

Engineering Report – Battery Capacity Test Device LilonCheck9V125mA

Version 1.a

Date: November 4, 2022

Signature <u>GF</u> <hr/> Gerald Fischer	Signature <u>CHR</u> <hr/> Christoph Ragonig
Signature BP <hr/> Bernhard Pfeifer	-

Revision History:

v1.a creation of document

This document was created for training in the field of technical documentation. It describes a test device, which was designed and built for assessing the capacity of (rechargeable) batteries. It is assumed that the batteries are used in a fictive medical product. Therefore, testing must meet strict requirements.

Training settings in an academic environment can never perfectly match the situation in an industrial environment. Throughout this document blue text is used for highlighting unavoidable restrictions.

Contents

1. Purpose	4
2. User-Requirement	4
3. Architecture	5
4. Specifications	6
4.1. Manufacturer Specifications	6
4.2. Test Specifications & Risk Mitigation	7
4.3. USB Supply	7
5. Design	8
5.1. Test Phases	8
5.2. Hardware	9
5.3. Accuracy	13
5.4. Test Recording	13
6. Verification & Validation	14
6.1. V& V of Test Specifications and Accuracy	14
6.2. Validation of User Requirement	16
7. Summary & Conclusions	17
8. Documentation & Literature	18
8.1. Internal Documents	18
8.2. Standards and Guidelines	18
8.3. Datasheets & Test Reports	18
Appendices	19
A. Risk-Management	20
A.1. Operator-Protection	20
A.2. Protection of the Battery-Pack Under Test	20
A.3. Test Reliability	22
B. Shield Design	23
B.1. Circuit Design & Calculation	23
B.1.1. Design Current Sink	23
B.1.2. Design Reference Voltage	26
B.1.3. Room Temperature Sensor	27
B.1.4. Duo-LED	27
B.2. Tolerances	28
B.2.1. Uncertainty in Reference Voltage	28
B.2.2. Uncertainty in AD-conversion	28
B.2.3. Uncertainty in Battery Data	29
B.2.4. Uncertainty in Temperature	29
B.3. Risk Mitigation	29

B.4. Eagle Circuit & Board Layout	30
B.5. Shield Prototypes	31
B.5.1. Test Results	31
B.5.2. Interpretation of Shield Test	33
C. Firmware Design	36
C.1. Controller Design	36
C.1.1. Analog Input – Physical Parameters	36
C.1.2. PI-Control	37
C.2. Risk Mitigation	40
C.3. Firmware Implementation	40
C.3.1. Routines setup() and loop()	41
C.3.2. Routine RunWaitingPhase01	42
C.3.3. Routine RunVoltageTest01	42
C.3.4. Routine RunDischargePhase01	42
C.3.5. Routines for Measurement & Control	43
C.3.6. Output Routines	44
C.4. Test Results	44
C.5. Listing	49
D. Inspection, Measuring, and Test Equipment (IMTE)	58
D.1. Multimeters	58
D.2. Other Test Devices	58
E. Miscellaneous Battery Types	61
E.1. Primary Batteries	61
E.1.1. Zinc-Carbon Battery	61
E.1.2. Alkaline Battery	61
E.1.3. Li-Primary Battery	62
E.2. USB-Rechargeable 9 V Block	65
E.3. Summary 9 V Blocks	65
F. Material Cost	68

1. Purpose

A battery-pack containing two LiIon rechargeable batteries in series is used for supplying the fictive product (see *URS-BioPotHolter 1.0*). During incoming inspection and/or ahead of product assembly, the capacity of the batteries must be tested (in samples or all units) in a non-destructive way. This *Engineering Report* lists technical specifications of the developed test device *LiIonCheck9V125mA v1.a*, describes its hard- and firmware in detail, and provides verification & validation (V&V) for its correct function (pilot series of test devices). It further provides a risk management for the test procedure. The standard DIN ISO 13485 in its current version was **considered** for development.

2. User-Requirement

This section summarizes the user-requirements and defines the criteria for verification & validation (V&V) of each user-requirement.

