Masterthesis

Project Documentation

Benjamin Aigner (Supervisor), Vanessa Prankl (Student)

Prerequisits

This masterthesis is part of the ASTERICS-Project funded by the European Union.

Prerequisites:

- SW stack: ASTERICS-project “Mouse/Keyboard for BLE HID” (https://github.com/asterics/esp32_mouse_keyboard)

Reference Product Testing

GlassousePro has been tested to set some usability benchmarks. It has a well thought out design which enables the user to simply plug-and-play when turned on. The following observations have been made during testing its features:

- **Startup:** The device needs to be calibrated after startup. This requires to place it on a flat surface and wait a few seconds. Otherwise the mouse always drifts.
- **Power source:** 250 mAh battery
- **Connection types with host device:** Bluetooth only
- **4 connectors for individual external buttons** that are used to enable right- and left-click as well as scrolling and switching the connected device.
- **1 button** for switching the connected device (PC, TV, Tablet, …)
- **1 button** for switching the mouse pointer sensitivity (3 sensitivities available)
- **1 LED** for battery charge status
- **3 LEDs** to show, which device is connected

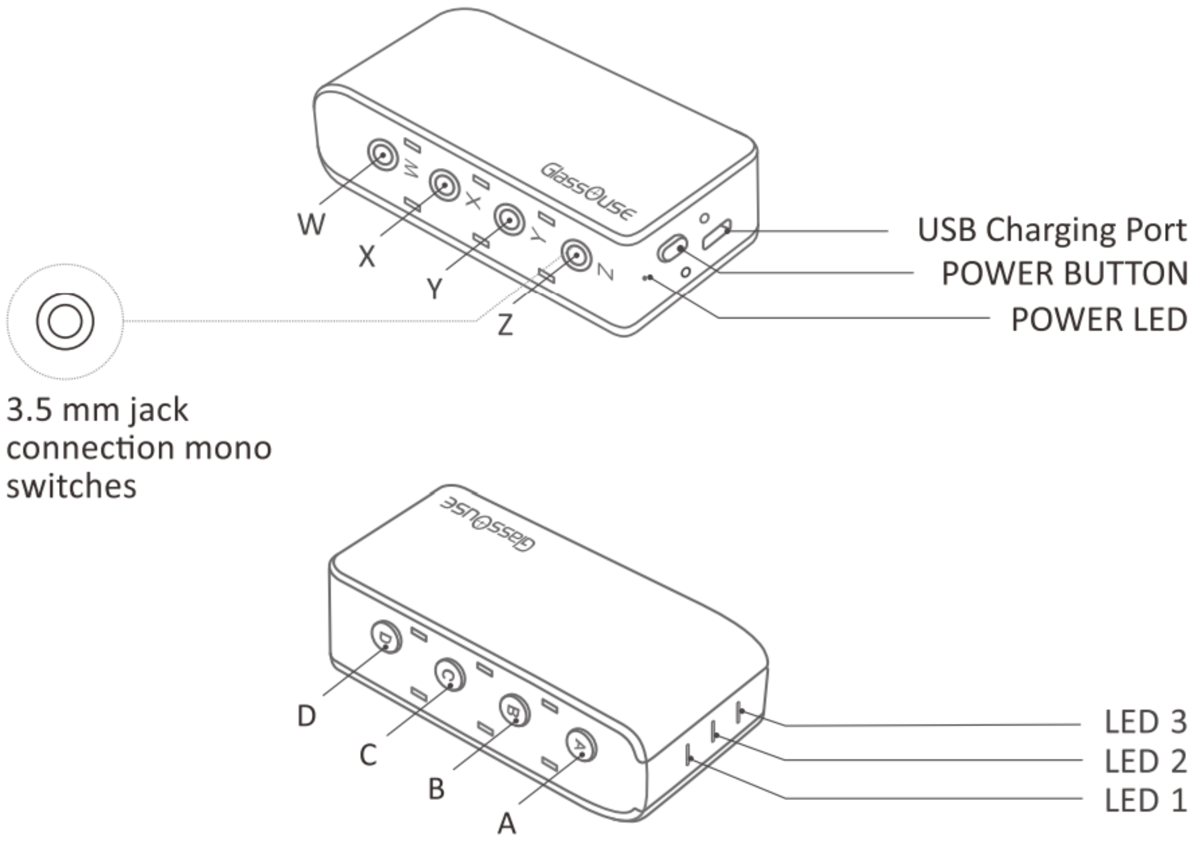

Figure 1: https://glassouse.com/wp-content/uploads/2022/03/GlassOuse-Pro-user-manual-english.pdf

Furthermore, Glassouse uses a 9-DOF IMU, which includes a magnetometer in order to reduce mouse pointer drift.

