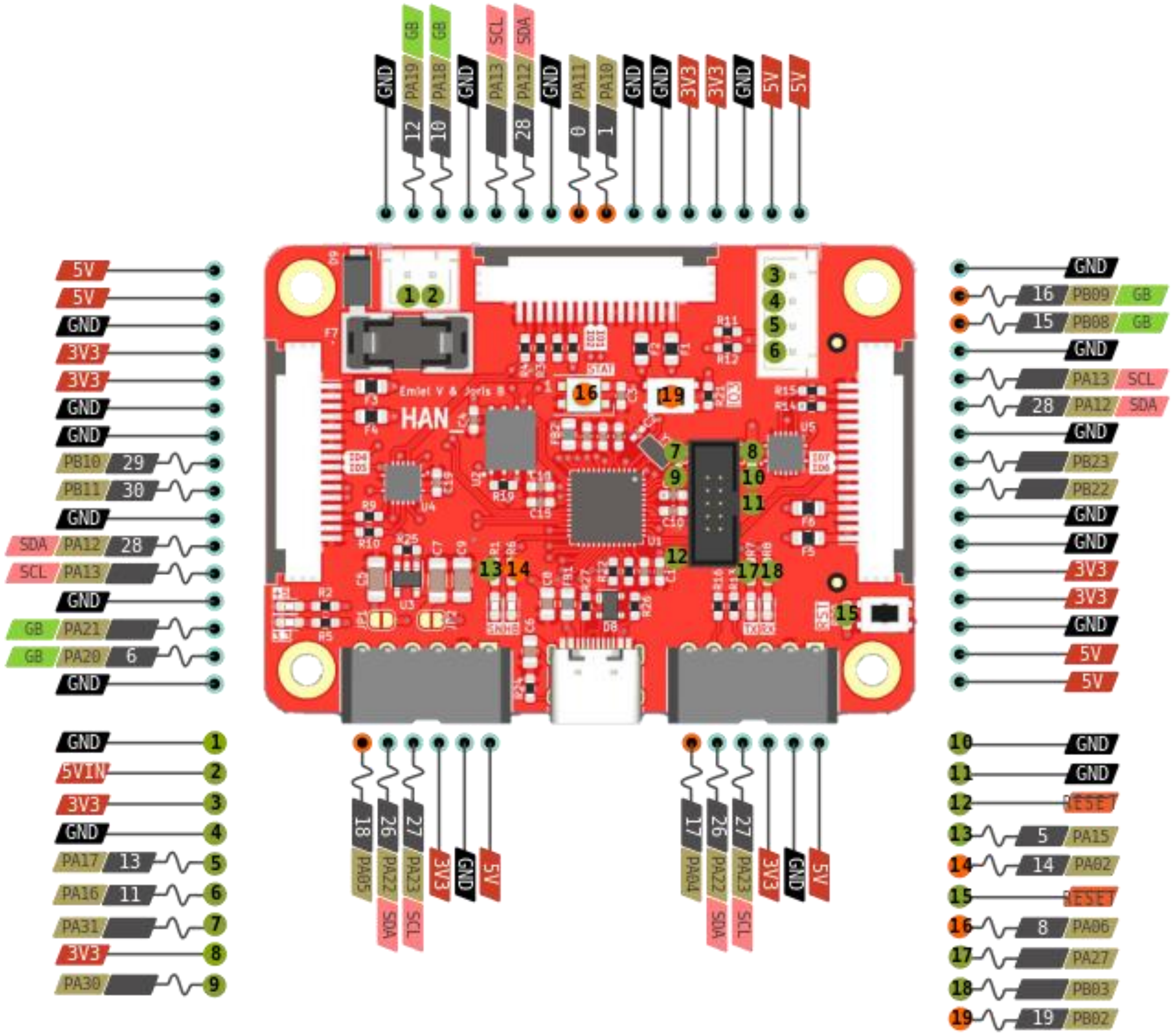


Figure 11 Pinout ledDriverBoardV3

For the full pin description go to: Appendix B – IO diagram

5.1.2 – LedDriverBoard

The complete schematic of the driver board for the flexible led strips is described here.

Main schematic:

In this schematic the microcontroller section is shown. We have added pullup resistors and test pads for all the I2C data busses. You can find them in the top left. The microcontroller can be found in the top middle section of the schematic. We went with the ATSAMD21G18A. As discussed in paragraph 2.4.3.4 of the preliminary research.

We have also added two buttons. A user programmable button and a reset button. These buttons have test pads as well.

If you look to the left, you'll find the flash chip. We chose the IS25LP032D-JLLE flash chip. The reason why we chose this flash chip can be observed in paragraph 2.4.3.3 of the preliminary research.

Below the flash chip circuit, you will find the status neopixel and the USB termination resistors. The neopixel has a decoupling capacitor. It's value can be found in the datasheet for the SK6812 led (Mouser, 2023). The USB termination resistor values have been taken from the Adafruit feather M0 express schematic (Adafruit, 2023).

Below the microcontroller you will find the mounting holes and fiducials. The mounting holes have a size of Ø3.2mm. The fiducials have no electrical purpose. They are location markings for the manufacturer.

Next to the fiducials we have placed the debug header. With this header you can debug the microcontroller and upload code or bootloaders. The pin header is a 2x5 pin header with a 1.27mm pitch.

Below that you will find the I2C bus connectors for the main bus in and out and the auxiliary connector. The Main bus in and out connectors have an extra pin connected to an interrupt enabled I/O pin. The auxiliary connector does not have this option.

Next to the auxiliary I2C connector you will find the clock circuit. You need to calculate the load capacitors for the crystal. The formula and specifications can be found in the datasheet on page 929 (Microchip, 2023) and 931 (Microchip, 2023)

The Clext in the formula on page 929 depends on your clock crystal, this value can be found in the datasheet of your clock crystal (Microchip, 2023)

The Cstray value is the capacitance of the traces going to the microcontroller and the clock circuitry in the microcontroller. This is usually around 5 or 6pF. However, the datasheet tells us that our Cstray is about 1.5pF per 12.5mm with a trace width of 0.175mm. We used a trace width of 0.25mm. The total trace length is around 8mm so a Cstray of 1.5pF should be good.

$$12.5pF = ([3.2pF + C_{LEXT}] * [3.7pF + C_{LEXT}]) / ([3.2pF + C_{LEXT} + C_{LEXT} + 3.7pF]) + 1.5pF$$

$$C_{LEXT} = 18.6pF = 18 \text{ or } 19pF$$

Note: This means the calculation and Clext (C1 and C2) in the main schematic are incorrect.

Next to the clock circuitry you will find the leds for the power indication, the TX and RX indicator leds, the heartbeat and health leds.

Above the led and clock section you will find a section with touch pads. These pads are connected to most of the IO pins and other data lines.

Above the test pad section, you will find the three FFC/FPC connectors which will connect the flexible PCBs to the mainboard. We added two I/O pins to each of the three FFC/FPC connectors. You can do whatever you want with them. We also connected an I2C bus to each of the three connectors, more on that later. The connector also has two led channels connected to it which you can use for addressable leds. Note: The led channels are 5V out!

The connectors have two 5V and two 3.3V power lines. These power lines have a 1A PPTC fuse on them. Note: Don't run all led channels at their maximum current. The fuse will blow.

The next two squares are the power and led driver sheets. See the next paragraph.

