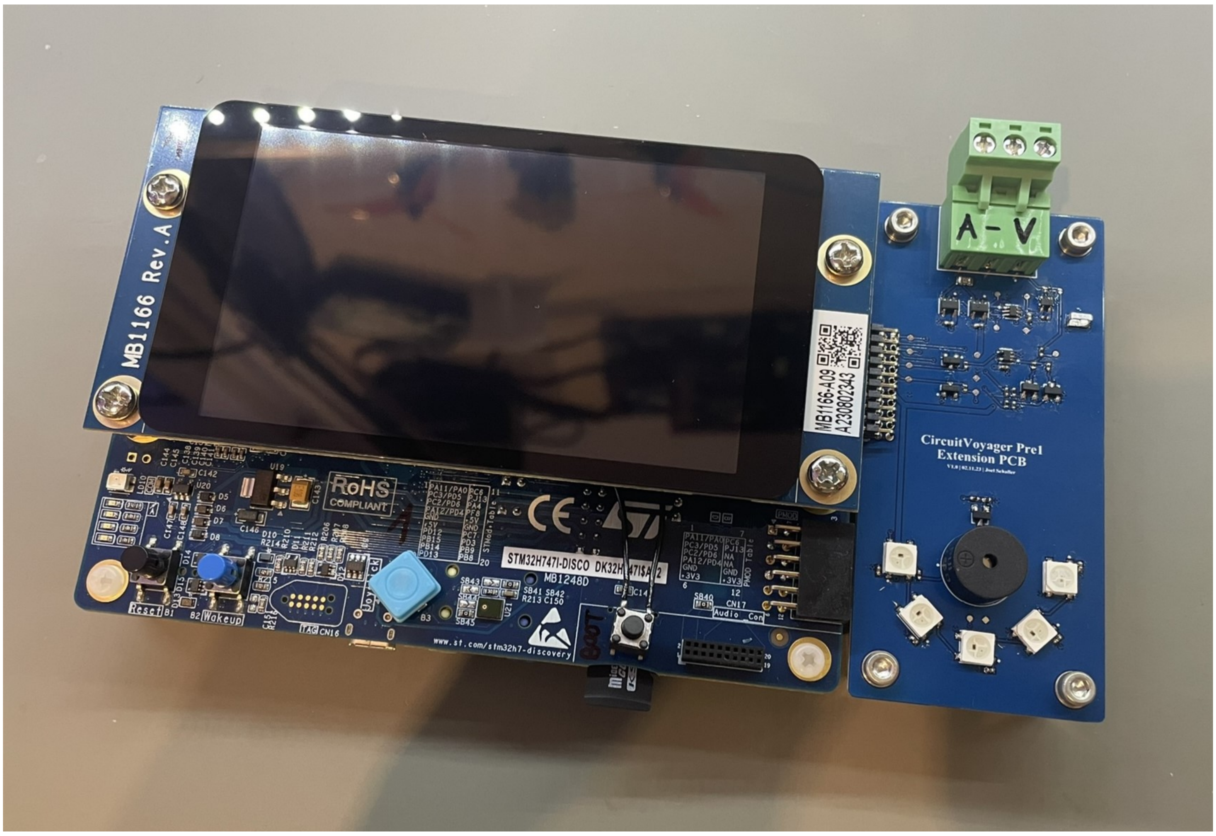
HST Project S5

CircuitVoyager Pre1



Submitted by:
Joel Schaller

Lecturer: Enrico Malacarne

22.12.2023

Abstract

In this Project, my goal is to develop a DMM prototype, that can measure voltage, current and power. The Project can be divided into 3 parts. First the Hardware: I made a PCB whose files are stored in the "5_Hardware" folder and can be opened with Altium Designer (Viewer). This PCB gives the DevBoard the skill to measure voltage, current, beep and light up in color. In extent there's a Firmware which runs on the DevBoard, that automatically measures voltage and current, manages Overloads and range switching and send the measured values over the ST-Links UART to a PC. This Firmware is in the folder "6_Software/CV_FW" and can be opened and flashed using STM32CubeIDE. And last there's a Software running on a PC, which is stored in "6_Software/CV_UI", where the exe file can be opened directly or the python script can be run with PySerial installed. This Software connects to the DMM and displays the voltage current and power values. Later this Software should also be able to log the measured results.

My most important results were, that I learned a lot about developing measurement hardware and I was able to connect a microcontroller to an actual Windows app. I've also learned a lot about LaTeX. I'd rather say this was the most valuable experience gathered in this project.

Contents

1 Introduction **1**
1.1 "Lastenheft" 1
1.2 Mindmap 2

2 Main Body **3**
2.1 "Pflichtenheft" 3
2.2 Extension PCB 4
2.3 The MCU 17
2.4 STM32 Multicore Debugging 17
2.5 Boot Problems 19
2.6 QSPI Flash 20
2.7 User Interface 23

3 Conclusion **24**

4 Appendix **26**
4.1 Journal 26
4.2 Weekly plans 28
4.3 GANTT Chart 30
4.4 Project Agreement 31
4.5 Interview 1 32
4.6 Interview 2 33
4.7 Review 34
4.8 Extension PCB Schematics 38
4.9 Extension PCB Manufacture 11.23 BOM 39

Bibliography **44**

List of Figures **45**

List of Tables **46**

Listings **47**

Acronyms **49**

1 Introduction

The goal of this project is to develop a tiny extension Board for the STM32H747i-Disco Board, to allow it to act as a DMM. Additionally, a SW, that measures the DMM Values and displays them on the touch display. If there's more time I could extend the project with measurement-logging via a SD-Card or over USB to a desktop application.
I want to learn how to implement high speed protocols such as Mipi DSI or QPSI. Later in the last year of my apprenticeship I'd like to develop a whole DMM on my own, but with a different approach as standard ones like these from Fluke. For example, I want to make the DMM rechargeable and modernize it a bit.
To make this project I'm going to use the following tools: Altium Designer, STM32-CubeIDE, LaTeX, TouchGFX, DrawIO.
Also I won't make a diary, because it's easier for me to write my findings down sorted by theme rather than date. But to keep the chronological order of the stuff I've done, there's a journal in chapter: [4.1].

1.1 "Lastenheft"

This is a request from the imaginary customer, I'm making this project for:
I need a prototype for a DMM, that can measure voltage, current and continuity. The DMM should have a touchscreen that displays said values. The UI should be intuitive, so anybody who's ever used a DMM can use it to. And it would be great if you could fit in a power mode, where the DMM uses the voltage and current measurement to calculate the drawn power from the measured device. Because this project will only be used for the proof of concept, the DMM doesn't have to support mains voltage, and we also won't need any safety circuits, AC measuring or negative voltages / currents. It's mainly about the SW. So you can also use DevBoards if there are any available.

1.2 Mindmap

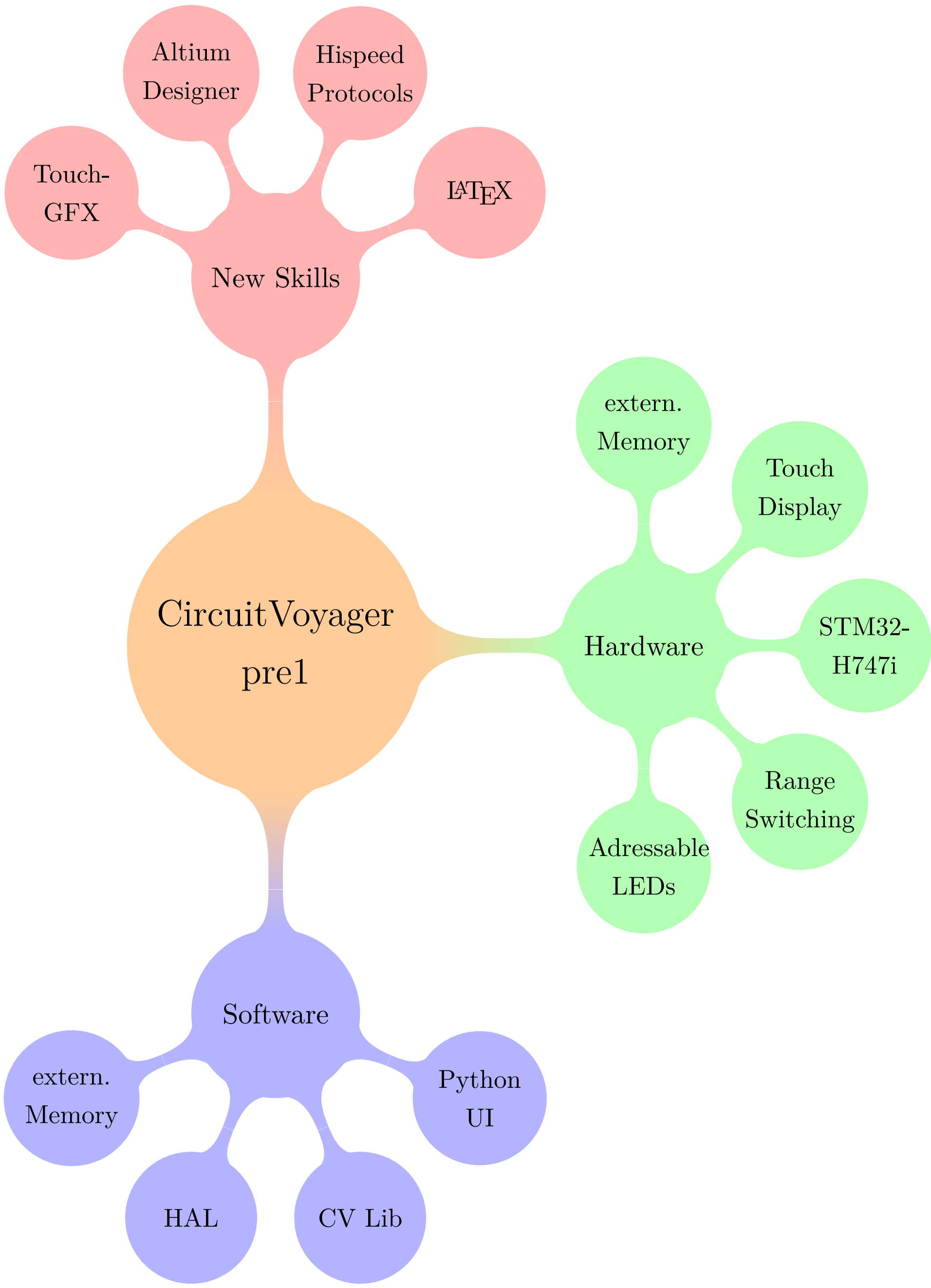


Figure 1.1: Project Mindmap

2 Main Body

2.1 ”Pflichtenheft”

Cost

I’ve already bought two DevBoards one of them stays at TBZ and the other is at home. One of these boards was paid by Mr. Malacarne. Further expenses e.g. the PCB will be paid by me and shouldn’t exceed about 50 CHF, as the HW isn’t that complicated.

Time

The majority of the time in the project, I will work at home because it’s a rather big project to execute in one semester. I will also have much time in the fall holidays to work on it. The project will approximately take 100h to complete. Also, the more detailed time plan is in chapter [4.3] and sequential 2-week plans in chapter [4.1].

Tools

To realize this project I will mainly use, the SW STM32CubeIDE with HAL, STM32 CubeProgrammer and Altium Designer. The documentation is written in LaTeX in VSCode. And I’m planning to order the PCB on JLCPCB and will populate and reflow the PCB at ETHZ, where I’m also allowed to use the measurement equipment for the commissioning and miscellaneous measurements.

Technical Details

value	min.	typ.	max.	unit	description
supply voltage		5		V	over USB
curent to measure	0		1	A	2 ranges
voltage to measure	0		10	V	2 ranges
com. baud rate		115200		s^{-1}	UART

Table 2.1: Technical Details

2.2 Extension PCB

2.2.1 STMod+

Interface from DevBoard to Extension PCB.

- 5V Supply
- SPI
- I²C
- ADC
- Interrupt
- PWM
- GPIOs

I will use the STMOD#14 connection that was intended to use as PWM, as a second ADC input. To measure current and voltage at the same time to later show the power cosumption of the DUT.