Device Feature Definition

The HeadMouse device shall support the following features:

- Bluetooth communication
- Battery powered (rechargeable)
- UI:
    - 1 LED for device status
    - 1 LED for battery status
    - 3 jacks for user input buttons (left-,right-click, scroll)
    - Sip & puff sensor input for alternative control mechanism
- 6+9 axis IMU for comparison

Hardware Documentation

The goal is to implement a cost effective HeadMouse Design which supports all necessary features a computer mouse has.

Schematic Design V1

Power Management and Battery Charging

The device shall be powered either by USB-C connector or a battery. If both are attached, the device shall continue operation. The voltage regulator TC1262 ensures a constant voltage source for the hardware components on the PCB.

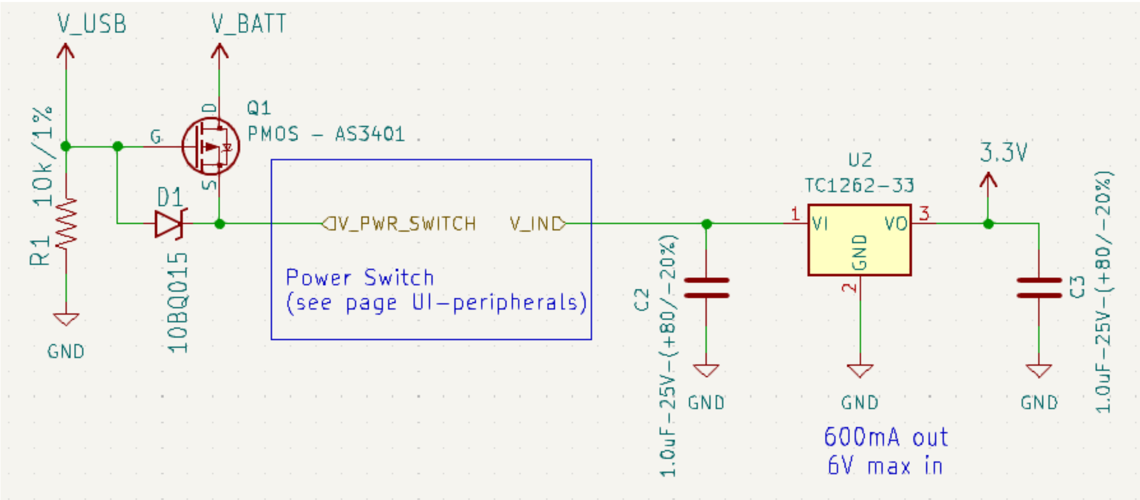

Figure 2: Power management schematic

Currently, a 500 mAh battery is implemented. The duration until the battery needs to be recharched shall be evaluated during the project. The design is based on the board design from ESP32Thing from Sparkfun ([https://cdn.sparkfun.com/assets/learn_tutorials/5/0/7/esp32-thing-schematic.pdf)](https://cdn.sparkfun.com/assets/learn_tutorials/5/0/7/esp32-thing-schematic.pdf)