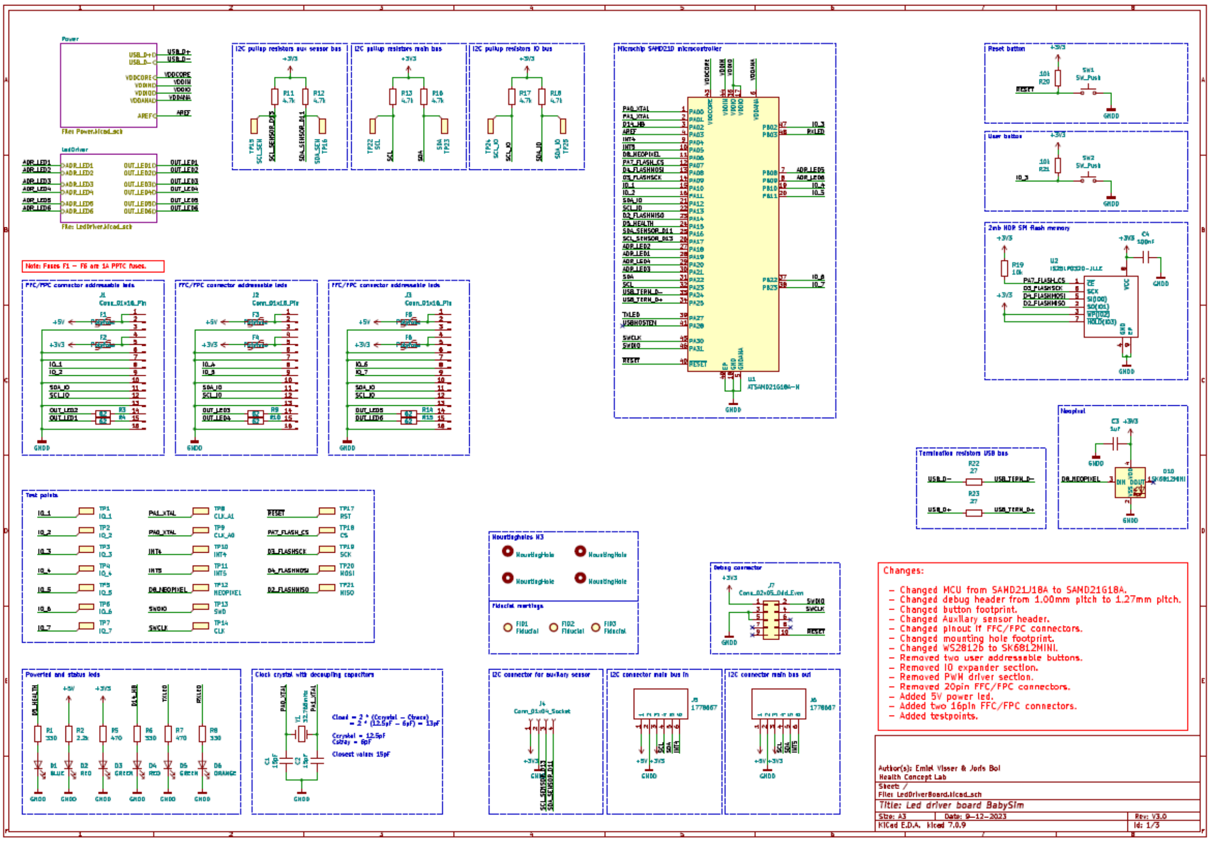


Figure 12 Main schematic

Power schematic:

We choose to go with a USB-C connector because the EU has begun to make the USB-C connector a legal requirement in portable devices. To make our design compliant with upcoming EU regulations, more futureproof, repairable, and thus more sustainable we decided to include this connector in our design. We added some filtering on the VUSB line. The incoming voltage is 5V@3A max. Next to the USB port circuitry you will find the ESD protection diode for the USB voltage and data lines. We also added two test points to the USB data + and USB data lines.

In this schematic you will also be able to find the decoupling capacitors for the microcontroller.

The auxiliary in connector has a 3A fast blow fuse. You can connect an external 5V power source to this connector. There is another test point connected to the connector.

The USB power lines, and Auxiliary power lines connect to the AP2112K-3.3 low dropout regulator (LDO) this LDO converts the 5V voltage to 3.3V. Be aware that the maximum output current of the LDO is about 600mA. We have added two more test points. One for 5V and one for 3.3V.

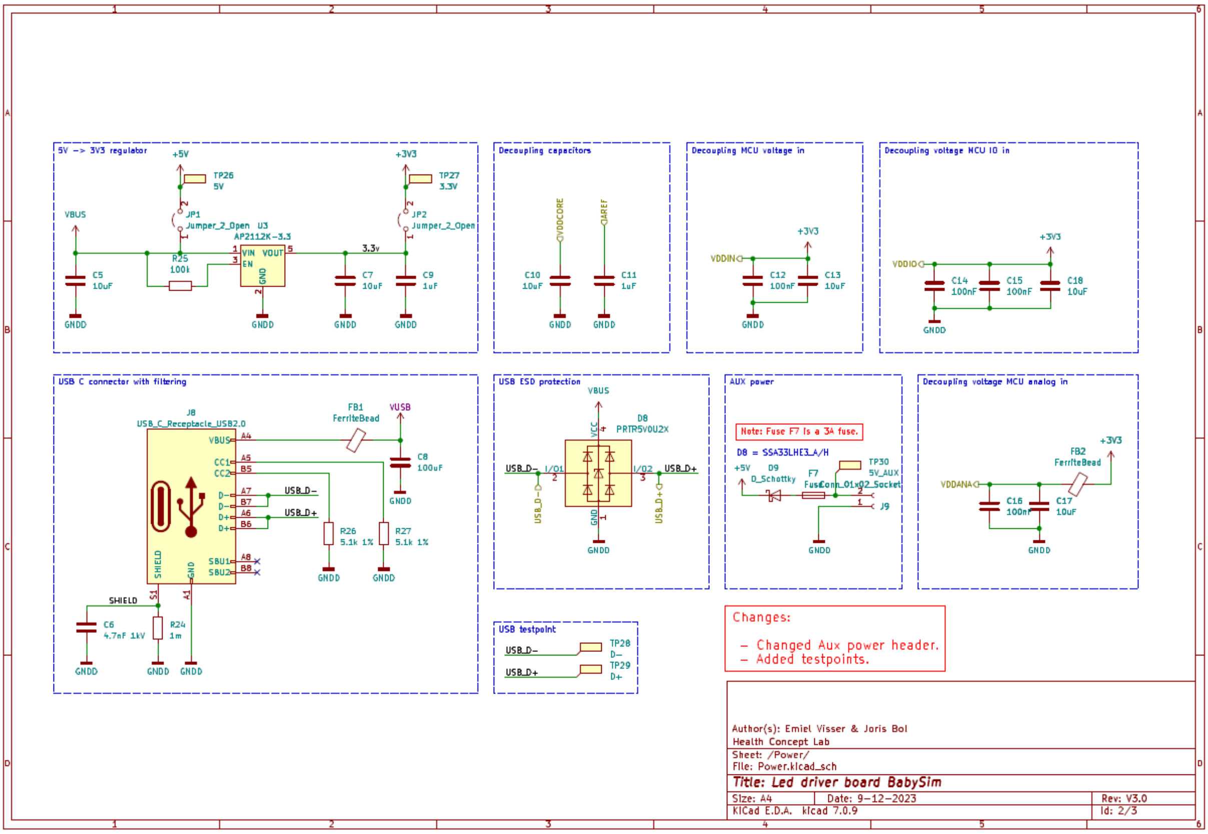


Figure 13 Power schematic

Led driver schematic:

In this schematic you will find the gate buffers (level shifters if you will) for converting the 3.3V led data signal to 5V. We have used the SN74AHCT125RYGR chip for this purpose. If you would like to know why we chose this particular chip. Have a look at our paragraph [2.4.3.2](#) of the preliminary research.