URS-001 The test device *LiIonCheck9V125mA* controls the test of an individual battery-pack by one single, non-destructive discharge cycle and records the relevant data.

Comment A: The test intends a single discharge cycle, performed prior to use of the battery-pack in a product. Thus, the battery must maintain its performance for use in a product (recharging required).

Comment B: For obtaining a practical test, discharge shall be performed quicker as compared to the product (accelerated test), but noticeable damage to the battery pack must be avoided.

Check: Test results must be available for prototypes. They must proof proper performance and data for individual discharge cycles must be available.

URS-003 For obtaining a practical and straight forward test, a continuous discharge shall be performed.

Comment A: Due to accelerated testing (**URS-001**) the test load is higher as in the product.

Comment B: Discontinuous test cycles as defined in the standard *ANSI C18.1M Part 1* do not reflect a realistic load for the use of the battery-pack in the product.

Check: The design must allow for control of the discharge current. Test results must be available for prototypes.

URS-005 The test device *LiIonCheck9V125mA* must contain a USB interface for starting the test from an external computer and for transmitting the recorded data to a laboratory computer during the test.

Comment A: The USB interface provides also the power supply for the test device.

Check: The design must allow for control via a USB-serial port. Test results must be available for prototypes.

URS-007 The test device *LiIonCheck9V125mA* must record data with sufficient accuracy.

Comment A: Accuracy limits should allow for practical and repeatable testing.

Check: In test settings a comparison with accurate, calibrated equipment should be performed.

URS-009 Each test device *LiIonCheck9V125mA* must meet accuracy requirements without any need for individual device adjustment.

Comment A: Calibration measures like hardware trimming or firmware correction factors impose potential error sources and require time consuming maintenance by qualified staff.

Comment B: Surveillance of device accuracy must be performed in regular intervals.

Check: The design of the test device must involve a concept for obtaining the specified accuracy.

URS-011 The production and operation of the test device must meet economic requirements.

Comment A: As an indication, material cost should be below 200 Euro per device.

Check: The cost and the time budget for building prototypes should be reviewed.

3. Architecture

Figure 1 provides an overview of the test device architecture. An *Arduino UNO (r3)* μ -controller board is at the center of the design. It allows for serial communication with a laboratory computer via a USB-port. The in-house developed firmware running on the *ATmega328P* μ -controller of the *Arduino* board is described in detail in Appendix C. The *Arduino* board is equipped with an in-house developed shield-board (*Shield-LiIonCheck9V125mA v1.a*), which is described in detail in Appendix B. The shield is connected with the battery-pack under test and (via an I²C-serial-bus) with an LCD.

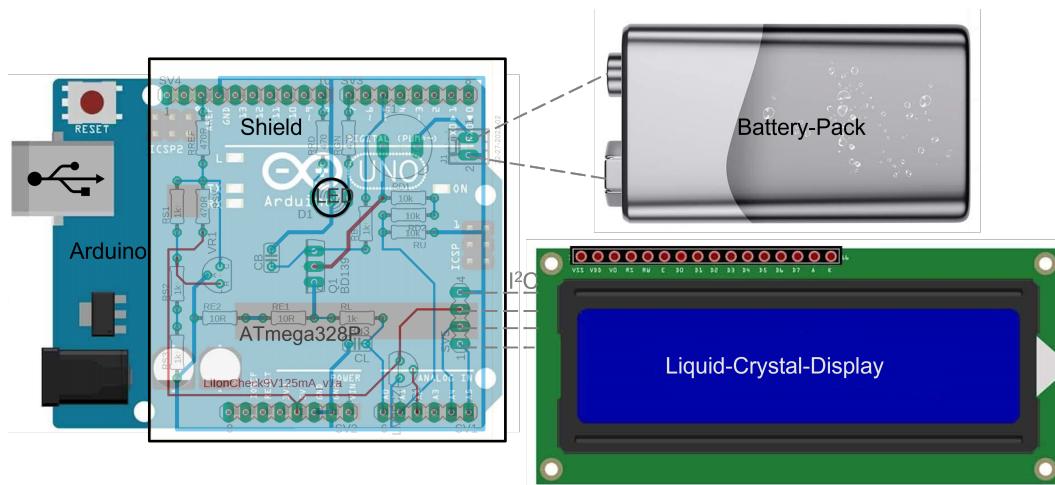


Figure 1: Schematic architecture of the *LiIonCheck9V125mA* test device (see text).