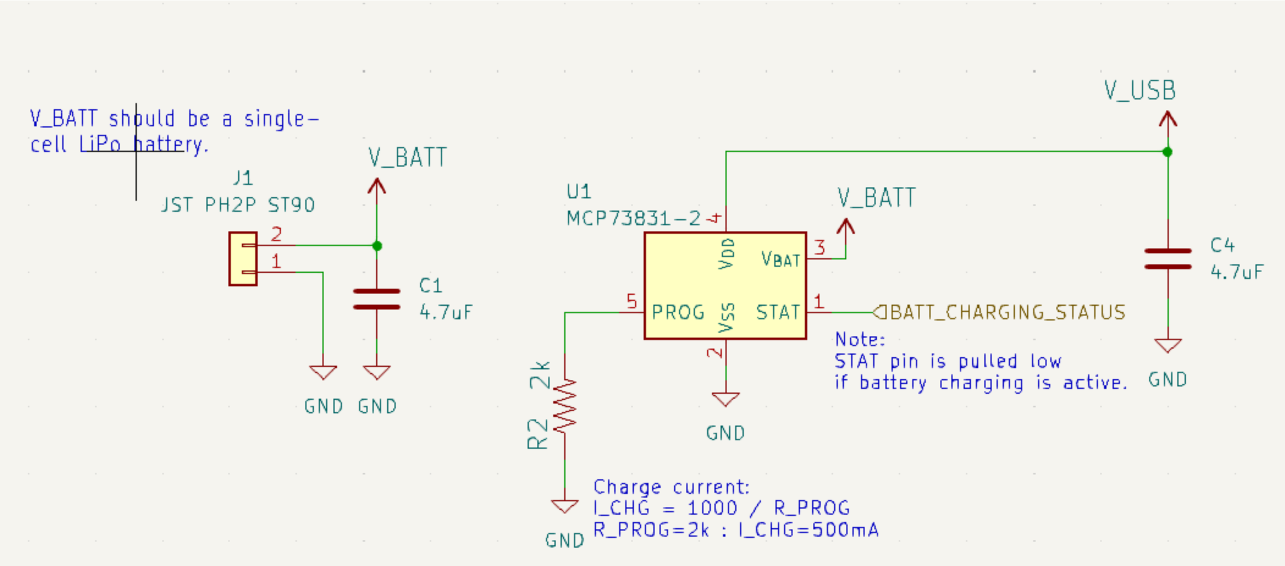

Figure 3: Battery management schematic

The battery charge level as well as the charging status are measured by the microcontroller, in order to give the user some feedback about the device status.

Microcontroller

Basic requirements for the microcontroller are:

- Bluetooth support
- Arduino compatible due to existing Arduino code
- 32 Bit architecture to ensure a modern and lasting design for the next years
- USB support (not only UART as early on)

Several microcontrollers fulfill these requirements at the moment. I chose an ESP32-S3-WROOM which is a hardware module including the newest generation of ESP32 and a Bluetooth antenna. This makes the implementation of the microcontroller easier and is one of the cheapest currently available modules.

Some sources for the decision:

- Discussion about alternatives to ESP32 (Arduino support is required so currently only RPPico W is an option): https://www.espboards.dev/blog/esp32-alternatives/

- Note: ESP32 first generation is not recommended for new designs by ESP and needs USB-UART converter. S2 generation does not have bluetooth support => Use S3 generation

- Difference between old ESP32 and new ESP32-S2 and ESP8266 (ESP-S3 missing): https://maker.pro/esp8266/tutorial/a-comparison-of-the-new-esp32-s2-to-the-esp32

- ESP-S3-WROOM Documentation: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