All input channels and output channels have test points on them just in case you need them.

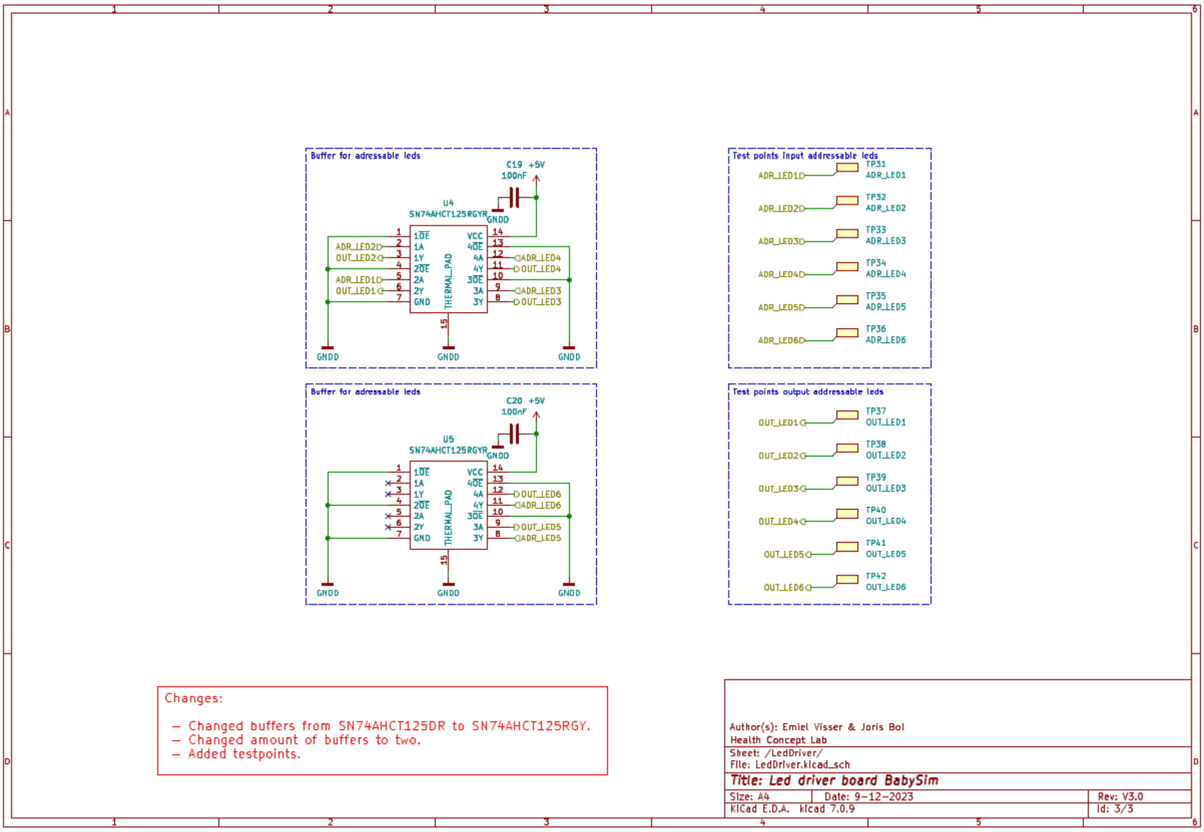


Figure 14 Led driver schematic.

LedDriver PCB:

Creating the layout for a PCB is always one of the hardest parts of creating a PCB design. We began by creating subgroups of components. After that we started placing the components in the subgroups in the correct orientation. We began by placing them in the PCB area. By doing this you quickly get a sense of scale.

After we had done this, we started routing the subgroups. Then we improved the component placement and updated the placement of some of the components in the subgroups. After everything was routed, we started adding the labels and logos.

For the PCB stack up we decided to go with a four-layer design. By doing this we gain a lot more room to route te traces. Going with this particular layer stack up also gives us more signal stability.

The stack up is as follows:

Layer:	Net name:
F.Cu (front)	Signal/Ground.
In1.Cu	Ground.
In2.Cu	Power 3.3V/5V.
B.Cu (Back)	Signal/Ground.

We tried to make the In1 and In2 layers a power and ground only layer, but we ended up having to route a couple of traces on these layers as well.

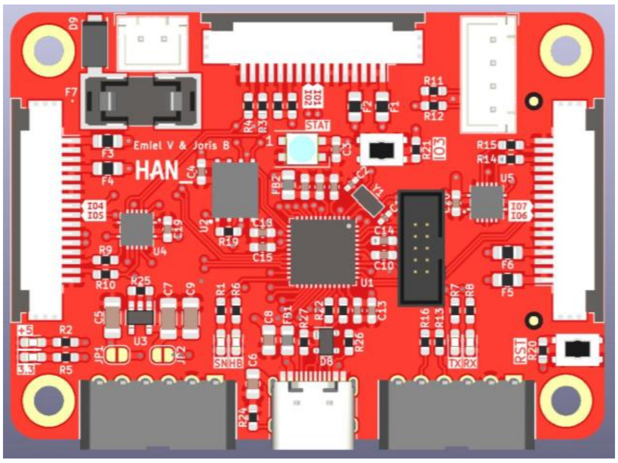


Figure 16 Driver board front

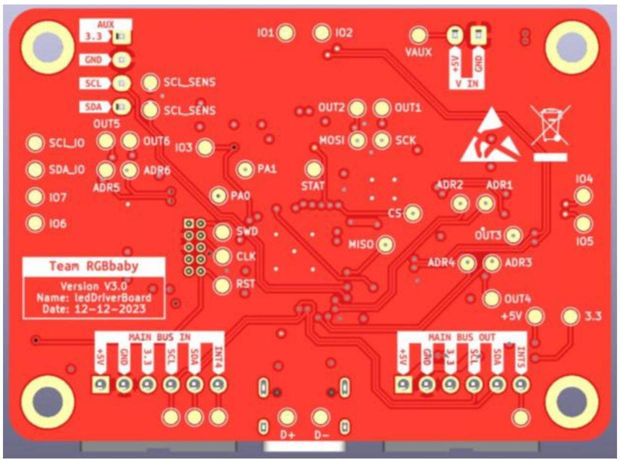


Figure 15 Driver board back

5.1.3 – Flexible PCB

This paragraph describes the function and considerations made during the design and assembly of the flexible PCB. This PCB is responsible for the colour changing capabilities of the skin we developed. It also contains the capability to sense touch, using a capacitive touch chip and pads. This particular part of the PCB is still in its testing face and has not been proven.

The flexible PCB consists of four main parts:

- The connector.
- The leds.
- The capacitive touch chip.
- Touch pads (which are not visible on the figure below.)