4. Specifications

4.1. Manufacturer Specifications

Table 1 lists the manufacturer specifications of the battery-pack containing two LiCoO_2 polymer cells with polymer anodes. Two cells are connected in series (“9 V LiIon-battery-pack”). The specifications were taken from the test report *TCT, Lithium Battery UN38.3*, provided by the manufacturer of the battery. The *TCT, Material Safety Data Sheet* describes chemical properties of the LiCoO_2 battery-pack.

Table 1: Manufacturer Specifications – Double Cell 650 mA h LiCoO_2 Polymer Battery-Pack.

ID	Parameter	Symbol	Value
MS-001	nominal voltage	V_{BN}	7.4 V
MS-003	nominal capacity	Q_{BN}	650 mA h
MS-005	cut off voltage – charge	V_{Bmax}	8.4 V
MS-007	cut off voltage – discharge	V_{Bmin}	5.5 V
MS-011	nominal current (charge & discharge)	I_{BN}	130 mA
MS-013	maximal current (charge & discharge)	I_{Bmax}	325 mA

A literature search was performed for linking manufacturer specifications to typical properties of LiCoO_2 batteries. Figure 2 displays voltage over time for a typical discharge cycle of a single LiIon-battery and re-labels the *y*-axis for a double-cell battery-pack. Voltage levels are in good quantitative agreement with manufacturer specifications. For MS-003 in Table 1 a discharge rate of 1C corresponds to a current of 650 mA.

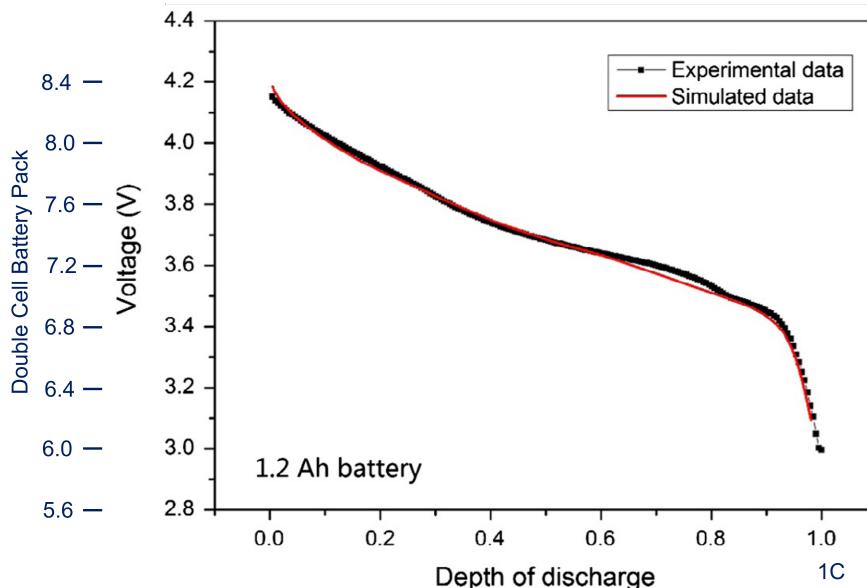


Figure 2: Voltage over time for a typical discharge cycle of a single LiCoO_2 -cell (data from Zhao et al. [1]). The *y*-axis was re-labeled to the voltage level of a double-cell battery-pack for allowing direct comparison with data in Table 1. For the depicted data discharge was performed at 1C, i.e., the battery was discharged within one hour.

4.2. Test Specifications & Risk Mitigation

The following test specifications were chosen for ensuring a nondestructive test. Thus, they contribute to risk mitigation:

RiMi-001 No deep discharge (see section A, Risk-201). The test capacity Q_{Test} was set to 500 mA h.

Comment A: This exceeds the minimal capacity requirement specified for the fictive product (480 mA h, see FS-BioPotHolter-007) by 4% and foresees also a buffer to the manufacturers nominal capacity Q_{BN} (see MS-003).