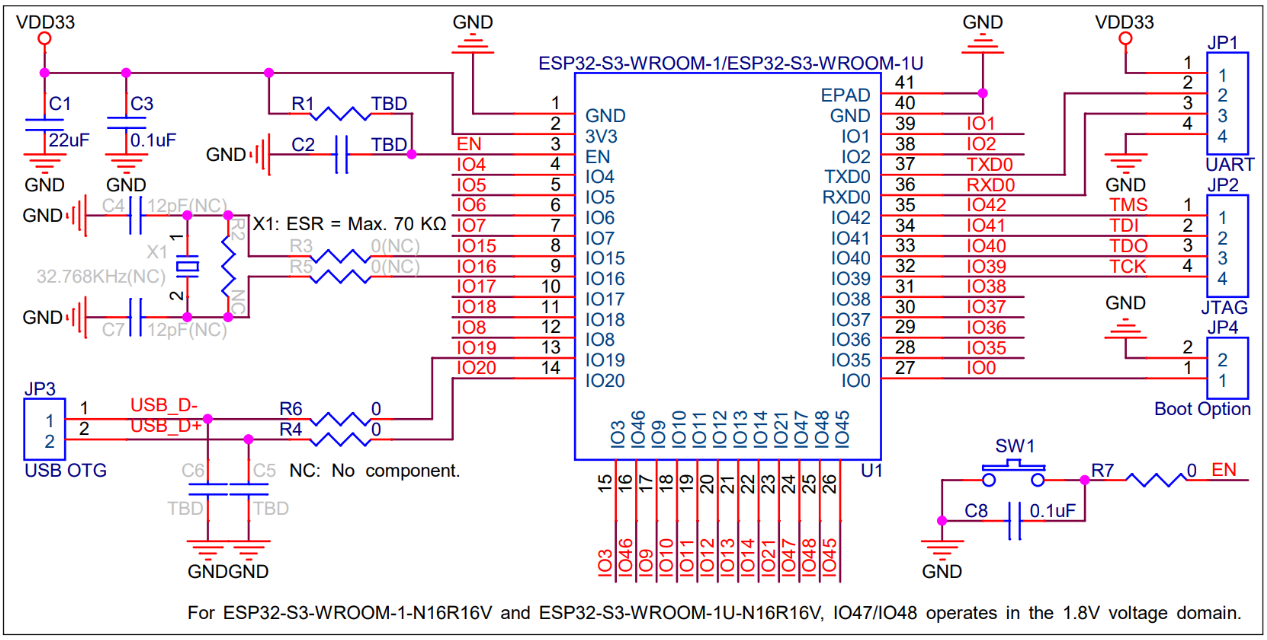

Figure 4: Peripheras schematics of ESP32-S3-WROOM

To enable communication over USB on the HeadMouse board, a USB-C connector is placed on the board.

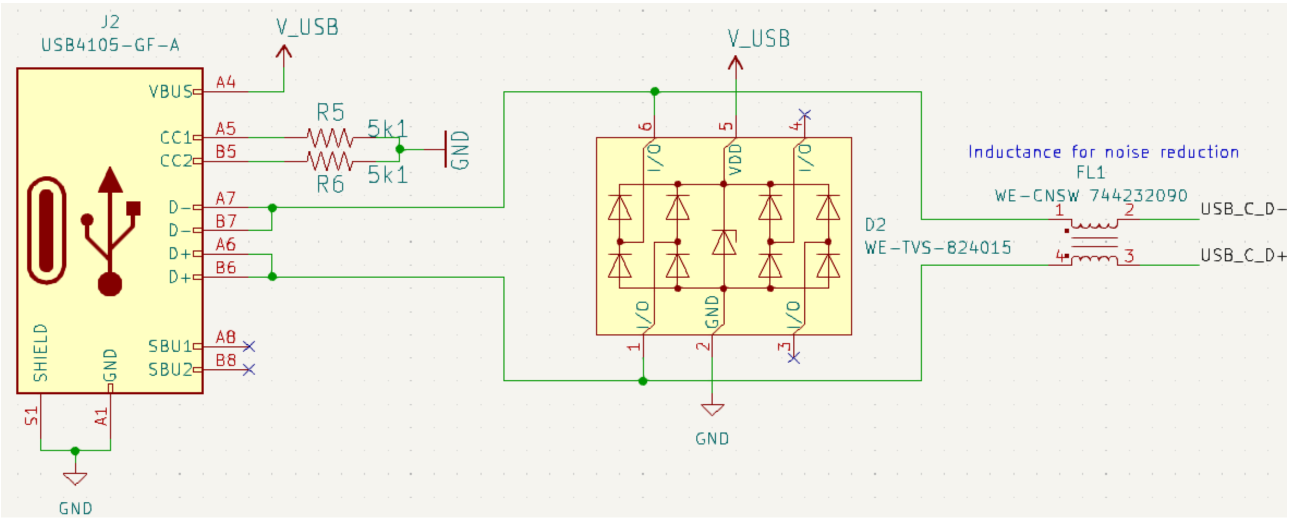

Figure 5: USB-C connector schematic

User Interface

Two low power (<10 mA) bi-color-LED for side surface mount were chosen ([https://www.mouser.at/ProductDetail/Wurth-Elektronik/155124RV73200?qs=5aG0NVq1C4yQQzqaqWGYAg%3D%3D)](https://www.mouser.at/ProductDetail/Wurth-Elektronik/155124RV73200?qs=5aG0NVq1C4yQQzqaqWGYAg%3D%3D)