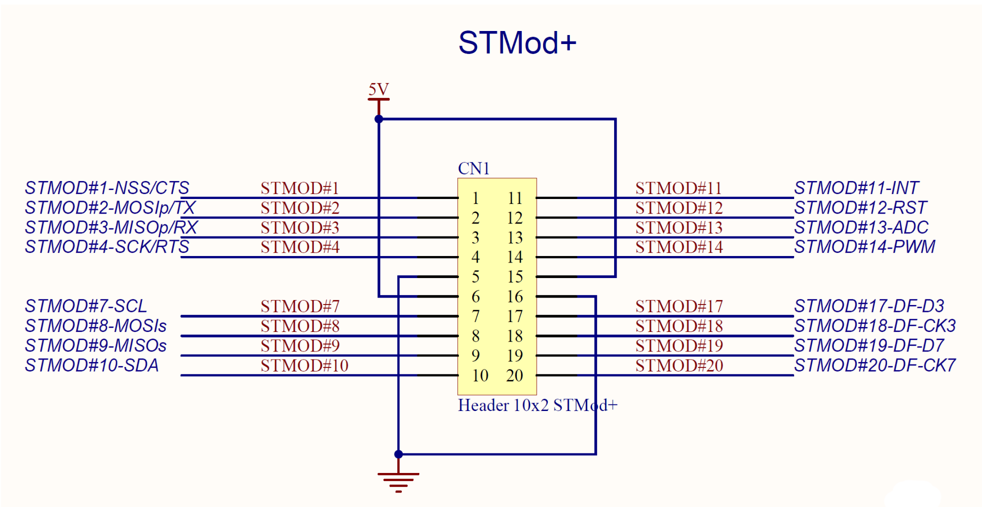


Figure 2.1: STMod+ Interface

2.2.2 Hardware concept

After some thoughts I came up with the following HW concept.

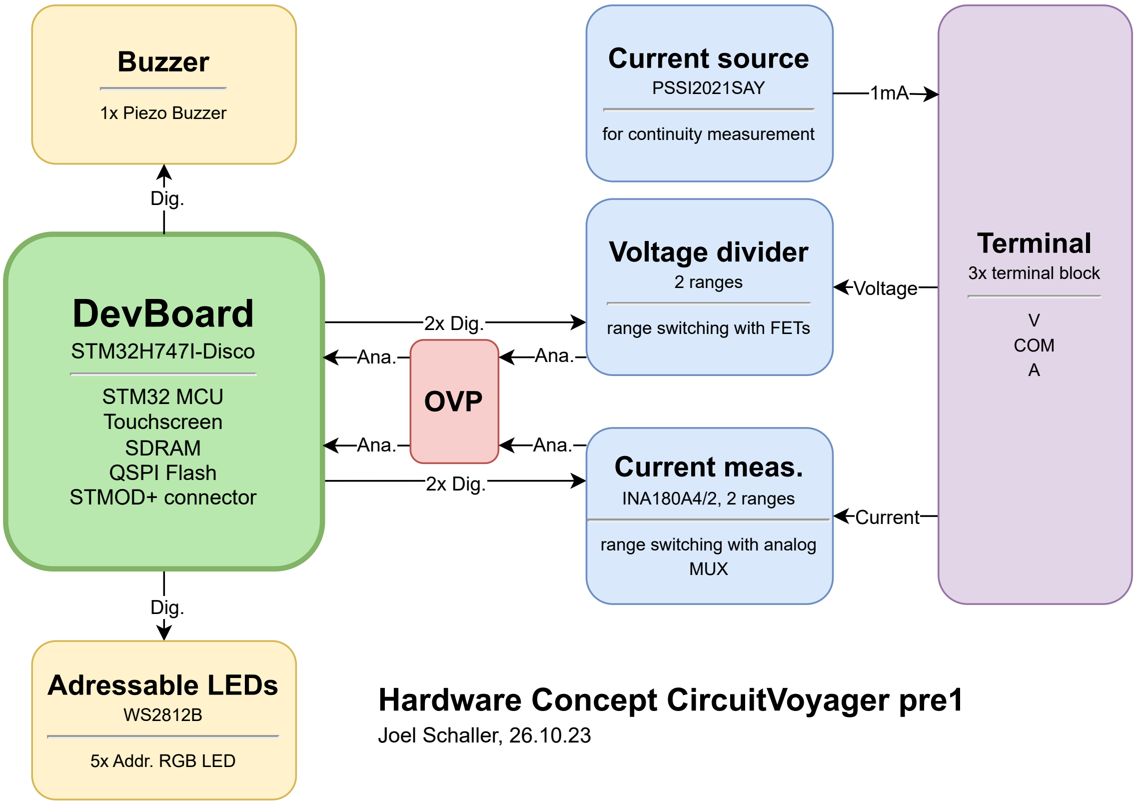


Figure 2.2: Extension PCB HW concept

Voltage measurement

To measure voltage, the DUT should be connected to the terminals V and COM. COM is connected internally to device GND. The V terminal is connected to the Voltage divider block. This block divides the input voltage down, so the ADC in the MCU doesn't overshoot. There are 2 ranges to measure voltage, which can be chosen by setting 2 digital outputs, that go from the MCU to the voltage divider. There's also an OVP, to protect the MCU from voltages higher than 3.3V. [1]

Current measurement

To measure current, the DUT should be connected to the terminals A and COM. COM is connected internally to device GND. The A terminal is connected to current measurement block. This block measures the current, by letting the current flow through a shunt resistor. The DMM can choose which range should be selected with

1 digital Output that is connected from the MCU to a MUX which selects one of two current-sense amplifiers. There's also an OVP, to protect the MCU from voltages higher than 3.3V. [1]

Continuity measurement

To measure continuity, both the voltage divider and the current source are used. The continuity between the V and COM pins is measured. For this a constant current produced by the current source is flowing out of the V terminal. Simultaneously the voltage across those terminals is measured and the resistance / continuity can therefore be evaluated. If continuity is detected, either the buzzer beeps or the LEDs blink. [1]

Note that the continuity measurement doesn't work, because of a schematics fault. More information in chapter [2.2.4].

2.2.3 Schematics

The schematics took me a bit longer than usual, because it's my first whole HW project in Altium Designer. Before I've used KiCAD and Altium has a lot more features and in my opinion is harder to learn. The schematics are in chapter [4.8].

Buzzer Circuit

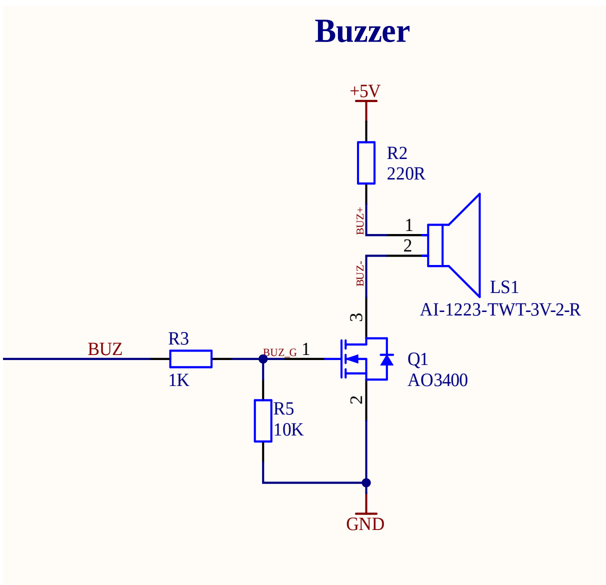


Figure 2.3: Buzzer Circuit

This is an active buzzer. If the BUZ line is pulled high by the MCU, the MOSFET starts to conduct and the buzzer starts beeping. This circuit will be used to give an acoustic feedback to the user, if for example a continuity has been detected.
The resistors R3 and R5 build a voltage divider with a ratio of 1/10. This has the advantage, that the gate capacitance of Q1 is charged with a limited current and if nothing's connected to the BUZ net, the MOSFET turns the buzzer off and the whole machine isn't in an indeterminate state.
The resistor R2 limit the current flowing through LS1. As LS1 is rated for 30mA at 3V.

$$R_2 = \frac{U_{VCC} - U_{LS1}}{I_{LS1}} = \frac{5V - 3V}{30mA} = \underline{\underline{66.\overline{6}\Omega}}$$

$$P_{R2} = I^2 \cdot R = (30mA)^2 \cdot 66.\overline{6}\Omega = \underline{\underline{60mW}}$$

Finally, I've chosen a 220Ω resistor for R2. With this value the sound should be loud enough, that the user hears it. And the power loss of the resistor will be smaller. That means it should be perfectly fine to use a 0402 resistor that is rated for 62.5mW.

Adressable LEDs Circuit

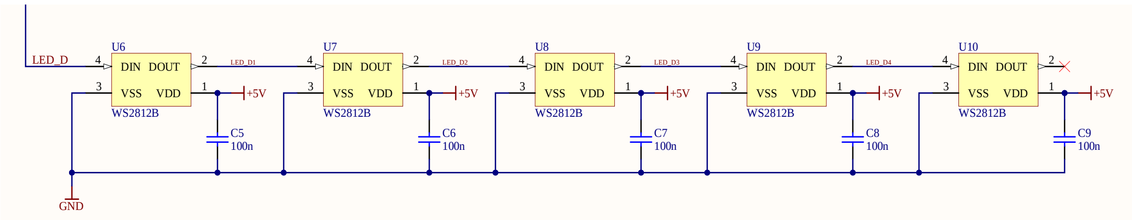


Figure 2.4: Adressable LEDs Circuit

An other option to show if continuity has been detected are these LEDs. The advantage of them is, that they only need one data connection and already include their logic and driving circuits. I've equiped them with one bulk capacitor each, because they're integrated components. They're driven over the 5V rail, which they're specified for. But if they're driven by 5V they theoretically detect all voltages over 3.5V as a logical high. But the MCU only outputs 3.3V. I've used those LEDs much in past projects and this was never a problem, so I'm assuming that it should also work this time.

Overvoltage Protection Circuit

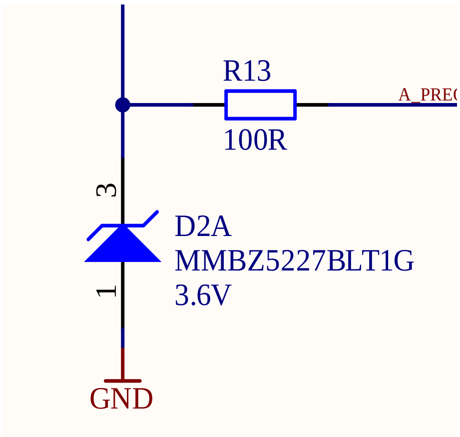


Figure 2.5: Overvoltage Protection Circuit

My Problem is, that the voltages coming from the extension PCB, can't exceed 4V according to the MCUs datasheet. But because this PCB is powered by the 5V rail, theoretically these voltages could damage the MCU. So the goal was to develop a circuit which protects the ADC inputs. The first idea that came to my mind was to use varistors. I've already used them once in a project, but after reading some application notes on this topic, I decided to use a zenerdiode for the OVP. [2]
The zenerdiode I've used has a nominal zenervoltage of 3.6V (max. 3.78V) at 20mA. Together with R13 this should result in a maximum voltage of 3.78V at the ADC inputs. As long as the PREOV voltage doesn't exceed 5.6V.

$$U_{PREOV(MAX.)} = R_{13} \cdot I_Z + U_Z = 100\Omega \cdot 20mA + 3.6V = \underline{\underline{5.6V}}$$

$$P_{R13} = R \cdot I^2 = 100\Omega \cdot (20mA)^2 = \underline{\underline{40mW}}$$