Comment B: According to [2, 3] some thousand discharge-charge-cycles can be obtained, when limiting depth of discharge to 80% of the nominal capacity. The chosen test capacity corresponds to 77% of the manufacturer specification (MS-003). Thus, when performing a very few discharge cycles testing is non-destructive.

RiMi-003 No deep discharge (see section A, Risk-201). The minimum cycle starting voltage V_{Smin} was set to 8.0 V.

Comment A: The test must **not** be performed for an insufficiently charged battery. The chosen limit is somewhat below V_{Bmax} (MS-005)

Comment B: Self discharge [4] may reduce available capacity prior to test and may require per-charging before test.

RiMi-004 No deep discharge (see section A, Risk-201). The minimum cycle end voltage V_{Emin} was set to 5.8 V.

Comment A: The chosen limit is above (MS-007).

RiMi-005 Moderate discharge current (see section A, Risk-203). The test current I_{Test} was set to 125 mA.

Comment A: The chosen test current is slightly below the nominal current according to manufacturer specification (MS-011).

Comment B: Test capacity Q_{Test} and test current I_{Test} define a test duration T_{Test} of 4 h. Thus, the test is significantly accelerated as compared to the minimal operation time of the fictive product (48 h according to URS-BioPotHolter-017).

Comment C: Despite it is common practice to operate LiIon-batteries at 1C charge or discharge rate, a capacity reduction of up to 10% is obtained at 1C [5]. For the chosen test current the battery is operated below $C = 0.2$ making capacity reduction negligible.

Accurate data on life cycles of LiCoO₂ cells confirms save operation within a larger temperature range of 10 °C to 30 °C (see [2], Figure 7). Table 2 summarizes all test specifications.

4.3. USB Supply

According to URS-005 the USB interface provides also the power supply for the test device. There is some variation in industrial USB-standards. In order to ensure proper function of the test device in a wide range of supply voltage, we specified the supply range twice as wide as in *Battery Charging Specification, Revision 1.2, 2010*. According to the used standard a

Table 2: Test Specifications for the Test Device *LiIonCheck9V125m*.

ID	Parameter	Symbol	Value	Origin
TS-021	nominal test current	I_{Test}	125 mA	RiMi-005
TS-023	nominal test capacity	Q_{Test}	500 mA h	RiMi-001
TS-025	test duration	T_{Test}	4 h	RiMi-005
TS-027	minimum starting voltage	V_{Smin}	8.0 V	RiMi-003
TS-029	minimum end voltage	V_{Emin}	5.8 V	RiMi-004
TS-031	ambient temperature	PZ-LiIonCheck9V125mV	10 °C to 30 °C	[2] (Fig. 7)
TS-033	minimal test discharge	I_{Tmin}	100 mA	RiMi-033
TS-035	maximal test discharge	I_{Tmax}	150 mA	RiMi-031
TS-037	maximal stand-by discharge	I_{0max}	1 mA	RiMi-035
TS-039	maximal test voltage	V_{Tmax}	10 V	RiMi-037

Experimental V&V of test specifications TS-021 to TS-029 is mandatory.

Test specifications TS-033 to TS-039 are limits checked by the firmware (see section [C.3]).

(non configured) USB interface can always deliver a supply current of at least 100 mA. This, is sufficient for the *LiIonCheck9V125m* test device which consumes a few tens of mA. Table [4.3] lists specifications for USB-supply.

Table 3: Functional Specifications for USB-Supply (test device *LiIonCheck9V125m*).

ID	Parameter	Symbol	Wert
FS-031	upper limit supply voltage	V_{5Vmax}	5.50 V
FS-033	nominal supply voltage	V_{5V}	5.00 V
FS-035	lower limit supply voltage	V_{5Vmin}	4.50 V
FS-037	min. USB supply current	I_{USB}	100 mA

Shield design (see section [B]) must ensure proper function within the specified supply limits.