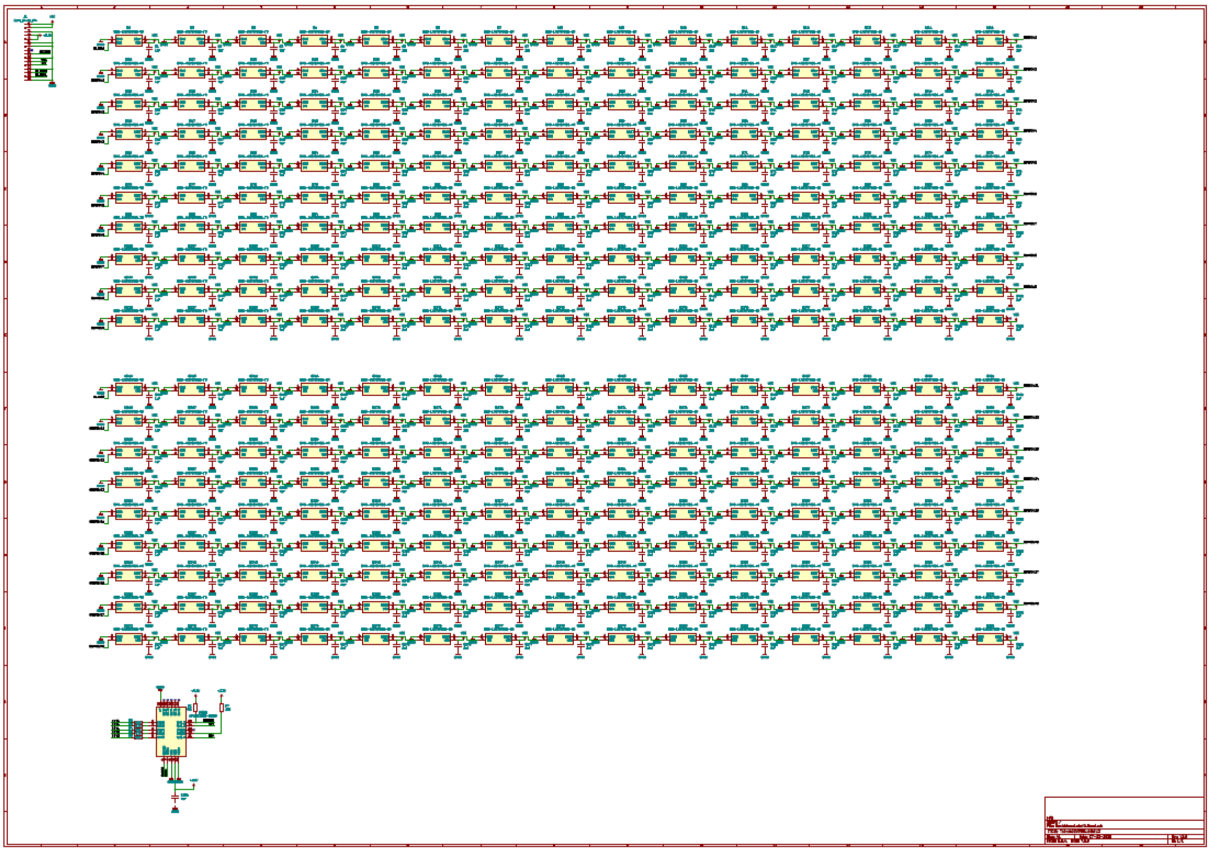


Figure 17 Main schematic

The connector:

The connector of the flexible PCB is not a component but rather the PCB itself. It must fit in a FFC connector, which is present on the driver board. To successfully make a piece of the PCB function as a connector we had to add traces in the correct place, and we needed to add a stiffener, so the connector does not bend or rip. The measurements and connection are depicted in the figures below.

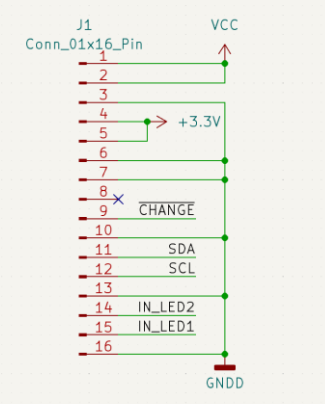


Figure 18 Connector schematic

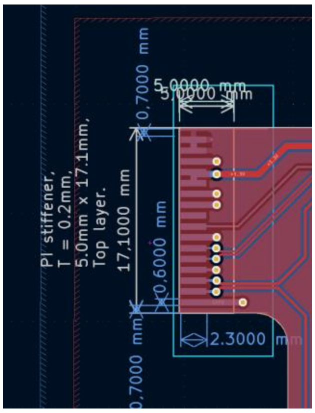


Figure 19 Connector layout

The leds:

As sufficiently explained in paragraph 2.6.2, we choose the SMDLX0707RGBTR leds. these are present on the PCB in great number. The led grid contains 15 X 19 leds. this totals 285 leds. they are arranged in such a way that the leds are spaced 3,9mm centre to centre. This came out of our tests as a small enough gap, so the leds don't cause spotting.

The leds are connected in such a way that it snakes down, starting at the left. The first 150 leds are driven by one IO pin and the other leds by another. This is because we were worried that driving all 285 leds with one io pin would exceed the maximum amount of ram we have available.

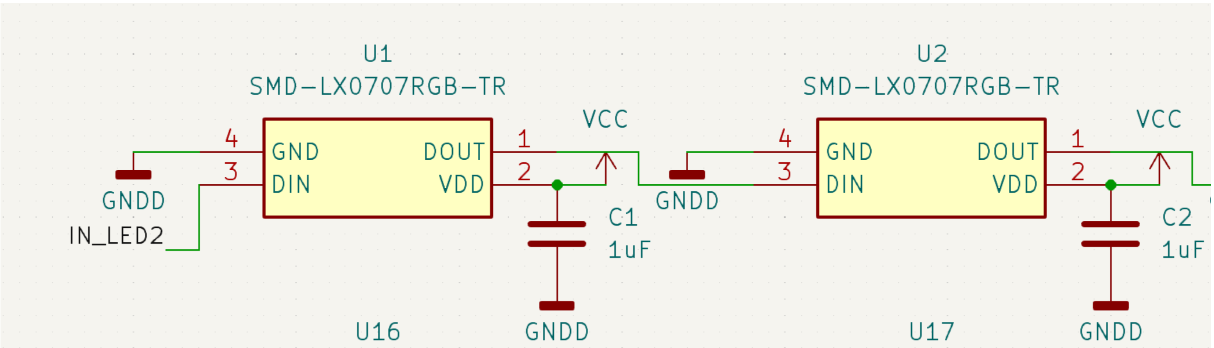


Figure 20 leds schematic

The capacitive touch chip:

The capacitive touch chip, which is the AT42QT2120-MMHR, is capable of sensing touch via the capacitive properties of touch. Have a look at paragraph 2.4.2 in the preliminary research to find our requirements for the touch sensor.

It has the capabilities to run in two modes: standalone mode and comms mode. We use comms mode because we want to be able to communicate with the chip via I2C. This enables us to read the state of registers rather than relay on the multiple out channels of the chip.

We use six pins for detecting touch. Five of the pins function as a key. These keys are capable of sensing direct touch, consequently, you need to put your finger directly on the pads. The other pin is something called a prox key. This pin – in contrast of the other keys – is capable of sensing distance to your finger.