Voltage Divider Circuit

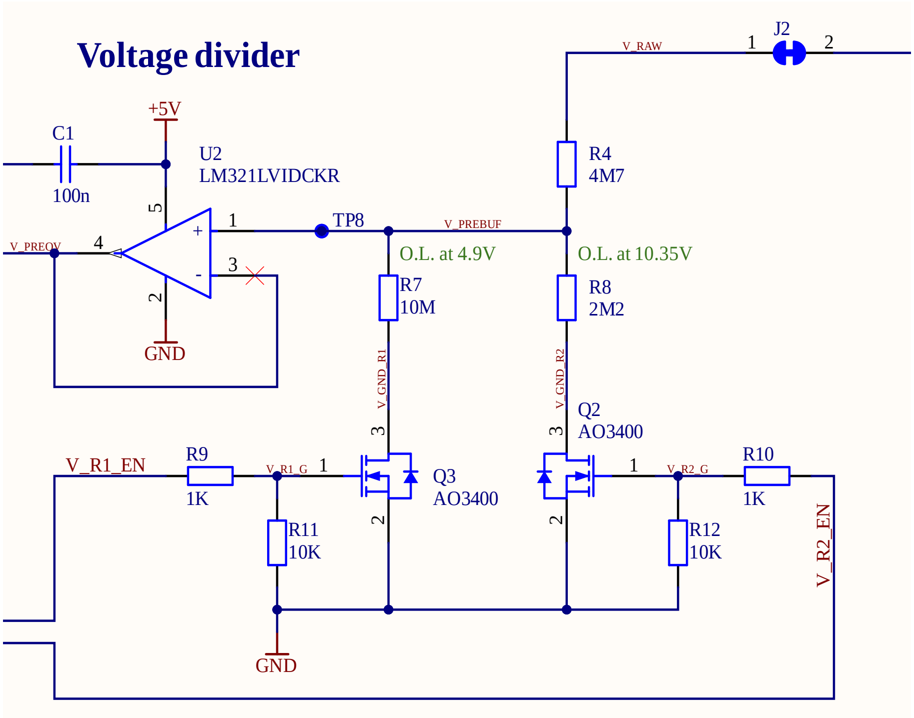


Figure 2.6: Voltage Divider Circuit

The voltage divider is used to measure voltages over the V terminal. This circuit is one approach to range switching, which is described more detailed in the following chapter [2.2.5]. This voltage divider features 2 ranges and I came up with this idea by myself. The ranges are calculated as following:

$$U_{OL(R1)} = \frac{(R_4 + R_7) \cdot U_{OL(MCU)}}{R_7} = \frac{(4.7M\Omega + 10M\Omega) \cdot 3.3V}{10M\Omega} = \underline{\underline{4.851V}}$$

$$U_{OL(R2)} = \frac{(R_4 + R_8) \cdot U_{OL(MCU)}}{R_8} = \frac{(4.7M\Omega + 2.2M\Omega) \cdot 3.3V}{2.2M\Omega} = \underline{\underline{10.35V}}$$

This divided voltage is then fed into an impedance converter which guarantees that the voltage divider isn't manipulated by the ADCs internal resistance and also helps on the DevBoards analog paths, because they're very long and therefore vulnerable to electromagnetic fields.

Current Source Circuit

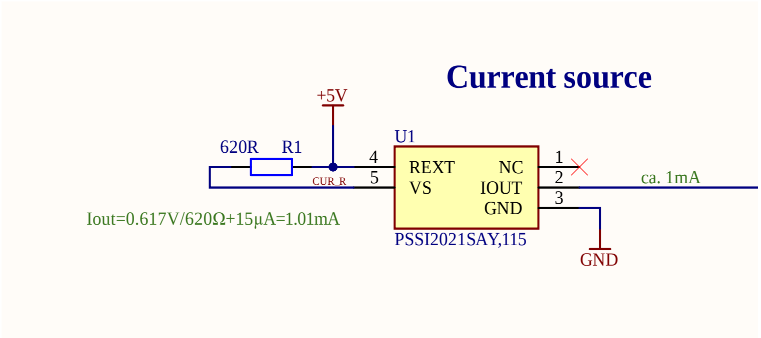


Figure 2.7: Current Source Circuit

The current source circuit will be used to measure continuity and resistance, together with the voltage measuring function. A constant current will be induced to the DUT and because the voltage over the DUT can be measured, the resistance can be calculated. The constant current can be calculated as following:

$$I_{OUT} = \frac{U_{VS}}{R_1} + 15\mu A = \frac{0.617V}{620\Omega} + 15\mu A = \underline{\underline{1.01mA}}$$

But this doesn't matter that much, because the DMM will be calibrated using the solderbridge J1.

Note that this circuit isn't used in the final design, because there's no option to disconnect this circuit from the voltage measuring circuit via the MCU. So I decided to leave the current source disconnected from the rest of the circuits. More information in chapter [2.2.4].

Current Measuring Circuit

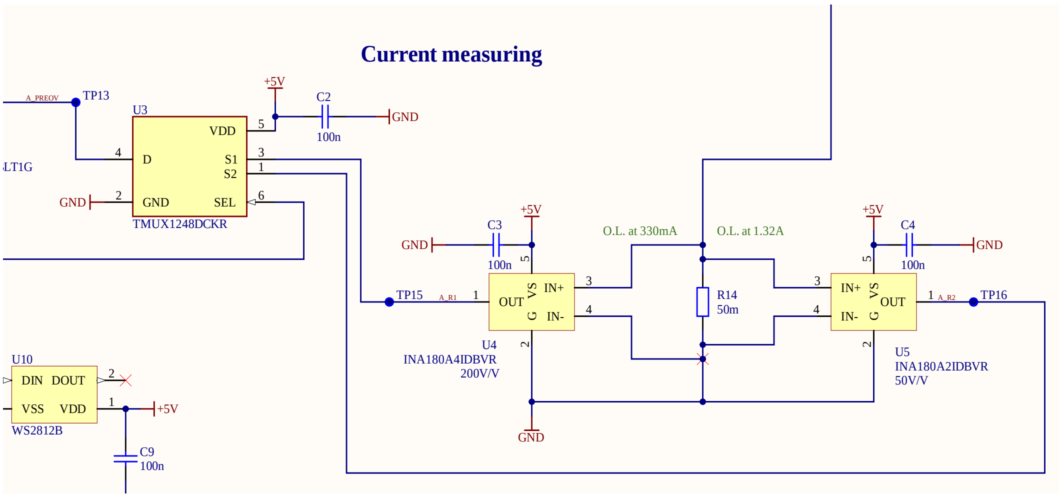


Figure 2.8: Current Measuring Circuit

The current measuring circuit is used to measure currents with the A terminal. This circuit is one approach to range switching, which is described more detailed in the following chapter [2.2.5]. This circuit features 2 ranges and I came up with this idea, while reading some articles on DMMs [3] [4]. The ranges are calculated as following:

$$I_{OL(R1)} = \frac{U_{OL(MCU)}}{R_{14} \cdot G_{U4}} = \frac{3.3V}{50m\Omega \cdot 200\frac{V}{V}} = \underline{\underline{330mA}}$$

$$I_{OL(R2)} = \frac{U_{OL(MCU)}}{R_{14} \cdot G_{U5}} = \frac{3.3V}{50m\Omega \cdot 50\frac{V}{V}} = \underline{\underline{1.32A}}$$

These voltages are then fed into an analog MUX, which switches the processed voltages to the ADC. This has the advantage, that the DMM doesn't manipulate the actual path, where the current is flowing through and therefore not influencing the DUT.

The power rating of R14 is 125mW, while the maximal power loss of R14 at 1.32A is 87.12mW.

$$P_{R14} = I^2 \cdot R = (1.32A)^2 \cdot 50m\Omega = \underline{\underline{87.12mW}}$$

2.2.4 PCB

Layout

The PCB layout was also made in Altium Designer. This was therefore also my first layout in Altium Designer. The only problem I had, were the vias. I had to improvise by setting their parameters manually instead of letting Altium do it.

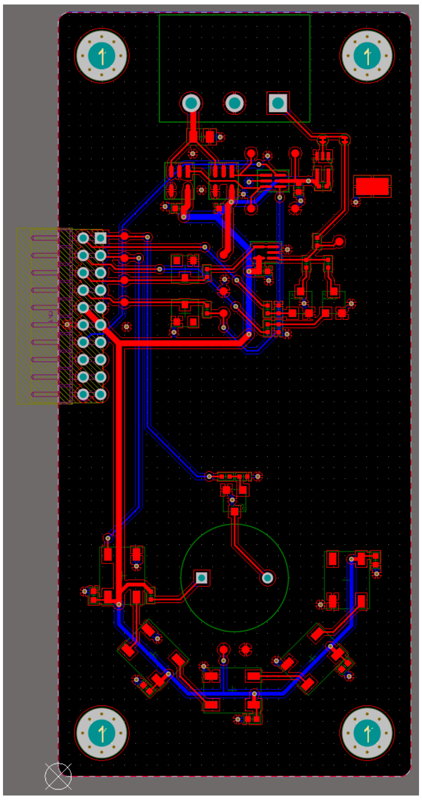


Figure 2.9: Extension PCB Layout

Later I've generated the output files (Gerber files, BOM, assembly drawing) and ordered the PCB on JLCPCB with 2 layers, tented vias, removed order number and blue solder mask. The BOM is in the appendix chapter [4.9]. The components were sourced using DigiKey and ETH SPH.

Assembly

Then I've assembled the PCB by hand, which wasn't a problem at all. But it turned out, that I've ordered the wrong connectors for the STMod interface. Instead of the 4mm long pins I've ordered the 2mm long pins, which don't make contact with the socket from the DevBoard. But luckily I was able to order longer ones, which arrived quite fast.

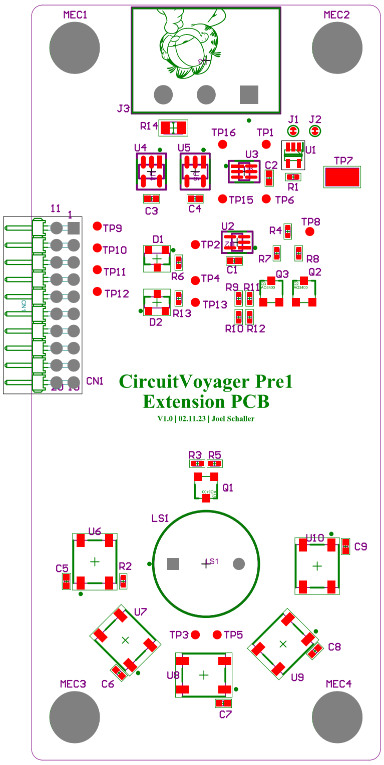


Figure 2.10: Extension PCB Assembly Drawing

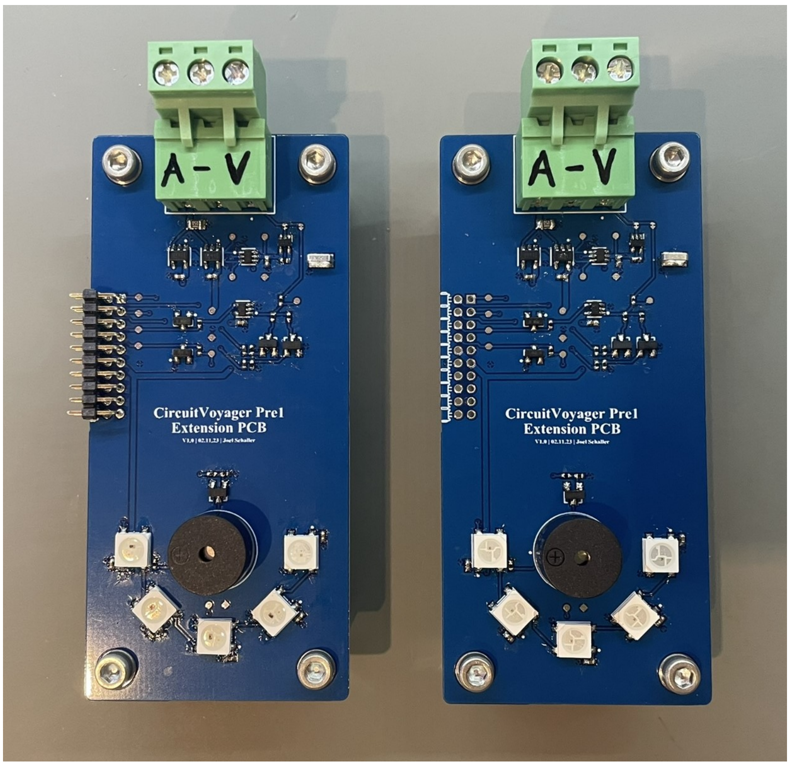


Figure 2.11: Extension PCB

Commissioning

I've only documented one commissioning of the 2 PCBs. The other one was tested, using the same process.