5. Design

5.1. Test Phases

Figure [3] provides a flow chart of the four test phases. The firmware (described in section [C]) runs the test device across the phases. A duo-LED is used for indicating the phases:

Waiting-Phase – Green-Blinking: When starting the firmware, the test device checks whether a battery has been placed. When detecting battery voltage, the program waits until ENTER was pressed for starting the test.

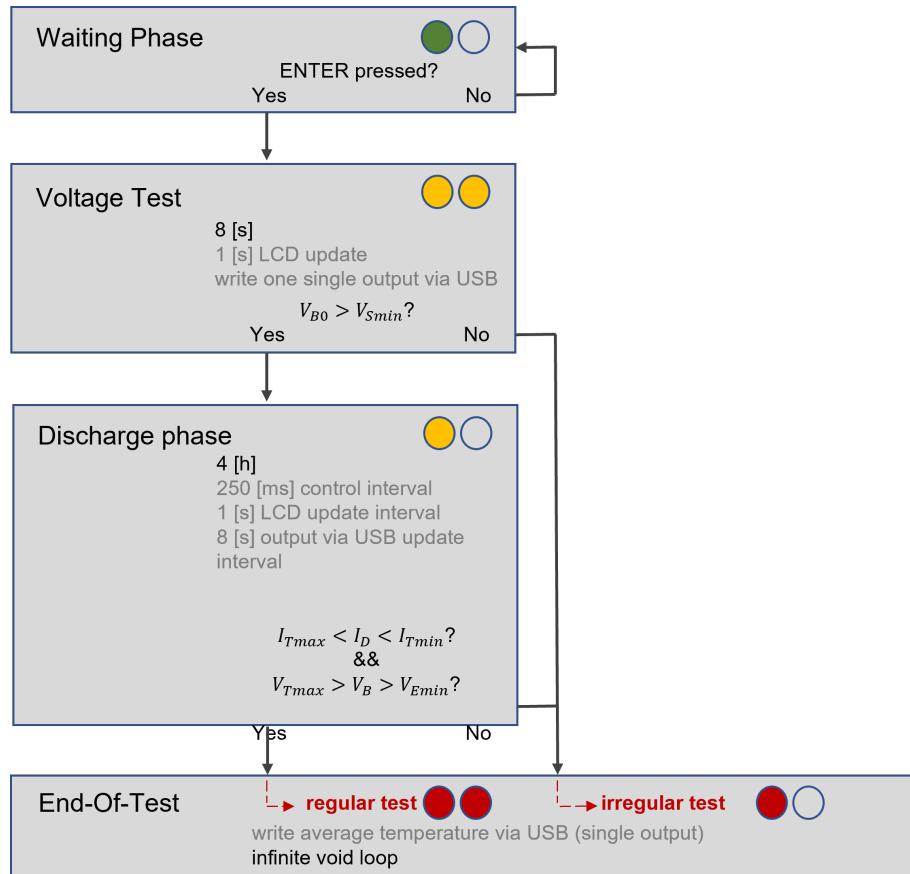


Figure 3: Flow-chart for test phases.

Voltage-Test – Continuous Yellow: The test device records the initial battery voltage V_{B0} at negligible load for 8 s. If voltage was below the minimum starting voltage V_{Smin} (TS-027) or if stand-by discharge was above the allowed limit I_{0max} , the test is aborted.

Discharge-Phase – Yellow-Blinking: The test device discharges the battery-pack under test at constant test current I_{Test} (TS-021) for a maximal test duration of 4 h (TS-025). If voltage or discharge current were out of predefined limits (see Table 2) the test is aborted.

End-of-Test – Red: Continuous – valid test; Blinking – aborted test.

A liquid crystal display (LCD) shows the run time of test phases and the battery voltage and discharge current reflecting the progress of the test.

5.2. Hardware

Figure 4 shows a picture of a test device prototype.