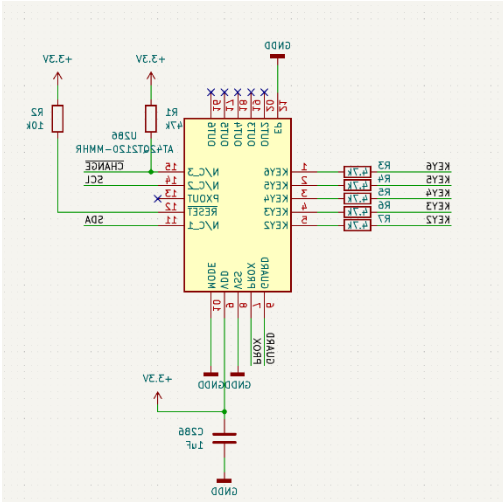


Figure 21 Capacitive touch chip

There is also a guard pin, which is responsible for protecting the pads for interference, more about this subject can be found in the next paragraph.

Touch pads:

The pads are responsible for supplying a surface with which you can interact. As previously mentioned, there are six pads. Five of which are keys, and one is a prox pad.

It is important to sufficiently protect the pads of interference, because of its fragile nature. We do this by using the guard channel and a ground plain.

The guard channel combats the noise that is present on the pads by putting the reverse of the interference on the guard channel. The guard channel raps around the pads so it can adequately protect them.

The ground plain is at the back side of the pads. For the pads to work correctly, there must be a precise reference point, from which the chip can derive the capacitive difference. A perfect 0V is ideal for this, hence the ground plain.

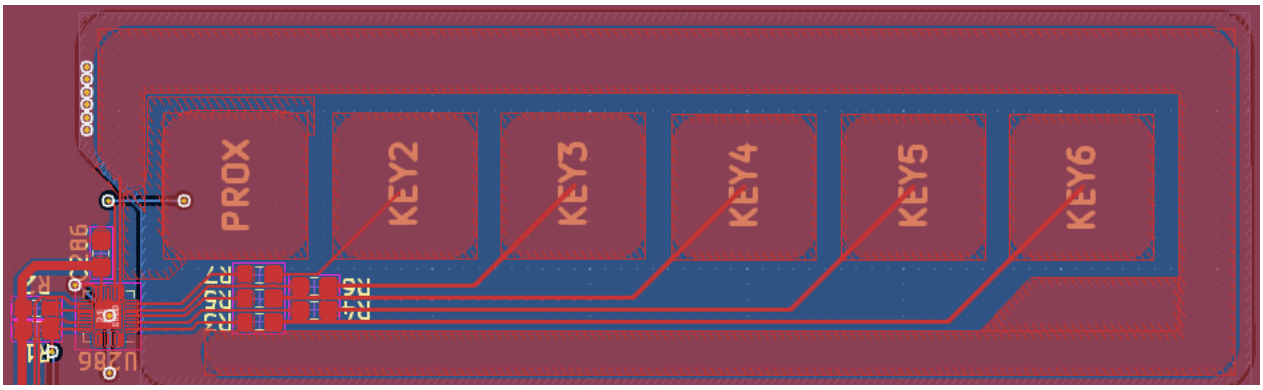


Figure 22 pad layout

6 – Testing

In this chapter we will describe the testing process and our findings. This will be done with the help of a testing method which will be described in its respective paragraph.

6.1 – Functional and physical tests

The Action/input variable of the test are determent by the functional an technical specifications.

<table>
<tr><td colspan="3">Project name: RGBbaby</td><td colspan="2">Date: -</td></tr>
<tr><td colspan="2">Test case ID: 001</td><td colspan="3">Name: -</td></tr>
<tr><td colspan="5">Function/module/program under test:

Main PCB</td></tr>
<tr><td>Test</td><td>Action/input</td><td>Expected result</td><td>Pass/Fail</td><td>Actual result</td></tr>
<tr><td>1.</td><td>The PCB turns on.</td><td>The PCB can successfully turn on.</td><td>pass</td><td>-</td></tr>
<tr><td>2.</td><td>3V3 is available.</td><td>When 3V3 is available the corresponding led turns on.</td><td>pass</td><td>-</td></tr>
<tr><td>3.</td><td>5V is available.</td><td>When 5V is available the corresponding led turns on.</td><td>pass</td><td>-</td></tr>
<tr><td>4.</td><td>The heartbeat led is used.</td><td>When the PCB is on the heartbeat led turns on and of with the frequency of 1Hz</td><td>-</td><td>-</td></tr>
<tr><td>5.</td><td>The serial leds are used.</td><td>The serial leds turn on when there is serial communication.</td><td>-</td><td>-</td></tr>
<tr><td>6.</td><td>The sense led is used.</td><td>The sense led turns on when a button is used ore when the skin senses touch.</td><td>-</td><td>-</td></tr>
<tr><td>7.</td><td>The RGB led is used.</td><td>The RGB led is capable of simulating the following behaver:

<table><tr><th>Colour:</th><th>State:</th></tr><tr><td>Off</td><td>Nothing is wrong, PCB is off.</td></tr><tr><td>White</td><td>The PCB went into hard fault.</td></tr><tr><td>Red</td><td>There is no connection</td></tr></table></td><td>-</td><td>-</td></tr>
</table>

		between the PCB and the main board. **Green** INIT finished successfully. **Blue** The PCB is in test mode.		
8.	The PCB uses the main I2C bus.	The PCB is capable to use the main I2C bus to read and write data.	-	-
9.	The PCB can generates the data signal for the addressable leds.	The PCB can successfully generate the data signal for the addressable leds.	-	-
10.	The PCB supplies the led strip with sufficient power.	The PCB can successfully supply 5V and 3V3 to the led strip.	-	-
11.	The PCB acts upon the input of the led strip.	The PCB is capable of acting upon the input of the led strip, which is generated by the sense chip.	-	-

Project name: RGBbaby	**Date:** -
Test case ID: 002	**Name: -**
Function/module/program under test: **Flex PCB**	

Test	**Action/input**	**Expected result**	**Pass/Fail**	**Actual result**
1.	The leds make the skin turn yellow.	The skin turns a realistic yellow, without becoming a light beacon.	-	-
2.	The leds make the skin turn red.	The skin turns a realistic red, without becoming a light beacon.	-	-
3.	The leds make the skin turn dark blue.	The skin turns a realistic dark blue, without becoming a light beacon.	-	-
4.	The leds make the skin turn pale.	The skin turns a realistic pale, without becoming a light beacon.	-	-
5.	One of the touch pads is activated by a user.	We can detect and use the input of the user.	-	-

6.	The proximity pad is activated by a user.	We can detect and measure the user input.	-	-

It may be apparent that the test are incomplete and lack conclusions. This is because of the technical issues we encountered. These issues are further described in the conclusion of this rapport.

6.2 – Software tests

Due to the technical complications mentioned in the conclusion. We were not able to supply a sufficient software testing plan and its corresponding conclusions.

7 – Conclusion

Our goal was to create a solution for inducing a colour change and touch in the skin of a baby CPR doll. Our objectives included making a test setup for the components and solutions we wanted to test, writing code for controlling the driver board and documenting our findings.

7.1 – Achievements

Testing:
We have done a lot of research on led driving and touch sensing options. Our findings can be reused for future projects.

Hardware:
We have created a test PCB that allowed us to test different hardware and software configurations. With the lessons we learned from that test PCB we created our final PCB. This PCB has six led channels capable of driving leds up to 1A per connector. Each connector has an I2C connection so that a capacitive touch sensor can be used.

Software:
We have written a software library that makes the implementation of the led channels, IO and touch sensor a lot easier.