Action	Measurement point	Expected	Measured
Connect power supply @ 5V	CN1.6 → TP7	max. 100mA	**18mA**
Test Buzzer (TP9 to 3.3V)	acoustic	beeps	**PASS**
Test const. Current	J3.1	1mA	**13mA**
LEDs on with AWG (CN1.2)	optical	lights up	**PASS**
Current meas. test Range 1	CN1.14	10V/A	**fig: 2.13**
Current meas. test Range 2	CN1.14	2.5V/A	**fig: 2.13**
Voltage meas. test Range 1	CN1.13	0.68V/V	**fig: 2.14**
Voltage meas. test Range 2	CN1.13	0.32V/V	**fig: 2.14**

Table 2.2: commissioning PCB 1

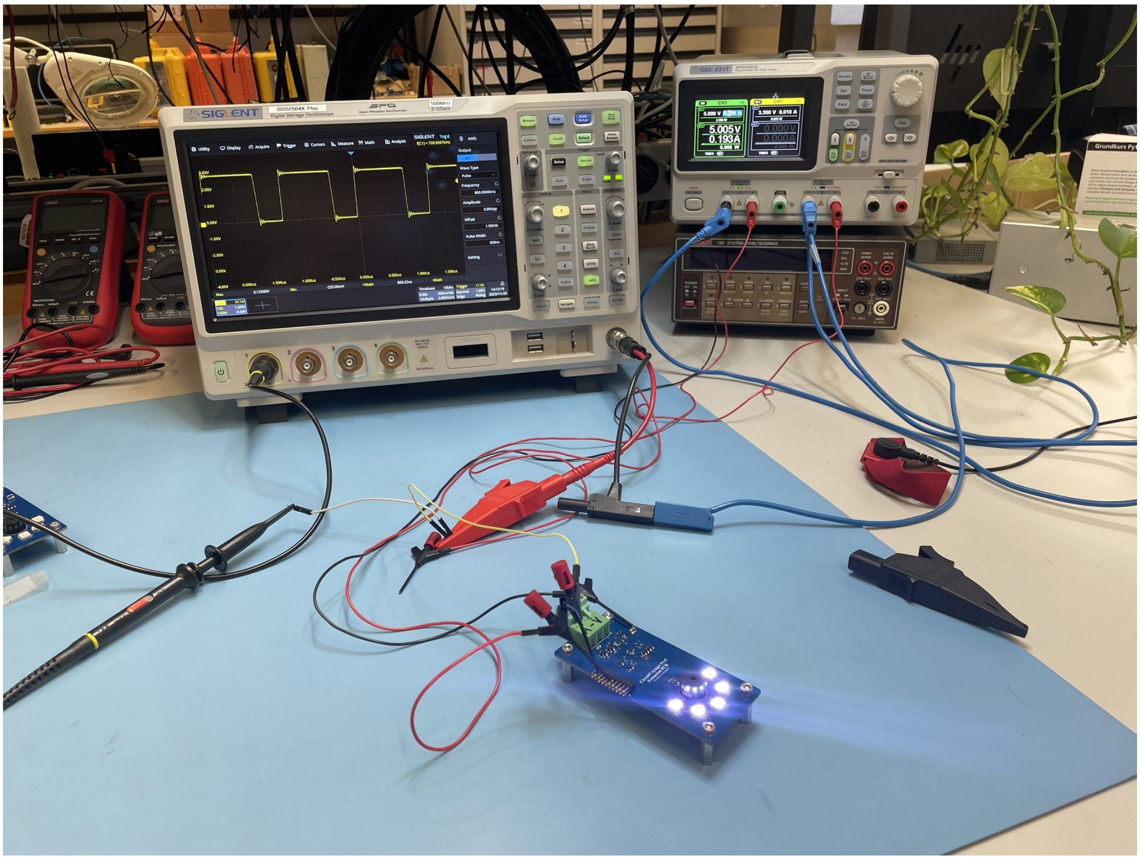


Figure 2.12: Commissioning Lab

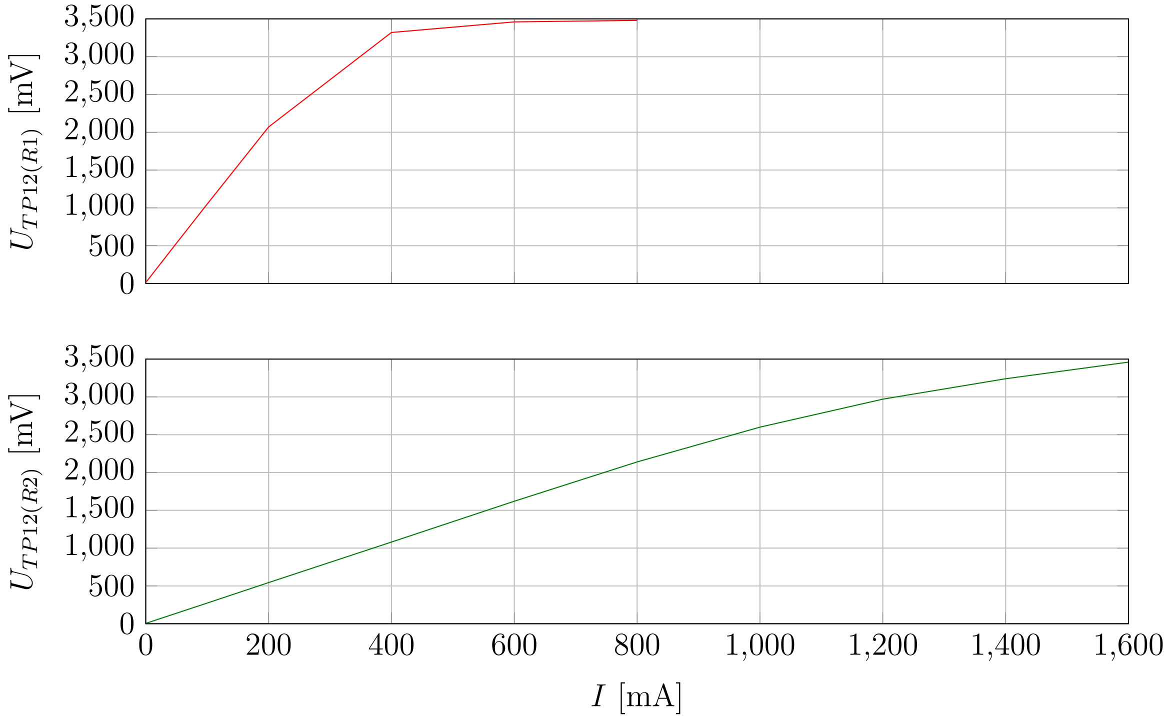


Figure 2.13: Commissioning PCB1 current measurements

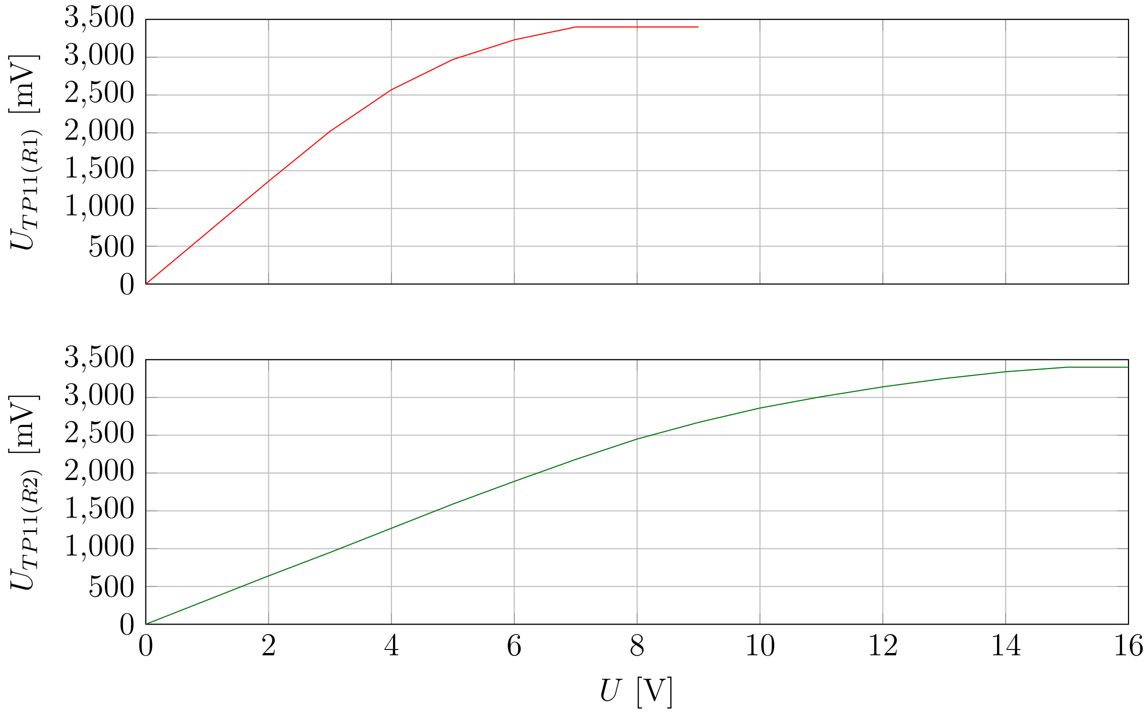


Figure 2.14: Commissioning PCB1 voltage measurements

Bad Design

During the commissioning I noticed, that some of my circuits are kind of misconstructions. This is due to my missing design review but gives me the opportunity to reflect a lot and learn about my fault the hard way.
So first my Current Source has the problem, that it's directly connected to the V terminal which is done correct, only considering the resistance mode of the DMM. But when trying to measure a voltage, the current source will inject a current into the voltage measuring circuit which makes the voltage measuring circuit / mode unusable. So the current source should be disconnected from the rest of the circuit, to use the voltage mode of the DMM. Because I don't have enough time to fix this problem, I will cut-through J1.
The current source also doesn't work as intended, as the constant current which should be 1mA is now around 13mA. But I don't bother, because I won't use the current source anyway. See also: Figure [4.9]

2.2.5 Range Switching

First the option to switch resistors in the measuring path, by using some kind of transistor. As currently used in the voltage measuring circuit. This range switching part has the advantage of using many ranges at once, as for example with 2 range switches can use 3 different ranges and with 3 range switches can use 7 ranges. On the other hand it has the disadvantage of only being able to divide the input voltage and not being able to amplify. Additionally, the internal resistance is always changing and therefore impacting the DUT. I wouldn't recommend this circuit because of said factors and only switch on the measured side. For example use different Amps / Divs to create the right voltage ranges and only switch the high Z path, that doesn't have an impact on the DUT.
The other option I've tested is to change the amplification factor. Like I've done in the current measuring circuit. This approach has the advantage that the DUT isn't affected by the range switching, because the shunt resistor is always the same and current can flow, even if the DMM is turned off. Additionally, this type of range switching can both, amplify and dampen the input signal. This can be approached by using multiple amplifying circuits and then select from them with a multiplexer, or by changing the amplification factor of one amplifier with a digitally controlled potentiometer.

2.3 The MCU

For this project I've chosen a high speed microcontroller. I'm using an STM32H747-XIH6U. The special thing about this MCU is, that it has 2 cores. The main core is a Cortex M7 at a frequency of up to 480MHz and the second one is a Cortex M4 running at up to 240MHz. But this also means that it's a bit expensive at about 20CHF. This MCU also features 1MB of flash per core and a total of 1MB RAM. But these memories can get full rather quickly, when working with graphical displays or the USB stack for example. That's why I'm using an external flash and RAM in this project.

Note that in the end I haven't used external memories because I didn't use the touchscreen nor the USB function of the MCU. More about that in chapter [2.6.2]

2.4 STM32 Multicore Debugging

As usual, I'll be testing the DevBoard by programming a good old blink sketch. The information for this multicore blink sketch I got from Controller Tech's YouTube channel [5], because the official SW documentation from STM isn't that good, and therefore I wasn't able to get the information I needed to run my first sketch.

2.4.1 Procedure

1. Create a project with the STM32 Cube template for the STM32H747XIH6. (Using default configuration)

2. Set up the RCC to use the external oscillator and configure clocks to their maximum frequencies.

3. Set up the GPIOs of for the debug LEDs. Special is, that you have to assign a core to each GPIO pin. This parameter says, which core is able to manipulate said GPIO and in which under project (CM4 or CM7) has the define with the GPIO name defined.

STM32 pin	LED number	LED color	Name in code	Core
PI12	LED1	Green	LED_G	CM7
PI13	LED2	Orange	LED_O	CM7
PI14	LED3	Red	LED_R	CM4
PI15	LED4	Blue	LED_B	CM4

Table 2.3: LED Annotation

4. Under: Project Manager → Code Generator: Enable the option: "Generate Peripheral initialization as a pair of .c/.h files per peripheral".
   This should make the project more structurized.

5. Generate code.

6. Now 2 under projects are created, for each core one. Rather than using the standard files, that are already full of redundant information, I will create my own main files, with the advantage that they're also easily separable between the CM4 and CM7. The files written by me are stored in the folders "Core/Usercode"

7. Write the code to blink the LEDs for each core:

```c
#include "usermain_CM4.h"

void usermain_cm4(void)
{

    while(1)
    {
        HAL_GPIO_TogglePin(LED_B_GPIO_Port, LED_B_Pin);
        HAL_Delay(500);
    }
}
```

Listing 2.1: Blink Code CM4

8. Now configure the debug configuration for the CM7: Enable "Halt all cores" and under Startup add the CM4 config with the "Debug" Build configuration selected.