The shield was developed and built in house. It is described in detail in section B. Briefly, a BD139 power transistor is used for adjusting the discharge current to its nominal value I_{Test} (see TS-021) via control voltage U_{D5} . Emitter and collector voltage are measured via pins A0

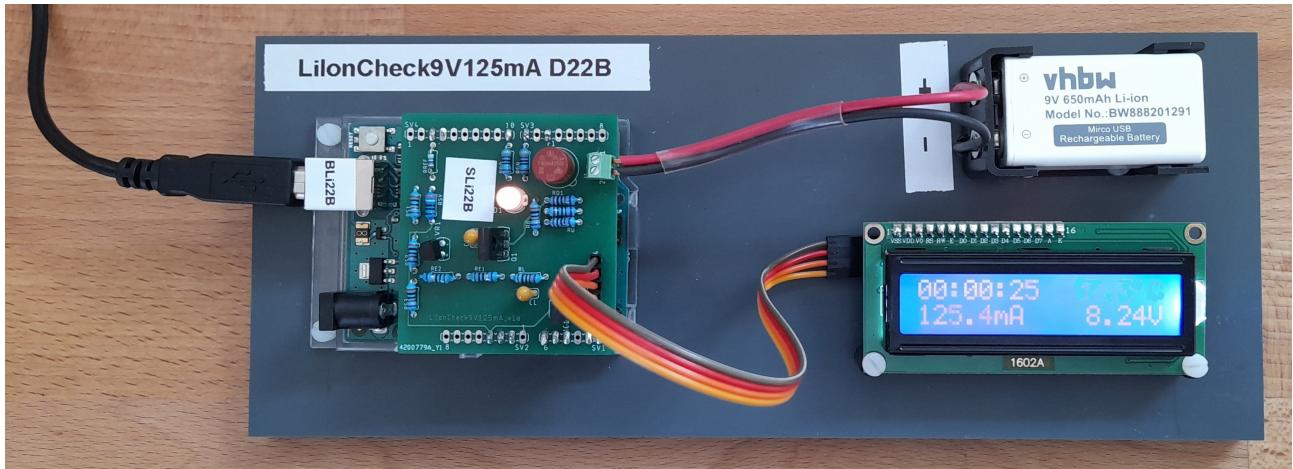


Figure 4: Prototype of the *LiIonCheck9V125mA* test device. On the left hand side the USB-cable is connected to the Arduino board which hosts the shield on its top. The shield is connected with the LCD (bottom right) and the battery (top right).

& A1 and discharge current I_D and battery voltage U_B are computed from this readings as described in detail in C.1.1. Based on the computed discharge current, the firmware PI-controller (see section C) re-adjusts the control voltage U_{D5} for keeping the current at its nominal value. Furthermore, the shield hosts an LM35 temperature sensor, for recording ambient temperature via pin A2.

For driving the duo-LED (see section 5.1) pins D7 and D8 are used for the green and red color respectively. Here, a positive logic is implemented. Thus, each color is turned on by switching its digital output to high. Therefore, a common cathode must be used for the duo-LED.

For driving the LCD via an I²C serial bus, a pin-header is foreseen. It hosts the pins for the 5 V supply, ground (GND), serial clock (SCL) and serial data (SDA). The pins A4 & A5 of the *Arduino UNO* board are internally connected with the I²C bus. Thus, they are connected to the pin-header.

An overview on pin connections is provided in Table 4.

An LCD1602-display (16 characters × 2 lines) with an integrated I²C-module was used. Several suppliers provide *Arduino* compatible LCD1602-displays. The actually used display was chosen from the ERM1602-6 Series (blue background light, HD44780 compatible controller, see datasheet). The supplier *GeeekPi* delivers this unit already pre-equipped with an IIC/I²C Serial Interface Adapter Module (*GeeekPi* Stock Keeping Unit [SKU]-code 19332, manufacturer Kuongshun). The I²C-module is pre-configured to the I²C-address 0X27, which is currently used by the firmware (see section C.3). As an option for future design modifications, the module allows for re-configuring of the address in the range 0X20 to 0X27 by three soldering patches. A poti on the module allows for adjustment of LCD-contrast. Figure 5 depicts the LCD1602-display together with the geometrical position of the I²C-pins and the cursor positions. Table 5 lists the bill of materials. Appendix F lists prizing.