7.2 – Challenges

We have encountered a lot of hurdles, slowing us down. This is the reason why we haven't been able to finish the project completely.

Microcontroller:
The microcontroller we had chosen for our first test board had no Arduino implementation in our configuration which meant we weren't able to use every functionality we wanted to try out.

Clock:
On both boards we have/had problems with the onboard clock. The clock crystal needs to put out a clock frequency that is within the 32.768kHz range. However, it refused to do that on both boards which wouldn't allow the code to be run using the Arduino bootloader because the Arduino bootloader uses the external clock instead of the internal clock circuitry. On our test board we manged to fix this by changing the load capacitors to a different value. However, this wouldn't work on the final version.

Programming:
Because the clocks wouldn't work we had to program in embedded C/C++. Creating libraries for the different chips is very time consuming which is what brought us into trouble. We were also unable to find a suitable Arduino bootloader for the test PCB which meant we couldn't use a lot of the functions we wanted to test.

We also wired up the debug header to the final PCB incorrectly which meant we had to solder jumper wires to pads on the back to be able to program it.

Shorts:
We also had a number of shorts on both boards and on our led strip due to an excess amount of solder paste resulting in bridges, which took a lot of time to solve those.

7.3 – Recommendations

Clock:

For the final PCB to function properly, your initial priority should be getting the clock up and running. Once that's sorted, you should be able to upload the Arduino code through the USB port. Establishing a stable clock is crucial as it forms the foundational step toward ensuring the functionality of the final PCB.

Fix short:

If you want to use the flexible led strip you will need to fix the short on 5V first.

Test code:

Because of the clocking issue and the shortage of time, we haven't had the chance to test the code we wrote for the LEDs and the touch sensor.

Debug header pinout:

If you find yourself in the process of designing a new flexible PCB, it would be advisable to check and alter the pinout arrangement of the debug header on the final PCB. Taking this step could prove beneficial in optimizing the functionality and compatibility of the overall design.

7.4 – Final thoughts

In conclusion, our technical project focused on enhancing the functionality and realism of a baby CPR doll by incorporating a capacitive touch sensors and a subdermal lighting system. The integration of these features not only elevates the overall user experience but also addresses the critical need for realistic training simulations in infant CPR scenarios. The tactile feedback provided by the capacitive touch sensors, combined with the subdermal lighting system that simulates physiological responses, creates a training environment that closely mirrors real-life situations. This innovation holds great promise in advancing the effectiveness of medical training, ensuring that practitioners develop the necessary skills and confidence to respond to emergencies involving infants. As we reflect on the journey of this project, we take pride in contributing to the advancement of healthcare education and ultimately, the improvement of infant CPR practices.

8 – Personal evaluation

In this paragraph, we will share our individual perspectives and insights regarding the project.

8.1 – Emiel Visser

Being a part of the team responsible for creating the PCBs was a great learning opportunity for me, and I genuinely enjoyed the experience. Despite the challenges that came with having a significant role in the project, it was a bit stressful at times. However, overcoming those challenges and contributing to the team effort added a sense of accomplishment that made the whole journey worthwhile.

Personal learning goals:
In setting my personal learning goals for this semester, I aimed to achieve a better work-life balance by reducing the time dedicated to school-related tasks. Last year, I found myself putting in 70-hour weeks, and I'm pleased to report that this semester, I successfully managed to cut it down to 50 hours per week. It's a significant improvement, and I consider this goal accomplished.
Another objective was to enhance my skills in PCB design. While I did make notable progress and acquired valuable insights, I acknowledge that there is room for further development in this area. I'm satisfied with what I've learned so far, but I'm eager to delve even deeper into PCB design in the future.

Contributions:
Throughout the project, I assumed the responsibility of designing both the test PCB and the final PCB. Additionally, I took a lead role in developing the majority of the software for the final PCB. As a natural extension of these tasks, I also undertook the responsibility of crafting comprehensive documentation for both the test PCB and the final PCB, capturing the intricacies of the design and software aspects. This holistic involvement allowed me to contribute not only to the technical implementation but also to the clarity and thoroughness of the project documentation.

Conclusions:
I must admit that I significantly underestimated the scope and complexity of this project. However, as I immersed myself in the tasks and challenges it presented, I found immense satisfaction and fulfilment in the work. The journey not only exceeded my expectations but also became a source of profound learning. The diverse experiences encountered throughout the project provided valuable insights, and overcoming the initial underestimation became a pivotal aspect of my personal and professional growth.

8.2 – Joris Bol

This project helped me to become more of a team player. This multidisciplinary project helped me develop my communication skills better and made me more aware of how other people function. I found this a most enjoyable experience, besides the work it sometimes created. I think of this project as overall a positive encounter.

Personal learning goals:
The goals that I want to achieve this semester were to introduce myself to PCB design and to get better at English report writing. For both of these goals there wase tremendous opportunity withing this project. I was able to get acquainted wit ECAD software and because of the main language within this project, I was forced to exercise my English skills. I would like to say that I was more than moderately successful, improving my English skills substantially and going beyond just the beginning of ECAD.

Contributions:
Due to my personal goal of learning the basics of ECAD we created an opportunity for me to develop three PCBs. Because of my involvement with the development of these PCBs and the debugging there of I was able to succeed in my second learning goal: writing documentation. Documentation and research were one of the biggest parts of this project, making up for at least 50% of the time I spend on this project. A large part of documentation was making it so that future projects had something to build upon, and I like to think that I was of help in that matter.

Conclusions:
This project is something I always wanted to do. From an early age I knew I wanted to do something withing the health sector in combination with technology. The assignment we did fell perfectly within these interests. I must admit that I found this project harder than I anticipated, this was mostly caused by the uncertainty of what to do in the beginning and my inexperience with some of the technologies we used. I did however find it an enjoyable project from which I learned a lot.

Appendix A – Document history

Preliminary research – ESE

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 conclusion	J. Bol

Preliminary research – IPO

No history available

Functional design

Version	Date	Change	Name
V1.0	5-9-2023	Initial setup	J. Bol
V1.0	9-9-2023	Added information	E. Visser
V1.0	11-9-2023	Added information	Everybody
V1.0	12-9-2023	Added information	J. Bol & E. Visser
V1.0	26-9-2023	Added UML.	J. Bol
V1.0	1-10-2023	Spellcheck.	E. Visser
V1.0	12-12-2023	Added changes.	E. Visser

Technical design

Version:	Date:	Change:	Name:
V1.0	5-9-2023	Creation document.	J. Bol
V1.0	17-9-2023	Added paragraphs.	E. Visser
V1.0	16-10-2023	Added info.	E. Visser
V1.0	27-11-2023	Added info.	E. Visser
V1.0	1-12-2023	Added info.	E. Visser
V1.0	9-12-2023	Added info.	E. Visser
V1.0	10-12-2023	Added info.	E. Visser
V1.0	18-12-2023	Added info.	J. Bol
V1.0	16-1-2024	Spellcheck.	E. Visser

Manual:

Version:	Date:	Change:	Name:
V1.0	11-9-2023	Creation document, preface, C1, C2 & C3.	J. Bol
V1.0	12-11-2023	Added programming environment tab.	E. Visser
V1.0	20-11-2023	Added spectrometer tab.	J. Bol
V1.0	04-01-2024	Added more information in the hardware section.	E. Visser
V1.0	05-01-2024	Added more information in the hardware section.	E. Visser
V1.0	06-01-2024	Added more information in the hardware section.	E. Visser
V1.0	07-01-2024	Added info in the hardware and debugging section.	E. Visser
V1.0	09-01-2024	Added bibliography, performed spellcheck.	E. Visser