9. Now configure the debug configuration for the CM4: set port to 61237, set reset behaviour type to "None" and under Startup → Edit: uncheck the download option.

10. Now to flash the MCU it's important, that all files are saved, because they're not saved automatically as usually in single core projects. Then Click on the "Run" button, with the CM7 configuration selected, the CM4 won't work.

11. To debug the MCU also save all files and then start debugging with the CM7 configuration selected. After the CM7 session started, click again on the debug button and start the session for the CM4. Now run both threads simultaneously by selecting them both with the "Ctrl" key and then starting them. After those threads started and the clock is configured, they can be manipulated separately.

Additionally, I'd recommend watching the video from Controllers Tech, as it's really informative. [5] For Example Controllers Tech explains how the startup of the cores work and how you can disable cores. There are also more videos from Controllers Tech in this "Multi Core STM32" series.

2.5 Boot Problems

While programming the MCU for the first times, I had the problem that the MCU wasn't detected sometimes by the ST-Link. Therefore, the MCU wasn't resetable nor new firmware could be uploaded. It turned out, that the ST-Link could program the MCU again, when accessing the BOOT mode, which is intended to boot the MCU over USB DFU. But this mode also stops the MCU from starting the script and therefore the ST-Link detects the MCU. The boot mode can be accessed by shorting the pads of R192. Then the MCUs memory can be erased and after removing the solderbridge, the MCU operates again as normal. After this problem, with the not detectable MCU occurred multiple times, I decided to add a button to access the boot mode. So I don't have to have a soldering iron around, when the MCU ain't detectable. This modification looks as following:

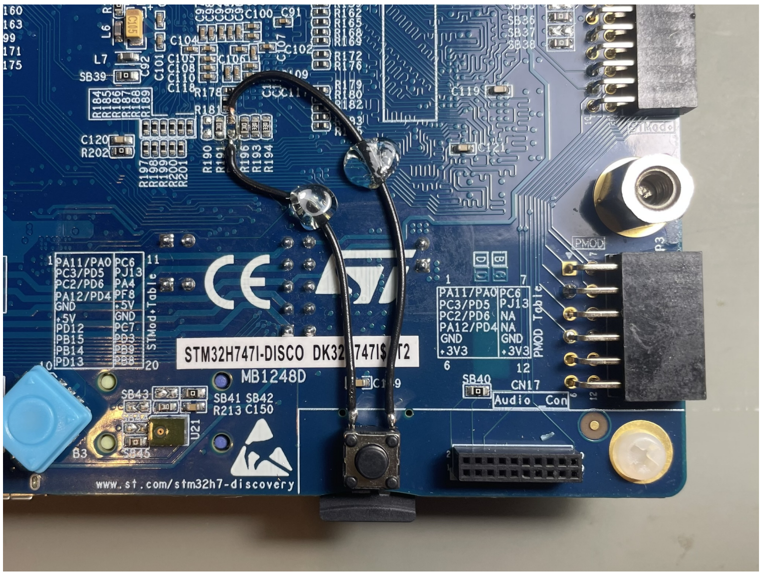


Figure 2.15: Hardware Boot Button Mod

Later it turned out, that those problems came from a faulty RCC configuration. That's something I should investigate further. Because right now I'm only able to get up to 400MHz for the CM7. This probably has something to do with the SMPS and Voltage scaling option in the RCC tab.

2.6 QSPI Flash

To extend the internal flash of the MCU, there is 128MB of QSPI Flash on the DevBoard. In this section I'll try to implement this flash, so I'm able to store the actual firmware on this external flash. This also means, that the MCU should then boot from the external memory.

2.6.1 Write Test

In this first test I'll try to simply write some data to the Flash and therefore confirm, that the communication between the MCU and flash works as intended. For this I've again followed the instructions from Controllers Tech on YouTube. [6]

1. Enable QSPI block in "Dual Bank with Quad Lines" mode (Because DevBoard uses two ICs, each 512Mbit) for CM7.

2. Set up pins according to the schematics. (standard cube config is wrong.) Configure Pins to CM7 in "very high" speed mode.

3. Enable chip select 1 for both banks.
4. Set clock prescaler to 2. (200MHz Clock divided by 2 to the power of 2 $\rightarrow$ 50MHz) (Flashs max. frequency in DTR mode (dual bank) is 90MHz)
5. Fifo threshold to 4.
6. Sample shifting to "Sample Shifting Half Cycle"
7. Flash size to 26. (Flash size is 2 to the power of 26 + 1 Bytes. $\rightarrow$ 128MB)
8. Chip select high time to "6 Cycles"
9. For Cortex-M7 Enable: ICache and DCache.
10. Generate code.
11. Then add user code lines from STM GitHub. [7] to qspi.c and qspi.h.

Now you can download the firmware to the MCU. And verify, that no error is triggered. (Orange LED blinking) Then to verify, that the string was written to the flash: open STM32CubeProgrammer and enable the external loader for the H747-Disco DevBoard. Now at the address 0x90000000, the string "msgOut" should be shown.

```c
#include "usermain_CM7.h"

uint8_t* msgOut = (uint8_t*)"1234567890";

void usermain_cm7(void)
{
    if(CSP_QUADSPI_Init() != HAL_OK)
    {
        Error_Handler();
    }

    if(CSP_QSPI_Erase_Chip() != HAL_OK)
    {
        Error_Handler();
    }

    if(CSP_QSPI_WriteMemory((uint8_t*)msgOut, 0, strlen((char
    *)msgOut)) != HAL_OK)
    {
        Error_Handler();
    }

    while(1)
    {
        HAL_GPIO_TogglePin(LED_0_GPIO_Port, LED_0_Pin);
        HAL_Delay(500);
    }
}
```

Listing 2.2: QSPI Write Test CM7

While playing a bit with this code, I've noticed, that the flash can't manipulate single bits. It can just turn a 1 into a 0 but nor in reverse. To set a 0 to a 1 you have to erase a whole sector or the whole chip. After some research on this topic I've learned, that this behaviour is common for flash memories, as they're normally only used to store data over a long time. For example: Firmware.

2.6.2 Boot from QSPI

After trying to boot from the QSPI flash, I decided to change my plans an go on without a bootloader and external RAM. At the moment I don't know to less debug the problems I've had and therefore I decided to go on with the easier tasks of this project like the actual measurements and move the Memories implementation to another project.

2.7 User Interface

I first tried to use TouchGFX to design an embedded UI using STMs TouchGFX which is a code generator for graphical displays with STM32 microcontrollers. But it turned out, that it's a lot more complicated that I originally thought. This is due to the usage of FreeRTOS and C++, which I both didn't know by then. So also because of the limited time I decided to make a tkinter app in python to show the measured values. The app is in the software folder and has the name "CircuitVoyager UI". As a base I used the following GitHUB repo: [8]

3 Conclusion

Project Start

I've noticed that it was quite hard for me to start with the project. There are so much topics to imply, that it can get overwhelming and very emotional. I've also underestimated the time needed to set up the documentation LaTeX files and the whole planning. It can be demotivating if you're spending a lot of time and in the end didn't really start with the project. But later I was able to start the project without much trouble.

Time Plan

Well, the GANTT chart as you can see didn't help out much for me. Because it's really hard to plan such a projects timing over half a year. There are so many unsolved problems and in the beginning I haven't been able to estimate the duration of all the processes involved. On the other hand my 2-week plans were working great and helped a lot, to notice what was to do.

HW development

I've learned much in this project part. Mainly this was Altium Designer, as this was my first complete project I've realized in Altium Designer. This also leaded to some not nicely solved solution. All components for example have their own properties and this leads to a unreadable BOM.

But in the end I've also noticed, that the dataflow (that usually should go from left to right) goes in the wrong direction. This had already started in the HW-Chart and therefore also ended up in the schematic.

I've also used Draw.io one of the first times and by now it looks very promising. It's much easier and more straight forward than MS Visio. Everything just works as intended.

SW development

Because in this project I've used a dual-core MCU for the first time, it was really hard to write the code at first. And with time I noticed that it's to complicated for me to implement such modern techniques. This is also due to the bad documentation of tools like FreeRTOS TouchGFX or multicore MCUs.

Project Idea

The Project Idea in the first was really good. But as you can see, some minor decisions weren't as intelligent as they could be. I noticed that I'm much better at the "old style" electronics, without any operating system, overpowered MCUs, or TouchGFX. But I'm happy, that I was able to switch the focus of my project from hi-speed protocols to more embedded software and LaTeX. In the end I have a product that I'm proud of and that works. But I won't move on with the Circuit Voyager in the next semester, because I also noticed, that It's rather hard to make the hardware development of measurement equipment. This takes a lot of time, also away from school. And in the next semester I want to focus more on stuff like Swiss Skills and therefore make an easier project.

Outlook

As described in the project idea chapter. I want to stop the development of the Circuit Voyager and move on with an easier more embedded and hardware close project with simpler protocols and so on. But on the other hand I'll also document my further projects with LaTeX, because I got a lot faster, and it's more predictable than MS Word.

4 Appendix

4.1 Journal

Date	Location	Duration	Activity
01.09.2023	TBZ	1.5h	Selected and bought DevBoard
08.09.2023	TBZ	2h	Tested DevBoard with demos
08.09.2023	TBZ	0.5h	Noted first ideas for DMM
15.09.2023	TBZ	1.5h	Written and signed Project Agreement [4.4]
21.09.2023	Home	3h	Created documentation template
22.09.2023	TBZ	2h	Started writing Journal [4.1]
24.09.2023	Home	1.5h	Made GANTT chart [4.3]
27.09.2023	Home	2h	Written detailed planning and introduction
29.09.2023	TBZ	1.5h	Added mindmap, Lasten-, Pflichtenheft
06.10.2023	TBZ	1h	Started with block diagram (extension PCB)
20.10.2023	ETH	0.5h	Started Altium Project, Schematic template
23.10.2023	ETH	3h	HW Concept / documentation
23.10.2023	ETH	3h	Start schematic / documentation
25.10.2023	Home	2h	Schematic: Current Src, Volt div
26.10.2023	ETH	1.5h	Schematic: Current meas, ERC
26.10.2023	Home	1h	Schematic: Cleanup, Comments, DS Saves
26.10.2023	Home	1h	Documentation & prepared Interview 1
27.10.2023	TBZ	1.5h	Documentation: Started with schematic part
28.10.2023	Home	2h	Documentation: schematic parts
29.10.2023	Home	1h	Docu: schematic done, resources cleanup
29.10.2023	Home	0.5h	Layout: Outline, Placement start
31.10.2023	Home	1h	Layout: Placement Done
01.11.2023	Home	1.5h	Layout: Routing Done
02.11.2023	Home	1h	Layout: Cleanup & Export
03.11.2023	TBZ	-	Interview 1: signed by mala
03.11.2023	TBZ	1h	Pepared BOM for supplying
04.11.2023	Home	1h	Documentation: PCB, Reflection

Table 4.1: Project Journal 1

Date	Location	Duration	Activity
05.11.2023	Home	0.5h	STM32 Multicore Debugging
05.11.2023	Home	1h	Documentation: STM32 Multicore Debug
05.11.2023	Home	1h	Boot problems / HW MOD
09.11.2023	Home	3h	Population of 2 Extension PCBs
11.11.2023	Home	0.5h	HW MOD of second DevBoard
14.11.2023	Home	2h	QSPI Write test and documentation
23.11.2023	Home	3.5h	QSPI: tried to Ext. boot
29.11.2023	ETH	3h	Commissioning both PCBs
29.11.2023	Home	1h	Documentation commissioning
01.12.2023	TBZ	1.5h	Documentation and Interview 2
13.12.2023	Home	3h	Made CircuitVoyager UI for PC
14.12.2023	Home	2h	Started with CV Firmware
15.12.2023	Home	3h	FW: Ranges and OL and lib
16.12.2023	Home	3h	FW: WS2812, CONECT-FB
16.12.2023	Home	1h	GANTT & PowerPoint
20.12.2023	ETH	1h	Documentation review
20.12.2023	ETH	0.5h	filled out review
21.12.2023	ETH	4h	completed documentation

Table 4.2: Project Journal 2

4.2 Weekly plans

4.2.1 KW39 & 40

- Write introduction
- Planning: Cost, Tools, When, Why
- Create project diagram (learning process)
- ”Lastenheft”
- ”Pflichtenheft”
- Make a HW-Digram for the Extension PCB.
- Make the schematic of the Extension PCB.
  - Part to measure voltage.
  - Part to measure current.
  - Part to measure continuity.
  - Addressable LEDs.
- Start with the Layout of the Extension PCB.
- Reflection of the start of the project.