Table 4: Pins used by Shield-Board *Shield-LiIonCheck9V125mA v1.a.*

Pin	Function Detail	Comment
A0	near-DC emitter voltage (analog input ADC)	measure discharge current
A1	collector voltage (analog input ADC)	measure battery voltage
A2	LM 35 output (analog input ADC)	measure temperature
A4	I ² C Serial Data (SDA)	serial bus to LCD
A5	I ² C Serial Clock (SCL)	serial bus to LCD
D5	set control voltage (ADC; 980 Hz PWM)	control discharge current
D7	drive duo-LED (digital output)	green channel
D8	drive duo-LED (digital output)	red channel
AREF	external reference voltage	reference for ADC

The shield is connected to the 5 V pin and to all ground pins.

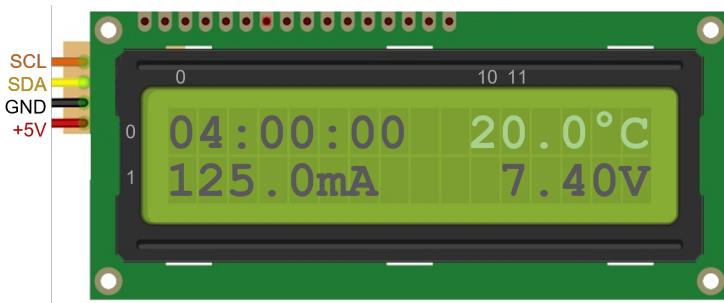


Figure 5: *Top left corner:* Geometrical position of the four I²C-pins. *Display:* recording time is displayed in the top-line in hh:mm:ss format (hours:minutes:seconds) starting at cursor position 0, 0. Discharge current I_D is displayed in the bottom-line starting at cursor position 1, 0. Battery voltage V_B is displayed in the bottom-line starting at cursor position 1, 11. Cursor position cursor position 0, 10 in the top-line was reserved for displaying ambient temperature Θ_A in future versions (if needed).

For teaching purposes, a universal 9 V battery holder was used. This allows also testing standard primary 9 V batteries for comparison.

Table 5: Bill of Materials – Test Device *LiIonCheck9V125mA v1.a.*

Item #	Part	Name	Comment
1	main board	<i>Arduino UNO r3</i>	see datasheet
2	shield board	<i>Shield-LiIonCheck9V125mA v1.a</i>	see section B
3	liquid crystal display	ERM1602-6 Series with I ² C-module	see text
4	battery holder	<i>KEYSTONE</i> part-nr. 1294	see datasheet
5	battery cable	red & black, 0.7 mm ² 120±20 mm	mounted with a shrink tube
6	ribbon cable	four poles for LCD	supplied together with LCD
7	spacers 10 mm	3 pieces	lab equipment
8	Nylon screws	M3 × 16 mm, 8 pieces	lab equipment
9	mounting plate	250 mm × 100 mm × 10 mm PVC	see Figure 6

The device needs a USB-cable with a “B-type” connector.