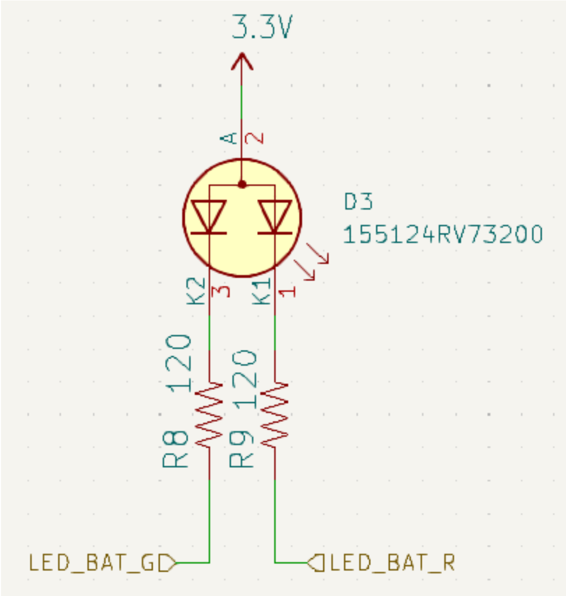

Figure 6: LED schematic

Auxillary button jacks are used to enable external buttons to connect.

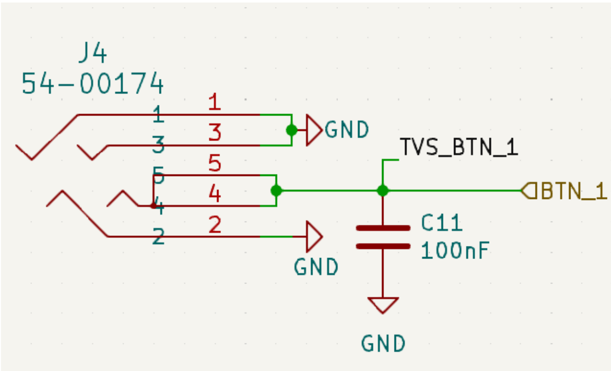

Figure 7: Button jack for external buttons schematic

IMUs

In order to make some benchmark tests, two different 6-axis IMUs and one 9-axis IMU are implemented on the first prototype. Currently, most 9-axis IMUs are not available at the market which reduced the number of options to a single IMU type.

Source of IMU comparison: https://oscarliang.com/flight-controller-explained/#Gyro

1) **6-Axis IMU:** MC6470

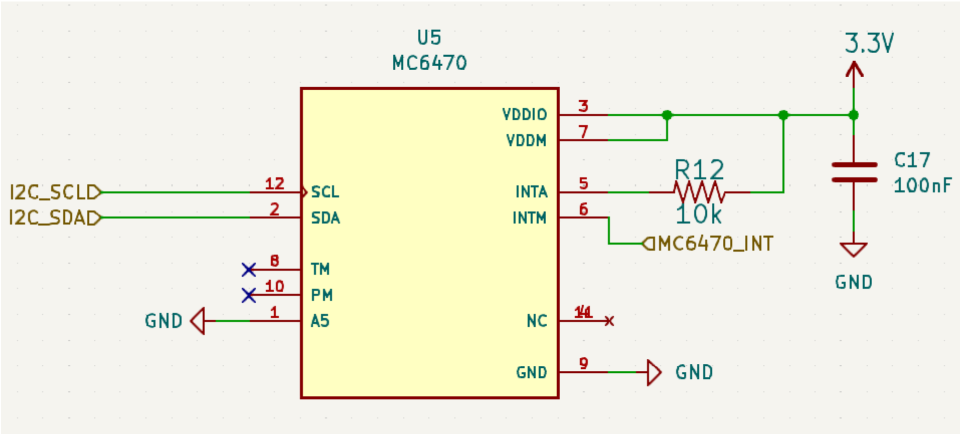

Figure 8: IMU MCP6470 schematic

2) **6-Axis IMU:** LSM6DS (same as on Arduino Nano RP2040 Connect) https://content.arduino.cc/assets/ABX00053-schematics.pdf)