Appendix B – IO diagram

Microcontroller:	Pin on microcontroller:	Schematic name:	Function:
ATSAMD21G18A-M	PA00	PA0_XTAL	Oscillator pin 0
ATSAMD21G18A-M	PA01	PA1_XTAL	Oscillator pin 1
ATSAMD21G18A-M	PA02	D14_HB	Heartbeat led
ATSAMD21G18A-M	PA04	INT4	Interrupt main bus in
ATSAMD21G18A-M	PA05	INT5	Interrupt main bus out
ATSAMD21G18A-M	PA06	D8_NEOPIXEL	Status neopixel led
ATSAMD21G18A-M	PA07	D4_FLASH_CS	Chip select pin flash chip
ATSAMD21G18A-M	PA08	D4_FLASHMOSI	Mosi pin flash chip
ATSAMD21G18A-M	PA09	D3_FLASHSCK	Clock pin flash chip
ATSAMD21G18A-M	PA10	IO_1	General purpose input/output pin FFC connector
ATSAMD21G18A-M	PA11	IO_2	General purpose input/output pin FFC connector
ATSAMD21G18A-M	PA12	SDA_IO	I2C datapin FFC connector
ATSAMD21G18A-M	PA13	SCL_IO	I2C datapin FFC connector
ATSAMD21G18A-M	PA14	D2_FLASHMISO	Miso pin flash chip
ATSAMD21G18A-M	PA15	D5_HEALTH	Health led
ATSAMD21G18A-M	PA16	SDA_SENSOR_D11	I2C datapin auxiliary sensor
ATSAMD21G18A-M	PA17	SCL_SENSOR_D13	I2C datapin auxiliary sensor
ATSAMD21G18A-M	PA18	ADR_LED2	Addressable led channel 2
ATSAMD21G18A-M	PA19	ADR_LED1	Addressable led channel 1
ATSAMD21G18A-M	PA20	ADR_LED4	Addressable led channel 4
ATSAMD21G18A-M	PA21	ADR_LED3	Addressable led channel 3
ATSAMD21G18A-M	PA22	SDA	I2C datapin main bus
ATSAMD21G18A-M	PA23	SCL	I2C datapin main bus
ATSAMD21G18A-M	PA24	USB_TERM_D-	USB data minus
ATSAMD21G18A-M	PA25	USB_TERM_D+	USB data plus
ATSAMD21G18A-M	PA27	TXLED	TX led
ATSAMD21G18A-M	PA28	USBHOSTEN	USB hoste enable (NOT CONNECTED)
ATSAMD21G18A-M	PA30	SWCLK	Serial Wire Debug clock
ATSAMD21G18A-M	PA31	SWDIO	Serial Wire Debug Data
ATSAMD21G18A-M	RESET	RESET	Microcontroller reset
ATSAMD21G18A-M	PB02	IO_3	General purpose input/output connected to button
ATSAMD21G18A-M	PB03	RXLED	TX led
ATSAMD21G18A-M	PB08	ADR_LED5	Addressable led channel 5
ATSAMD21G18A-M	PB09	ADR_LED6	Addressable led channel 6
ATSAMD21G18A-M	PB10	IO_4	General purpose input/output pin FFC connector
ATSAMD21G18A-M	PB11	IO_5	General purpose input/output pin FFC connector
ATSAMD21G18A-M	PB22	IO_6	General purpose input/output pin FFC connector
ATSAMD21G18A-M	PB23	IO_7	General purpose input/output pin FFC connector

Bibliography

Adafruit. (2023). *Downloads.* Retrieved from learn.adafruit: https://learn.adafruit.com/adafruit-feather-m0-express-designed-for-circuit-python-circuitpython/downloads

Atmel. (2023). *Atmel start.* Retrieved from strart.atmel: https://start.atmel.com/

Cypress. (2023). *S25FL116K/S25FL132K/S25FL164.* Retrieved from infineon: https://www.infineon.com/dgdl/Infineon-S25FL116K_S25FL132K_S25FL164K_16_MBIT_(2_MBYTE)_32_MBIT_(4_MBYTE)_64_MBIT_(8_MBYTE)_3.0_V_SPI_FLASH_MEMORY-DataSheet-v10_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ed4ebee537f

DigiKey. (2023). *2686.* Retrieved from digikey: https://www.digikey.nl/en/products/detail/adafruit-industries-llc/2686/5804107?s=N4IgTCBcDaIMoGkBsAOAjGAsgSQHLZAF0BfIA

DigiKey. (2023). *AT42QT2120-MMHR.* Retrieved from digikey: https://www.digikey.nl/en/products/detail/microchip-technology/AT42QT2120-MMHR/3678702

DigiKey. (2023). *MPR121QR2.* Retrieved from digikey: https://www.digikey.nl/nl/products/detail/nxp-usa-inc/MPR121QR2/2186527

Distrelec. (2023). *ILPL-K5001-RGBW-SK105-01.* Retrieved from distrelec: https://www.distrelec.be/nl/smd-led-5050-rgbw-460nm-505nm-615nm-intelligent-led-solutions-ilpl-k501-rgbw-sk105-01/p/30308082

Distrelec. (2023). *RND 135-00252.* Retrieved from distrelec: https://www.distrelec.nl/nl/smd-led-rgb-plcc-468-nm-520-nm-632-nm-rnd-components-rnd-135-00252/p/30158847?cq_src=google_ads&cq_cmp=18200849884&cq_con=&cq_term=&cq_med=pla&cq_plac=&cq_net=x&cq_pos=&cq_plt=gp&gclid=CjwKCAjwo9unBhBTEiwAipC118qLKgWXeMIWCFFvFx

Microchip. (2023). *Low-Power, 32-bit Cortex-M0+ MCU with Advanced Analog.* Retrieved from microchip: https://ww1.microchip.com/downloads/aemDocuments/documents/MCU32/ProductDocuments/DataSheets/SAM-D21-DA1-Family-Data-Sheet-DS40001882H.pdf

Mouser. (2023). *19-337/R6GHBHC-A01/2T .* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Everlight/19-337-R6GHBHC-A01-2T?qs=8cKuZ6Ok2lbHoW%2FFyZa1pA%3D%3D

Mouser. (2023). *AT25QL641-UUE-T .* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Renesas-Dialog/AT25QL641-UUE-T?qs=IS%252B4QmGtzzpVKdwhx%252BU4oA%3D%3D

Mouser. (2023). *ATSAMD21G18A-AU.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Microchip-Technology/ATSAMD21G18A-AU?qs=KLFHFgXTQiDh1vL0VLU76Q%3D%3D

Mouser. (2023). *ATSAMD21G18A-MU.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Microchip-Technology/ATSAMD21G18A-MU?qs=KLFHFgXTQiA%2FfMu%2FGM5u1A%3D%3D

Mouser. (2023). *ATSAMD21J18A-AU.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Microchip-Technology/ATSAMD21J18A-AU?qs=KLFHFgXTQiBwjHCNgKVs9A%3D%3D

Mouser. (2023). *ATSAMD21J18A-CU.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Microchip-Technology/ATSAMD21J18A-CU?qs=9KdFJXLqUo8%252BXJ5Hms4agg%3D%3D