4.2.2 KW41 & 42

Fall holidays. I planned to invest much time in the holidays, but it turned out that my plans changed, and I was busy.

4.2.3 KW43 & 44

Mainly catching up.

- Make a HW-Digram for the Extension PCB.
- Make the schematic of the Extension PCB.
  - Part to measure voltage.
  - Part to measure current.
  - Part to measure continuity.

  - Addressable LEDs.
- Make the whole Layout of the Extension PCB.
- Order the Extension PCB and the components.
- Reflection of the start of the project.

and maybe if the time is sufficient I could start with implementing the SDRAM and QSPI Flash.

4.2.4 KW45 & 46

- Populate and commission the extension PCB.
- Implement QSPI flash with bootloader.
- Implement external SDRAM.
- Implement touch display. (MIPI DSI, Touch)

and maybe if the time is sufficient I could start with learning TouchGFX.

4.2.5 KW47 & 48

- Boot from QSPI flash.
- Commissioning of Extension PCB.

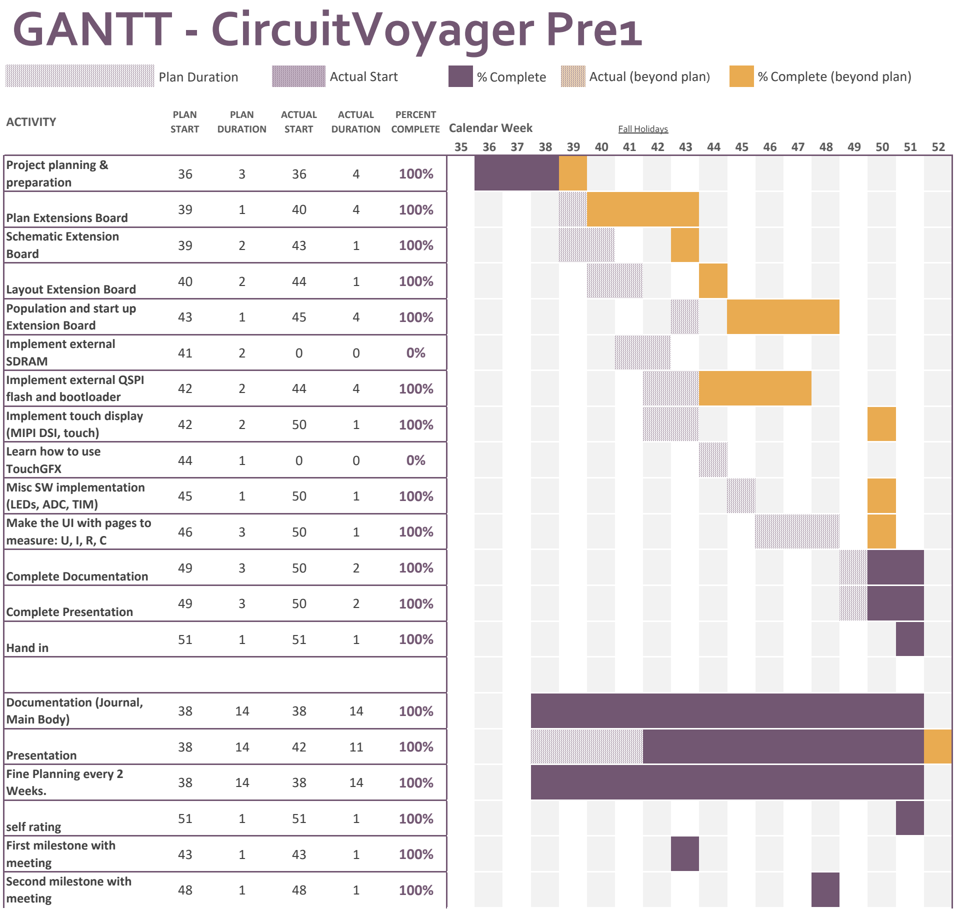
4.2.6 KW49 & 50

- Make Firmware for MCU.
- Make UI with python for PC, which shows voltage, current and power.

4.2.7 KW51

- Complete documentation.
- Fill out review.
- Start with PowerPoint. Finalize in Christmas holidays.

4.3 GANTT Chart



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Project planning & preparation	36	3	36	4	100%
Plan Extensions Board	39	1	40	4	100%
Schematic Extension Board	39	2	43	1	100%
Layout Extension Board	40	2	44	1	100%
Population and start up Extension Board	43	1	45	4	100%
Implement external SDRAM	41	2	0	0	0%
Implement external QSPI flash and bootloader	42	2	44	4	100%
Implement touch display (MIPI DSI, touch)	42	2	50	1	100%
Learn how to use TouchGFX	44	1	0	0	0%
Misc SW implementation (LEDs, ADC, TIM)	45	1	50	1	100%
Make the UI with pages to measure: U, I, R, C	46	3	50	1	100%
Complete Documentation	49	3	50	2	100%
Complete Presentation	49	3	50	2	100%
Hand in	51	1	51	1	100%
Documentation (Journal, Main Body)	38	14	38	14	100%
Presentation	38	14	42	11	100%
Fine Planning every 2 Weeks.	38	14	38	14	100%
self rating	51	1	51	1	100%
First milestone with meeting	43	1	43	1	100%
Second milestone with meeting	48	1	48	1	100%

Figure 4.1: GANTT Chart

4.4 Project Agreement

Information	TBZ/EE / **2337.00** jschaller	HST
HST: BüP	*Sem:* **5**	1_Project_Agreement.docx

TBZ Technische Berufsschule Zürich Abteilung Elektro/Elektronik

1 Projektvereinbarung

Verfasser/innen: Joel Schaller **Klasse:** BEN21
Titel: CircuitVoyager pre1

1. **Thema** (Hintergrund, Überblick, gegenwärtiger Wissensstand)

Develop a tiny extension Board for the STM32H747i-Disco Board, to allow it to act as a DMM. Additionally, a software, that measures the DMM Values and displays them on the Touch Display. If there's more time I could extend the Project with Measurement Logging via a SD-Card or over USB to a Desktop application.

2. **Eigene Fragestellung / Untersuchungsgegenstand**

2.1 Eigene Fragestellung (Leitfrage)
How to implement the following functions / protocols? (QSPI Flash, SDRAM, TouchGFX, Mipi DSI) and if the time is sufficient: (FAT with SDcards, Bootloaders)
2.2 Hypothese (Vermutung über das Ergebnis)
I want to learn, working with High Speed MCUs and implement such protocols.
2.3 Methoden und Vorgehen (mindestens 2 Methoden müssen angewendet werden)
HW-Dev (Altium), SW-Dev (STM32Cube with HAL), Documentation in LaTeX
2.4 Hilfsmittel
Internet, literature
2.5 Kontaktpersonen, Informationsstellen, Institutionen
Teachers, Instructor at ETH, Dad

3. **Persönlicher Bezug / Motivation**
In the next 2 Years I want to develop my own DMM, because I think there's much to improve with standard DMMs as Fluke. For Example: Touch Display, Rechargeable Battery…

4. **Bewertungsform**
This Project will only be done by me. Time: about 28 lesson and unknown time at home. Project delivery on: 12.01.2023

5. **Besprechungstermine mit Lehrperson** (vorgeschrieben sind zwei Besprechungen)
Termin 1: 27.10.2023
Termin 2: 01.12.2023

Datum: 15.09.23 Die Lernenden: ……………
Datum: 15.9.23 Die Lehrperson: ……………

DMM = Digital Multimeter
TouchGFX = Graphical Designer for Embedded Touch Displays

BEN21 Seite 1 (1) 1_Project_Agreement.docx

Figure 4.2: Project Agreement

4.5 Interview 1

Projektbewertung **MAL / 2109.00** -- HST / BüP
HST / BüP: Projektarbeit 2023 *Sem:* **5** 96_Interview_1.docx
TBZ Technische Berufsschule Zürich Abteilung Elektro/Elektronik

Protokoll der Zwischenbesprechung

Zwischenbesprechung **1** Datum: 27.10.23

Name: Joel Schaller Klasse: BEN21a

Titel der Arbeit: CircuitVoyager pre1

Dieses Formular ist soweit möglich ausgefüllt an die Zwischenbesprechung mitzubringen. Weiter sind möglichst alle Unterlagen mitzubringen, vor allem das Geschrieben (Arbeit, Dokumentation, aktuelles Projektjournal, Zeitplan, Notizen, Korrespondenzen) sowie die wichtigsten Informationsmaterialeien (Bücher, Datenblätter etc.)

Stand der Arbeit / Zeitplan

- Viel Doku geschrieben (17 Seiten) mit ganzem LaTeX aufgesetzt. (sehr zeitaufwändig gewesen)
- Projektplanung inklusive GANTT, Lasten-/Pflichtenheft.
- DevBoard ausgetestet mit Demo Code.
- HW Block Diagramm / Konzept erstellt.
- Schema aufgesetzt und fertiggezeichnet.

Nächste Schritte

- Layout für Extension PCB erstellen.
- Extension PCB bestücken.
- SDRAM und QSPI implementieren.

Änderung der Projektvereinbarung (Bestätigung und vorliegend Begründung)

Keine Änderungen, auch wenn ich in meinem Zeitplan sehr hintendrein bin. Doch ich denke ich kann besonders in der Software gut wieder aufholen und werde Alles geben, wieder „on time“ zu sein.

Probleme, Schwierigkeiten fachlicher Art

eingetretene und noch zu erwartende Probleme, Massnahmen, Lösungen

Direkte Probleme hatte ich bisher nicht. Ich kam immer gut an die Infos, die ich brauchte. Jedoch bin ich im Zeitplan etwas hintendrein, wie oben beschrieben.

Zürich, 03.11.23 Unterschrift Lehrperson ______________________

Zürich, 27.10.23 Unterschrift Lernende ______________________

96_Interview_1.docx 1(1) 03.11.2023 / jschaller

Figure 4.3: Interview 1

4.6 Interview 2

Projektbewertung **MAL / 2109.00** -- HST / BüP
HST / BüP: Projektarbeit 2023 *Sem:* **5** 96_Interview_2.docx TBZ Technische Berufsschule Zürich Abteilung Elektro/Elektronik

Protokoll der Zwischenbesprechung

Zwischenbesprechung **2** Datum: 01.12.23

Name: Joel Schaller Klasse: BEN21a

Titel der Arbeit: CircuitVoyager pre1

Dieses Formular ist soweit möglich ausgefüllt an die Zwischenbesprechung mitzubringen. Weiter sind möglichst alle Unterlagen mitzubringen, vor allem das Geschrieben (Arbeit, Dokumentation, aktuelles Projektjournal, Zeitplan, Notizen, Korrespondenzen) sowie die wichtigsten Informationsmaterialeien (Bücher, Datenblätter etc.)

Stand der Arbeit / Zeitplan

- Layout PCB fertig und sourcing.
- PCBs bestückt.
- PCBs in Betrieb genommen.
- Start Code:
- QSPI Flash
- Dual Core debugging

Nächste Schritte

- Code für TouchGFX
- Messmodi programmieren

Änderung der Projektvereinbarung (Bestätigung und vorliegend Begründung)

- Programm wird QSPI und SDRAM nicht benutzen.