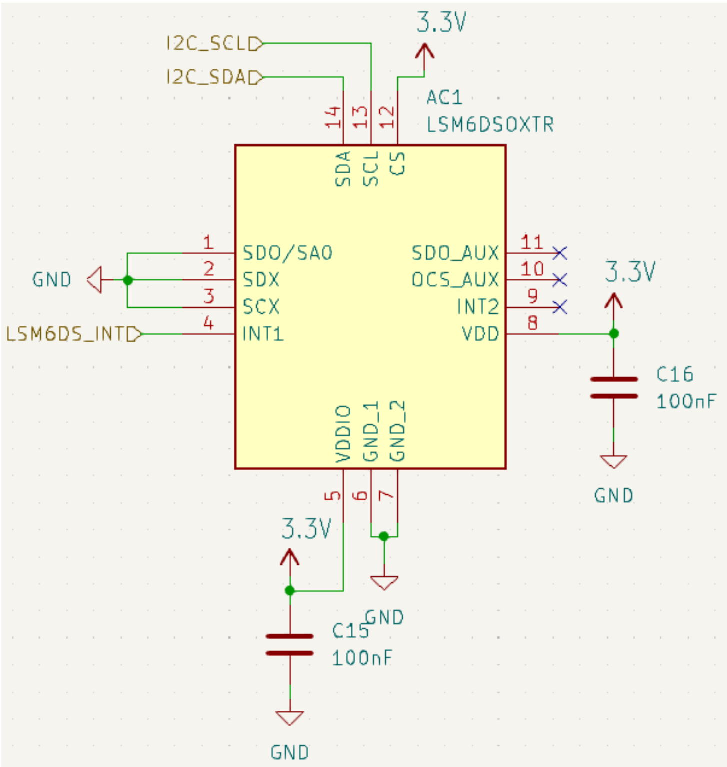

Figure 9: IMU LSM6DS schematic

3) **9-Axis IMU:** BNO55

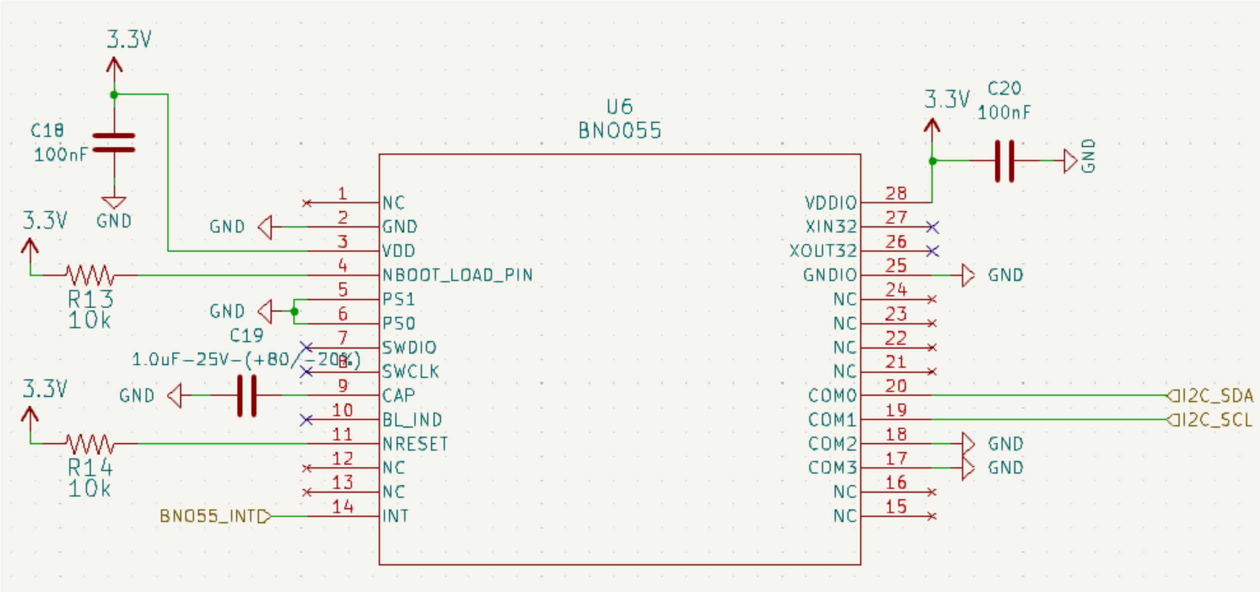

Figure 10: IMU BNO55 schematic

PCB Testing V1

A simple peripheral test program was written for Arduino and flashed on the HeadMouse board V1. Three boards are assembled and tested. For each board a test-report was generated.