Mouser. (2023). *ATSAMD21J18A-MU.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Microchip-Technology/ATSAMD21J18A-MU?qs=KLFHFgXTQiA5nYxZpqmjnQ%3D%3D

Mouser. (2023). *CLQ6B-TKW-S1L1R1H1TBB7935CC3.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Cree-LED/CLQ6B-TKW-S1L1R1H1TBB7935CC3?qs=Rp5uXu7WBW8kseQbCcO3Lg%3D%3D

Mouser. (2023). *IN-P55QDTRGBW.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Inolux/IN-P55QDTRGBW?qs=stqOd1AaK78tLn2kfF8cBA%3D%3D

Mouser. (2023). *IS25LP032D-JLLE.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25LP032D-JLLE?qs=DXv0QSHKF4wnam7hnzWEbg%3D%3D

Mouser. (2023). *IS25LP040E-JYLE-TR .* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25LP040E-JYLE-TR?qs=eP2BKZSCXI5j9mdUcQ6FYg%3D%3D

Mouser. (2023). *IS25LP064A-JLLE.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25LP064A-JLLE?qs=ULk7ZlyDR4f8APOHOBVPpg%3D%3D

Mouser. (2023). *IS25LP064D-JBLA3.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25LP064D-JBLA3?qs=uwxL4vQweFM2ST8t4VRbqQ%3D%3D

Mouser. (2023). *IS25LP064D-JKLE .* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25LP064D-JKLE?qs=xZ%2FP%252Ba9zWqaUbrcT%2FGTrkA%3D%3D

Mouser. (2023). *IS25LP080D-JULE-TR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25LP080D-JULE-TR?qs=%252BEew9%252B0nqrBxcCxQG%252BlQOg%3D%3D

Mouser. (2023). *IS25WP032D-JKLE.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25WP032D-JKLE?qs=DXv0QSHKF4zsRyVNAdaZLg%3D%3D

Mouser. (2023). *IS25WQ040-JNLE.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ISSI/IS25WQ040-JNLE?qs=TO8kDJPmCBRusxfzIU0lYg%3D%3D

Mouser. (2023). *PCA8574D,518.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/NXP-Semiconductors/PCA8574D518?qs=LOCUfHb8d9tI6M3Msz5kkw%3D%3D

Mouser. (2023). *PCA8575BS,118 .* Retrieved from mouser: https://nl.mouser.com/ProductDetail/NXP-Semiconductors/PCA8575BS118?qs=LOCUfHb8d9uphqANyNY67Q%3D%3D

Mouser. (2023). *PCA9536DR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Texas-Instruments/PCA9536DR?qs=aEuGZpxfbxUrMrsVTt7%252BhA%3D%3D

Mouser. (2023). *PCA9548ABS,118.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/NXP-Semiconductors/PCA9548ABS118?qs=LOCUfHb8d9uF0VYCksl9OA%3D%3D

Mouser. (2023). *PCA9670BS,118.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/NXP-Semiconductors/PCA9670BS118?qs=LOCUfHb8d9vyec%2F1PtVhAw%3D%3D

Mouser. (2023). *PCA9685BS,118.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/NXP-Semiconductors/PCA9685BS118?qs=beN0Cyoe8Yle51zseU5uxw%3D%3D

Mouser. (2023). *PCAL9722HNMP.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/NXP-Semiconductors/PCAL9722HNMP?qs=ulEaXIWI0c%2F4GQ1orHpXww%3D%3D

Mouser. (2023). *SC-32S.* Retrieved from mouser: https://nl.mouser.com/datasheet/2/360/file_PRODUCT_MASTER_50812_GRAPHIC03-542305.pdf

Mouser. (2023). *SK6812MINI.* Retrieved from mosuer: https://www.mouser.com/datasheet/2/737/SK6812MINI_REV_01_1_2-1915201.pdf

Mouser. (2023). *SMD-LX0707RGB-TR .* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Lumex/SMD-LX0707RGB-TR?qs=u16ybLDytRYfAsx%252BT62puQ%3D%3D

Mouser. (2023). *SMLP34RGBN1W3.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/ROHM-Semiconductor/SMLP34RGBN1W3?qs=Wj%2FVkw3K%252BMB6e39pO3qaCw%3D%3D

Mouser. (2023). *SN74AHCT125DR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Texas-Instruments/SN74AHCT125DR?qs=B7lSdIxWQkm9VQkJOjfRaA%3D%3D

Mouser. (2023). *SN74AHCT125RGYR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Texas-Instruments/SN74AHCT125RGYR?qs=WzgTT80quPG58uovxcQWdQ%3D%3D

Mouser. (2023). *TL3342F260QG.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/E-Switch/TL3342F260QG?qs=6C6BR4UgC3Of90YOH%2FEXyw%3D%3D&mgh=1&gclid=CjwKCAjwo9unBhBTEiwAipC11_SAAIPyCmNt-0Y2xMbigZ8T0VGch7C7V15uZdR1J16C5-VcodVioBoC14IQAvD_BwE

Mouser. (2023). *TLC59401RHBR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Texas-Instruments/TLC59401RHBR?qs=%2Fqzd9s%252BcLd5jLWxG9iB8xw%3D%3D

Mouser. (2023). *TLC5940RHBR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Texas-Instruments/TLC5940RHBR?qs=p6lVfQR1GSql0QiKiHXw6g%3D%3D

Mouser. (2023). *TLC5947RHBR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Texas-Instruments/TLC5947RHBR?qs=XGzIaZb%2FFYKg0KxHbVWJ0A%3D%3D

Mouser. (2023). *TLC59711PWPR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/Texas-Instruments/TLC59711PWPR?qs=0gJBXdIUCJmwn8liNLSqqA%3D%3D

Mouser. (2023). *TS17-48-05-YE-260-SMT-TR.* Retrieved from mouser: https://nl.mouser.com/ProductDetail/CUI-Devices/TS17-48-05-YE-260-SMT-TR?qs=tlsG%2FOw5FFgMrmlTrPhqZw%3D%3D

OpenAI. (2023). *ChatGPT.* Retrieved from chat.openai: https://chat.openai.com/

Reichelt. (2023). *WUE 150141M1731.* Retrieved from reichelt: https://www.reichelt.nl/nl/nl/rgb-led-smd-3528-rood-groen-blauw-270-950-230-mcd-120--wue-150141m1731-p232068.html?PROVID=2809&gclid=CjwKCAjwo9unBhBTEiwAipC117XO2lOvVa-Djn6QAyeZty87x3jeCeQMU89jCTVnrXIqnveYX9ChUxoCqtcQAvD_BwE

TME. (2023). *KAA-3528RGBS-11.* Retrieved from tme: https://www.tme.eu/nl/details/kaa-3528rgbs-11/led-diodes-smd-gekleurd/kingbright-electronic/?brutto=1&currency=EUR&gclid=CjwKCAjwo9unBhBTEiwAipC113EiGMlFlGeXmuBZfEKRWcfQH-Kf50QJdRfDlysYgNVbN9WMtGjeNRoC0iIQAvD_BwE

TME. (2023). *WS2812B BLACK.* Retrieved from tme: https://www.tme.eu/nl/details/ws2812b-b/led-diodes-smd-gekleurd/worldsemi/ws2812b-black/

Wright, L. (2023). *PySpectrometerV2.* Retrieved from github: https://github.com/leswright1977/PySpectrometer2