Probleme, Schwierigkeiten fachlicher Art
eingetretene und noch zu erwartende Probleme, Massnahmen, Lösungen

- Konstantstromquelle funktioniert leider nicht wie erwartet, deshalb wird die Widerstandmessung nicht funktionieren. (In vorgegebener Zeit nicht lösbar)

Zürich, 01.12.23 Unterschrift Lehrperson

Zürich, 01.12.23 Unterschrift Lernende

96_Interview_2.docx 1(1) 01.12.2023 / jschaller

Figure 4.4: Interview 2

4.7 Review

TBZ Abt EE / EN Projektarbeit BüP oder HST

Bewertung Projektarbeit technische Produktion

Verfasser/in: **Joel Schaller**

Klasse: **BEN21A**

Titel des BÜP: **CircuitVoyager Pre1**

Bewertungskategorien

5	sehr gut	2	ungenügend
4	gut	1	schlecht
3	genügend	0	nicht vorhanden

1 Erarbeitungsprozess

Kriterien	5	4	3	2	1	0	Kommentar
Projektvereinbarung und Zeitplan Projekt in Grundzügen skizziert, Zeitplan erstellt und bei Bedarf angepasst	✓	❑	❑	❑	❑	❑	Zur Projektidee schon vor dem offiziellen Start recherchiert und detaillierte Planung mit GANTT und 2-wochen Plänen gemacht.
Projektjournal und Reflexion a) Projektjournal: Erarbeitungsprozess nachvollziehbar, Probleme und Fortschritte bezüglich des Produkts ersichtlich	❑	✓	❑	❑	❑	❑	
b) Dokumentation und Visualisierung des Arbeitsprozesses (Vorarbeiten, Entwürfe, Skizzen, technische Versuche, Pläne, Fotos, Film u.a.)	✓	❑	❑	❑	❑	❑	Viele Diagramme, Konzepte und andere Grafiken in Dokumentation eingefügt und aufwändig als Vektorgrafiken mit LaTeX ein Doku eingefügt.
c) Kritische Reflexion des Erarbeitungsprozesses (1.5 bis 2 Seiten)	❑	✓	❑	❑	❑	❑	
Mitarbeit und Kooperation Konstruktive Zusammenarbeit mit Betreuungsperson, Vorbereitung auf vereinbarte Termine, Einhaltung von Vereinbarungen, Führung der Protokolle	✓	❑	❑	❑	❑	❑	Alle Besprechungen immer rechtzeitig abgegeben und unterschrieben. Projekt rechtzeitig abgegeben und bei Rückfragen bei Herrn Malacarne gemeldet.
Total (max. 25 Punkte)							**23/25**

21.12.23 / 2_Review_CircuitVoyager_Pre1_jschaller.docx Seite 1(4)

Figure 4.5: Review page 1

2a Produkt

Kriterien	5	4	3	2	1	0	Kommentar
Konzept / Konstruktionsplan Konzeptidee / Konstruktionsidee geeignet, Methodenwahl sinnvoll	❑	✓	❑	❑	❑	❑	
Intensität der Auseinandersetzung, Innovation, Kreativität	✓	❑	❑	❑	❑	❑	Sehr viel Freizeit in Projekt gesteckt und viele neue Erfahrungen gesammelt. Zusätzlich mal etwas anderes ausprobiert mit Cube und LaTeX.
Resultat a) Einsatz geeigneter Mittel, Materialien und Techniken	✓	❑	❑	❑	❑	❑	Durch Cube/HAL konnte ich sehr effizient programmieren. Zudem bemerkte ich, dass LaTeX ebenfalls meine Effizienz stark erhöht und mit der Einbettung PDFs auch besser als Word für technische Dokumentationen geeignet ist.
b) Erkennbarer Zusammenhang zwischen Konzept und Gestaltung	❑	✓	❑	❑	❑	❑	
c) Ausführungs- und Gestaltungsqualität der Arbeit (handwerklich, technisch)	✓	❑	❑	❑	❑	❑	Qualitatives Endprodukt mit eigenem PCB (anstatt Prototypenaufbau), umfassender Dokumentation und durchdachter Software. (Embedded und Windows App)
d) Aussagekraft in Bezug auf das Thema (interessante, originelle, eigenständige Ideen)	❑	✓	❑	❑	❑	❑	
e) Einsatz der verwendeten Mittel	✓	❑	❑	❑	❑	❑	Problemloser Umgang mit neuen Tools. Siehe auch: Punkt a
Zwischentotal (max. 35 Punkte)							**32/35**

Figure 4.6: Review page 2

TBZ Abt EE / EN Projektarbeit BüP oder HST

2b Schriftlicher Kommentar zum Produkt

Kriterien	5	4	3	2	1	0	Kommentar
Vollständigkeit, Layout und Gestaltung a) Titelblatt, Inhaltsverzeichnis mit Seitenangaben, Gliederung allgemein b) Leserfreundlichkeit, Darstellung, Grafiken, Bilder, Kreativität, Innovation	✓	❑	❑	❑	❑	❑	Übersichtliche und gleichmässig-formatierte Dokumentation mit allen benötigten Verzeichnissen, Referenzen und Links.
Sprache a) Rechtschreibung, Zeichensetzung, Satzbau, Semantik b) Angemessenheit, Leserführung, thematische Entfaltung	✓	❑	❑	❑	❑	❑	Um mich herauszufordern habe ich die Dokumentation und Präsentation auf English gemacht. Da dies meine Zweitsprache ist, wird der Text nicht so akkurat sein, wie er in Deutsch gewesen wäre, jedoch konnte ich viel dazulernen.
Abstract Einhaltung der formalen Kriterien, Erwähnung von gestalterischem Konzept oder Konstruktionsplan, Methode und wichtigsten Ergebnissen	❑	✓	❑	❑	❑	❑	
Einleitung Vorstellung von Thema und eigenem gestalterischen Konzept oder Konstruktionsplan in Bezug auf das Oberthema, Darlegung der Motivation für die Arbeit, Erläuterung der verwendeten Methoden, Überblick über den schriftl. Kommentar	✓	❑	❑	❑	❑	❑	Detaillierte Einleitung mit Pflichtenheft, Lastenheft, Mindmap, HW-Konzept. Zusätzlicher Text zu Motivation für dieses Projekt.
Hauptteil a) Logischer Aufbau und Zusammenhalt, Informationsgehalt, sachliche Richtigkeit, Konzentration auf das Thema, sinnvolle Gewichtung der einzelnen Aspekte, Auswahl der Quellen b) Eigene Fragestellung klar und sinnvoll formuliert und in Themenbereich eingebettet, Vorgehen nachvollziehbar beschrieben, Originalität, Eignung der methodischen Vorgehensweise c) Präsentation der Ergebnisse in geeigneter Form, Verknüpfung der einzelnen Elemente der Arbeit, Einbettung der Quellen in die Arbeit, Entwicklung eines eigenen Standpunktes, selbstständige und kritische Auseinandersetzung mit Vorgehen, Quellen und Ergebnissen	✓	❑	❑	❑	❑	❑	Sehr detaillierte und aussagevolle Dokumentation mit vielen Grafiken und Quellen.
Schlusswort Gesamtschau, Arbeitsergebnis, Gesamturteil, evtl. Ausblick	✓	❑	❑	❑	❑	❑	Zusätzlicher Text, wegen nicht Fortführens des Projektes.
Zitate und Quellenverzeichnis a) Quellen (Bilder und Zitate) innerhalb der Arbeit korrekt ausgewiesen b) Korrektes Quellen-/ Abbildungs-/ Tabellenverzeichnis	✓	❑	❑	❑	❑	❑	Alle Verzeichnisse korrekt und automatisch mit LaTeX generiert.
Zwischentotal (max. 35 Punkte)							**34/35**

21.12.23 / 2_Review_CircuitVoyager_Pre1_jschaller.docx Seite 3(4)

Figure 4.7: Review page 3

3 Präsentation

Kriterien	5	4	3	2	1	0	Kommentar
Inhalt a) Einleitung anregend, weckt Interesse, spricht Hörer an, gibt Überblick	☐	☐	☐	☐	☐	☐	
b) Aufbau logisch, repräsentativer Einblick in Arbeit (struktureller Aufbau), verständliche Darlegung der Eigenleistung und der verwendeten Methoden, Verknüpfung der einzelnen Elemente, Zusammenfassung der wichtigsten Punkte zur Abrundung	☐	☐	☐	☐	☐	☐	
c) Sachkompetenz, Fragen kompetent beantwortet	☐	☐	☐	☐	☐	☐	
Sprache Klar und verständlich, Lautstärke, Intonation klar und korrekt, sprachlich korrekt, angemessen	☐	☐	☐	☐	☐	☐	
Auftreten Freies Sprechen, Kontakt zu Zuhörenden, Präsenz, Ausdrucksweise lebendig, Körperhaltung, Mimik und Gestik, Gesprächsführung	☐	☐	☐	☐	☐	☐	
Medienkompetenz Sinnvoller Einsatz und übersichtliche Gestaltung von Anschauungsmaterial (Skizzen, Modelle, Entwürfe, Recherchematerial etc.) bzw. von visuellen Präsentationsmitteln (AV-Medien)	☐	☐	☐	☐	☐	☐	
Total (max. 30 Punkte)							

Notenberechnung

Teil	Punktzahl
Erarbeitungsprozess	23/25
Produkt	66/70
Präsentation	--/30
Gesamtpunktzahl	--/125
Note BÜP	

Zürich,______________ Unterschrift Lehrperson ____________________________________