The test program includes:

- I2C port scan
- Read out all button inputs
- Read out battery charging status
- Measure battery voltage

```
=== I2C Scanner ===
Device at address: 0x0C
Device at address: 0x4C
Device at address: 0x6A
Device at address: 0x7E
I2C scan finished

Button values: 1, 1, 1, 1
Battery charging status: 0
Battery voltage ADC value: 4095
```

Figure 11: Example for peripheral test program output

General observation notes:

- The battery poles are swapped on pcb and must be corrected.
- The battery voltage measurement needs to be done with an adjustable voltage source to benchmark the ADC-measurements.
- The I2C bus-scan results in 4 instead of 3 (assumed) found devices. Nevertheless, only 3 out of 4 IMUs are found. It needs to be determined, why 2 more devices show up (and not only 1) and why the third IMU cannot be found.
- Bluetooth testing is still open and will be implemented later on.

Test Report Board V1-pcb1-test1

Hardware Component	Status	Notes
ON/OFF switch	tested - ok	
Manual boot over button	tested - ok	
Manual reset over buttons	tested - ok	
uC flashing	tested - ok	
Autonomous reboot after flashing	tested - ok	
Communication uC-PC over USB-C	tested - ok	
Communication uC-PC over Bluetooth	TBD	Will be tested later on
uC powered by battery	tested - does not work	Battery connectors are swapped on PCB -> Killed battery management chip. Chip needs to be replaced
Battery charging	tested - does not work	According chip is dead
LED (battery status)	tested - ok	
LED (device status)	tested - does not work	Always keeps dark -> Soldering issue needs to be solved
Button 1	tested - ok	
Button 2	tested - ok	
Button 3	tested - ok	
Button 4	tested - ok	
IMU-BNO55 addressing over I2C	tested - does not work	Should have I2C addr.: 0x28 but only 0x0C and 0x7E show up. Source of issue needs to be determined
IMU-LSM6DS addressing over I2C	tested - ok	I2C addr.: 0x6A
IMU-MCP5479 addressing over I2C	tested - ok	I2C addr.: 0x4C

Test Report Board V1-pcb2-test1

Hardware Component	Status	Notes
ON/OFF switch	tested - ok	
Manual boot over button	tested - ok	
Manual reset over button	tested - ok	
uC flashing	tested - ok	
Autonomous reboot after flashing	tested - ok	
Communication uC-PC over USB-C	tested - ok	
Communication uC-PC over Bluetooth	TBD	Will be tested later on
uC powered by battery	tested - does not work	uC does not boot if powerd by battery. Issue source needs to be determined
Battery charging status measurement	tested - ok	Tested with battery connected and not attached
Battery voltage measurement	TBD	Could not be tested yet due to missing adjustable voltage source.
LED (battery status)	tested - ok	
LED (device status	tested - ok	
Button 1	tested - ok	
Button 2	tested - ok	
Button 3	tested - does not work	Input pin of uC always stays high - Soldering issue is likely and needs to be determined and fixed
Button 4	tested - ok	
IMU-BNO55 addressing over I2C	tested - does not work	Should have I2C addr.: 0x28 but only 0x0C and 0x7E show up. Source of issue needs to be determined
IMU-LSM6DS addressing over I2C	tested - ok	I2C addr.: 0x6A
IMU-MCP6470 addressing over I2C	tested - ok	I2C addr.: 0x4C

Test Report Board V1-pcb3-test1

Hardware Component	**Status**	**Notes**
ON/OFF switch	tested - ok	
Manual boot over button	tested - ok	
Manual reset over buttons	tested - ok	
uC flashing	tested - does not work	uC does not mount -> Soldering issue very likely and needs to be resolved
Autonomous reboot after flashing	TBD	
Communication uC-PC over USB-C	TBD	
Communication uC-PC over Bluetooth	TBD	
uC powered by battery	TBD	
Battery charging	TBD	
LED (battery status)	TBD	
LED (device status	TBD	
Button 1	TBD	
Button 2	TBD	
Button 3	TBD	
Button 4	TBD	
IMU-BNO55 addressing over I2C	TBD	
IMU-LSM6DS addressing over I2C	TBD	
IMU-MCP5479 addressing over I2C	TBD	