Figure 4.8: Review page 4

4.8 Extension PCB Schematics

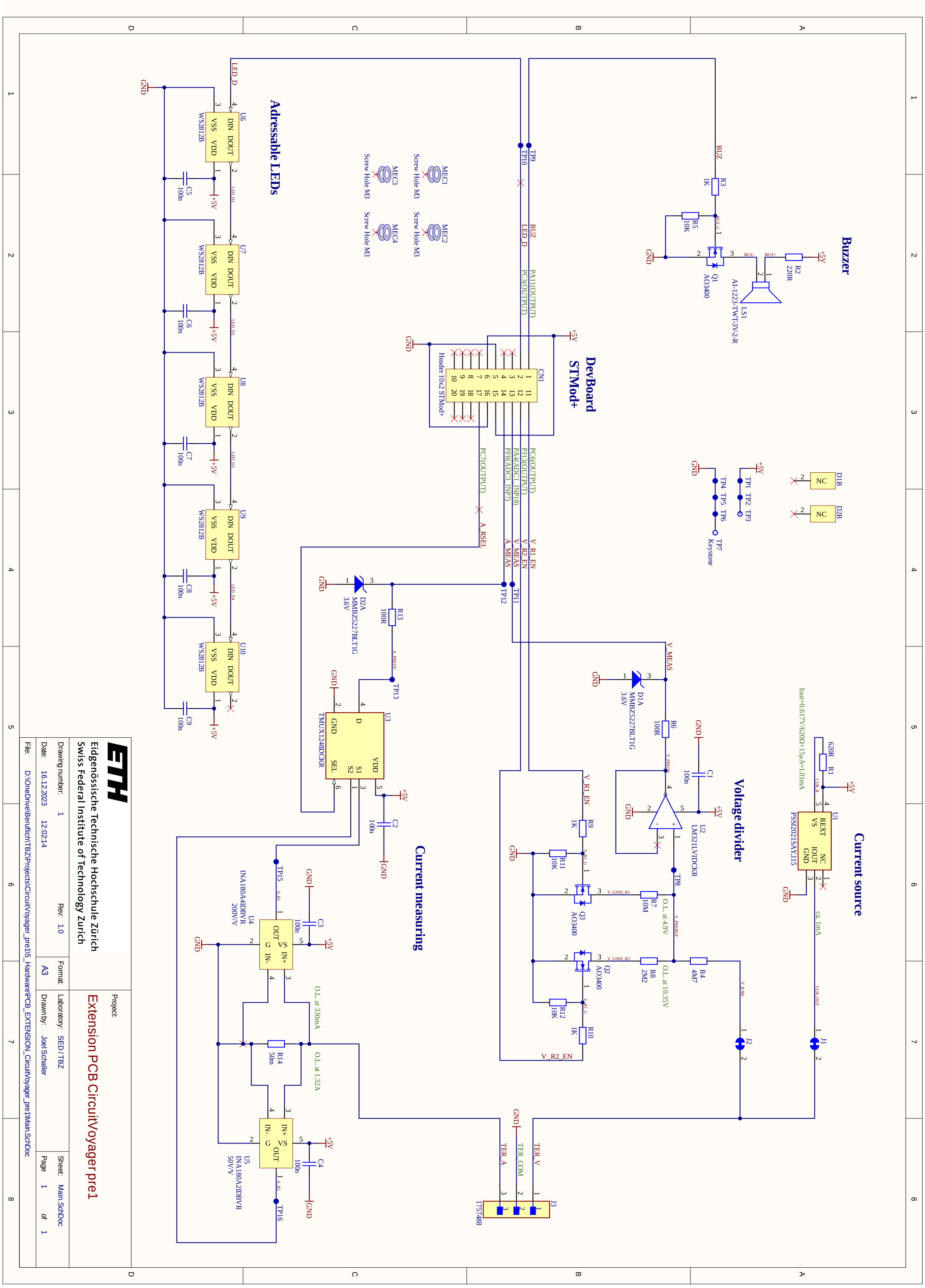


Figure 4.9: Extension PCB Schematics

4.9 Extension PCB Manufacture 11.23 BOM

Designator	Comment	Value	Description	Quantity	Digi-Key Part number
C1, C2, C3, C4, C5, C6, C7, C8, C9	C0402	100n	CERCAP 100n 16V 5% X7R 0402	9	311-1338-1-ND
CN1	Header 10x2 STMod+	Header 10x2	Header 10x2, 2mm, RA, TH, Height 3.20mm/3.	1	2073-BF060-20A-B-0200-0280-0305-L-D-ND
D1, D2	MMBZ5227BLT1G		Zener Voltage Regulator, 225 mW, 3-Pin SOT-2	2	MMBZ5227BLT1GOSCT-ND
J3	1757488		TERM BLOCK HDR 3POS 90DEG 5MM	1	
LS1	AI-1223-TWT-3V-2-R		BUZZER MAGNETIC 3V 12MM TH	1	668-1456-ND
Q1, Q2, Q3	AO3400		N-channel Enhancement Mode Field-Effect Tra	3	
R1	R0402	620R	RES 620R 62.5mW 1% 0402	1	311-620LRCT-ND
R2	R0402	220R	RES 220R 62.5mW 1% 0402	1	311-220LRCT-ND
R3, R9, R10	R0402	1K	RES 1K 62.5mW 1% 0402	3	311-1.00KLRCT-ND
R4	R0402	4M7	RES 4M7 62.5mW 1% 0402	1	13-RC0402FR-074M7LCT-ND
R5, R11, R12	R0402	10K	RES 10K 62.5mW 1% 0402	3	311-10.0KLRCT-ND
R6, R13	R0402	100R	RES 100R 62.5mW 1% 0402	2	311-100LRCT-ND
R7	R0402	10M	RES 10M 62.5mW 1% 0402	1	311-10.0MLRCT-ND
R8	R0402	2M2	RES 2M2 62.5mW 1% 0402	1	13-RC0402FR-072M2LCT-ND
R14	R0805	50m	RES 0R05 125mW 1% 0805	1	13-PE0805FRF070R05LCT-ND
U1	PSSI2021SAY,115		NEXPERIA - PSSI2021SAY,115 - IC, CURRENT SC	1	1727-4328-1-ND
U2	LM321LVIDCKR		IC OPAMP GP 1 CIRCUIT SC70-5	1	296-53426-1-ND
U3	TMUX1248DCKR		3- LOW RON, 5-V, 2:1 (SPDT) GENE	1	296-TMUX1248DCKRCT-ND
U4	INA180A4IDBVR	200V/V	Operational Amplifier, 1 Func, 500uV Offset-M	1	296-47655-1-ND
U5	INA180A2IDBVR	50V/V	Operational Amplifier, 1 Func, 500uV Offset-M	1	296-47653-1-ND
U6, U7, U8, U9, U10	WS2812B		Intelligent Control LED Integrated Light Source	5	
TP7	Keystone		TP SMD 3.81x 2.03mm	1	36-5019CT-ND

Digi-Key
Already supplied

Figure 4.10: Extension PCB Manufacture 11.23 BOM

```c
#include "usermain_CM7.h"

void usermain_cm7(void)
{
    //init local variables
    uint8_t uartBuf[50] = "";
    uint8_t uartInBuf[1] = "";

    uint8_t VMsg[7] = "";
    uint8_t AMsg[7] = "";
    uint8_t WMsg[8] = "";

    double voltage = (double)0;
    double current = (double)0;

    uint8_t noMsgCnt = 5;
    uint8_t connected = 0;

    uint32_t colorConnectionState = CV_COLOR_BLACK;

    CV_VRange_Enum VRange = CV_VRange_1__4_9V;
    CV_ARange_Enum ARange = CV_ARange_1__0_330A;

    //Set LED Ring to red
    for(int i = 0; i < 5; i++)
    {
        CV_LED_BUF[i] = colorConnectionState;
    }
    CV_LED_SendBuf();

    //Turn on green LED on DevBoard, to indicate that CM7 has
    booted.
    HAL_GPIO_WritePin(LED_G_GPIO_Port, LED_G_Pin, !1);

    //Start in voltage range 1 (O.L. at 4.9V)
    CV_VRange_Set(VRange);

    //Start in current range 1 (O.L. at 330mA)
    CV_ARange_Set(ARange);
```

Listing 4.1: CV FW 1

```c
    //main loop
    while(1)
    {
        if((HAL_UART_Receive(&huart1, uartInBuf, 1, 100) == HAL_OK) && (uartInBuf[0] == '1')) noMsgCnt = 0;
        else if(noMsgCnt < 10)                          noMsgCnt++;

        if(noMsgCnt >= 2)
        {
            colorConnectionState = CV_COLOR_BLACK;
            if(connected == 1) CV_Buz_BeepBlocking(150);
            connected = 0;
        }
        else
        {
            colorConnectionState = CV_COLOR_GREEN;
            if(connected == 0)
            {
                CV_Buz_BeepBlocking(125);
                HAL_Delay(50);
                CV_Buz_BeepBlocking(125);
            }
            connected = 1;
        }



        //get voltage reading from adc1 (V_MEAS pin)
        voltage = CV_VMeas_RawBlocking() * CV_VRange_Factor[VRange];

        if((VRange == CV_VRange_1__4_9V) && (voltage > CV_VThreshold_R12))
        {
            CV_Buz_BeepBlocking(5);
            VRange = CV_VRange_2__10_35V;
            CV_VRange_Set(VRange);
            HAL_Delay(1);
            voltage = CV_VMeas_RawBlocking() * CV_VRange_Factor[VRange];
        }
```

Listing 4.2: CV FW 2

```c
        else if((VRange == CV_VRange_2__10_35V) && (voltage < CV_VThreshold_R21))
        {
            CV_Buz_BeepBlocking(5);
            VRange = CV_VRange_1__4_9V;
            CV_VRange_Set(VRange);
            HAL_Delay(1);
            voltage = CV_VMeas_RawBlocking() * CV_VRange_Factor[VRange];
        }


        //get current reading from adc3 (A_MEAS pin)
        current = CV_AMeas_RawBlocking() * CV_ARange_Factor[ARange];

        if((ARange == CV_ARange_1__0_330A) && (current > CV_AThreshold_R12))
        {
            CV_Buz_BeepBlocking(5);
            ARange = CV_ARange_2__1_32A;
            CV_ARange_Set(ARange);
            HAL_Delay(1);
            current = CV_AMeas_RawBlocking() * CV_ARange_Factor[ARange];
        }
        else if((ARange == CV_ARange_2__1_32A) && (current < CV_AThreshold_R21))
        {
            CV_Buz_BeepBlocking(5);
            ARange = CV_ARange_1__0_330A;
            CV_ARange_Set(ARange);
            HAL_Delay(1);
            current = CV_AMeas_RawBlocking() * CV_ARange_Factor[ARange];
        }
```

Listing 4.3: CV FW 3

```c
        //format output buffer with voltage, current and power reading.
        if(voltage < CV_V_OL) sprintf((char *)VMsg, "%04.2fV\t", voltage);
        else                  sprintf((char *)VMsg, "O.L.\t");

        if(current < CV_A_OL) sprintf((char *)AMsg, "%04.2fA\t", current);
        else                  sprintf((char *)AMsg, "O.L.\t");

        if((voltage < CV_V_OL) && (current < CV_A_OL))
        {
            sprintf((char *)WMsg, "%04.2fW\t", voltage * current);
            for(int i = 0; i < 5; i++)
            {
                CV_LED_BUF[i] = colorConnectionState;
            }
        }
        else
        {
            sprintf((char *)WMsg, " O.L. \t");
            for(int i = 0; i < 5; i++)
            {
                CV_LED_BUF[i] = CV_COLOR_RED;
            }
        }

        sprintf((char *)uartBuf, "%s%s%sVR:%d\tAR:%d\n", VMsg, AMsg, WMsg, VRange, ARange);

        //Output buffer over UART 1. (ST-Link UART)
        HAL_UART_Transmit(&huart1, uartBuf, strlen((char *)uartBuf), HAL_MAX_DELAY);

        //Update LEDs
        CV_LED_SendBuf();

        //wait 250ms (aprox. 4Hz cycle frequency)
        HAL_Delay(250);
    }
}
```

Listing 4.4: CV FW 4

Bibliography

[1] ElectroNoobs, "Arduino 5 in 1 multimeter v2.0." Available at http://electronoobs.com/eng_arduino_tut112.php (2019/11/29).

[2] A. Walsh, "Protecting adc inputs." Available at https://www.analog.com/en/technical-articles/protecting-adc-inputs.html.

[3] A. Garaipoom, "Digital multimeter circuit using icl7107." Available at https://www.eleccircuit.com/digital-multimeter-circuit-using-icl7107/ (2022/08/16).

[4] TexasInstruments, "An-202 a digital multimeter using the add3501." Available at https://www.ti.com/lit/an/snoa592c/snoa592c.pdf?ts=1698230599203&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fde-de%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Dde-DE%2526searchTerm%253Dmultimeter%2526nr%253D5689 (2015/02).

[5] ControllersTech, "Stm32 dual core. getting started with stm32 dual core cpus." Available at https://www.youtube.com/watch?v=jI1k6p-fduE&list=PL6W_Zc9OpVoEW7mhVlfjzXFTc9skCltyI&index=1 (2021).

[6] ControllersTech, "Qspi in stm32, write and read, n25q." Available at https://www.youtube.com/watch?v=xIfh_uYy-OU (2021).

[7] STMicroelectronics, "Github: Qspi flash drivers for stm32." Available at https://github.com/STMicroelectronics/stm32-external-loader/tree/contrib/QSPI_Drivers/MT25QL512 (2020/11/03).

[8] m3y54m, "Github: Python serial port + tkinter gui." Available at https://github.com/m3y54m/python-serial-port-gui/tree/master (2023/08/08).

List of Figures

1.1 Project Mindmap 2

2.1 STMod+ Interface 4
2.2 Extension PCB HW concept 5
2.3 Buzzer Circuit 6
2.4 Adressable LEDs Circuit 7
2.5 Overvoltage Protection Circuit 8
2.6 Voltage Divider Circuit 9
2.7 Current Source Circuit 10
2.8 Current Measuring Circuit 11
2.9 Extension PCB Layout 12
2.10 Extension PCB Assembly Drawing 13
2.11 Extension PCB 13
2.12 Commissioning Lab 14
2.13 Commissioning PCB1 current measurements 15
2.14 Commissioning PCB1 voltage measurements 15
2.15 Hardware Boot Button Mod 20

4.1 GANTT Chart 30
4.2 Project Agreement 31
4.3 Interview 1 32
4.4 Interview 2 33
4.5 Review page 1 34
4.6 Review page 2 35
4.7 Review page 3 36
4.8 Review page 4 37
4.9 Extension PCB Schematics 38
4.10 Extension PCB Manufacture 11.23 BOM 39

List of Tables

2.1 Technical Details 3
2.2 commissioning PCB 1 14
2.3 LED Annotation 18

4.1 Project Journal 1 26
4.2 Project Journal 2 27

Listings

2.1 Blink Code CM4 18
2.2 QSPI Write Test CM7 22

4.1 CV FW 1 40
4.2 CV FW 2 41
4.3 CV FW 3 42
4.4 CV FW 4 43

Acronyms

CircuitVoyager The Name of the DMM I'm developing.

DevBoard main microcontroller developement board. (STM32H747I-Disco)

DMM digital multimeter

HW Hardware

SW Software

SPI Serial Peripheral Interface (low level protocol)

SDRAM Synchronous Dynamic Random Access Memory (external RAM)

UI User Interface

Mipi DSI Digital Serial Interface (Display Protocol)

FAT File Allocation System (Low Level Filesystem)

HAL Hardware Abstraction Layer (STM32 Abstraction Library)

ETHZ Eidgenössische Technische Hochschule

TBZ Technische Berufsschule Zürich

ADC Analog Digital Converter (STM32 unit)

TIM Timer (STM32 unit)

DUT Device under test

OVP Over voltage protection

TouchGFX Graphical UI designer for STM32 **mcu!**s

QPSI Quad SPI

OL Overload

RCC Reset and Clock Control (STM32 unit)

CM4 Cortex M4 (CPU)

CM7 Cortex M7 (CPU)

XIP Execute In Place

MSP Main Stack Pointer

AWG Arbitrary Waveform Generator