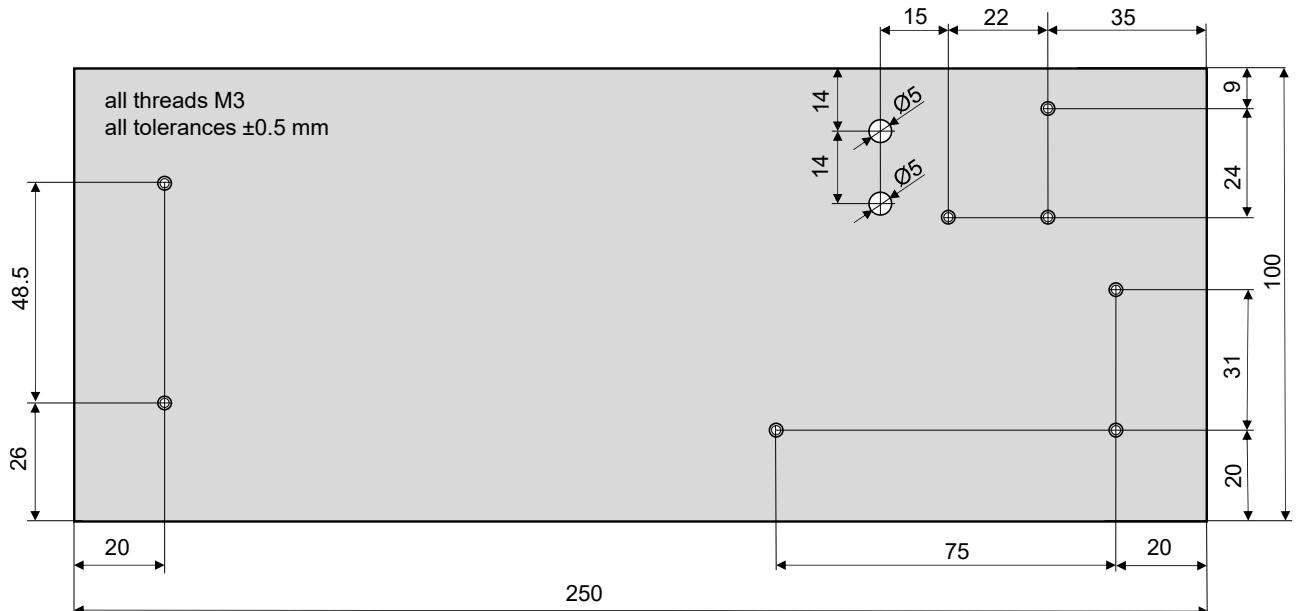


Figure 6: Drill and thread positions for the mounting plate (dimensions in mm).

5.3. Accuracy

Design features impacting accuracy are described in detail in section [B.2]. Briefly, the design was chosen such, that discharge current and battery voltage are assessed with an accuracy of $\pm 1.5\%$ within the relevant range. Assessment of capacity requires an additional measurement of time (with an additional tolerance of $\pm 0.25\%$). Thus, a total accuracy of $\pm 1.75\%$ was defined.

Future versions may include automatic recording of ambient temperature. Since the allowed temperature range is broad and accurate verification of ambient temperature recordings is extremely time and cost intensive a wide tolerance may be foreseen.

Table 6: Specified Accuracy.

Parameter	Accuracy	Specified Range
Discharge current I_D	$\pm 1.5\%$	I_{Tmin} to I_{Tmax} (see Table [2])
Battery Voltage V_B	$\pm 1.5\%$	V_{Bmin} to V_{Bmax} (see Table [2])
Test Capacity Q_{Test}	$\pm 1.75\%$	see text

5.4. Test Recording

A test recording was performed on March 18, 2022 using the 650 mA h LiIon battery pack labeled as *Accu A*. The *Arduino UNO r3* board labeled as *A_GF01* together with shield *A* was used. Prior to the test, the battery was charged with a current limit of 160 mA and voltage limit of 8.35 V until the charging current dropped below 50 mA. After a break of 5 min the test was started (protocol comparable to [2]). An overview on the hardware used for the tests is provided in Table [7].

Figure [7] depicts the results. The reading obtained from the voltage test was close to the voltage limit during charging. During discharge current was at 125 mA with a maximal deviation of 0.1 mA. The recorded voltage trace is in qualitative agreement with the typical discharge curve of a LiCoO₂-battery pack as depicted in Figure [2]. As intended, discharge was stopped after the predetermined test duration T_{Test} (see TS-025) for avoiding damage or wear of the battery due to deep discharge (see [RiMi-005]). The voltage at the last discharge sample was 6.99 V.

Thus, the tested battery meets the capacity requirement of 500 mA h. Beyond the scope of the test we can coarsely estimate the full capacity of the battery pack as follows. As can be seen from Figures [2] and [28] the voltage at the end of the discharge cycle ($\approx 7\text{ V}$) corresponds to $\approx 80\%$ of the total discharge cycle. Thus, we obtain an estimate of $500\text{ mA h}/0.8 = 625\text{ mA h}$. This estimate is in good agreement with the specified capacity of the battery.

Concluding, the measurement demonstrates that the developed test device accurately controls the discharge cycle, avoids for deep discharge of the battery pack under test and provides a reliable test of the capacity requirement.

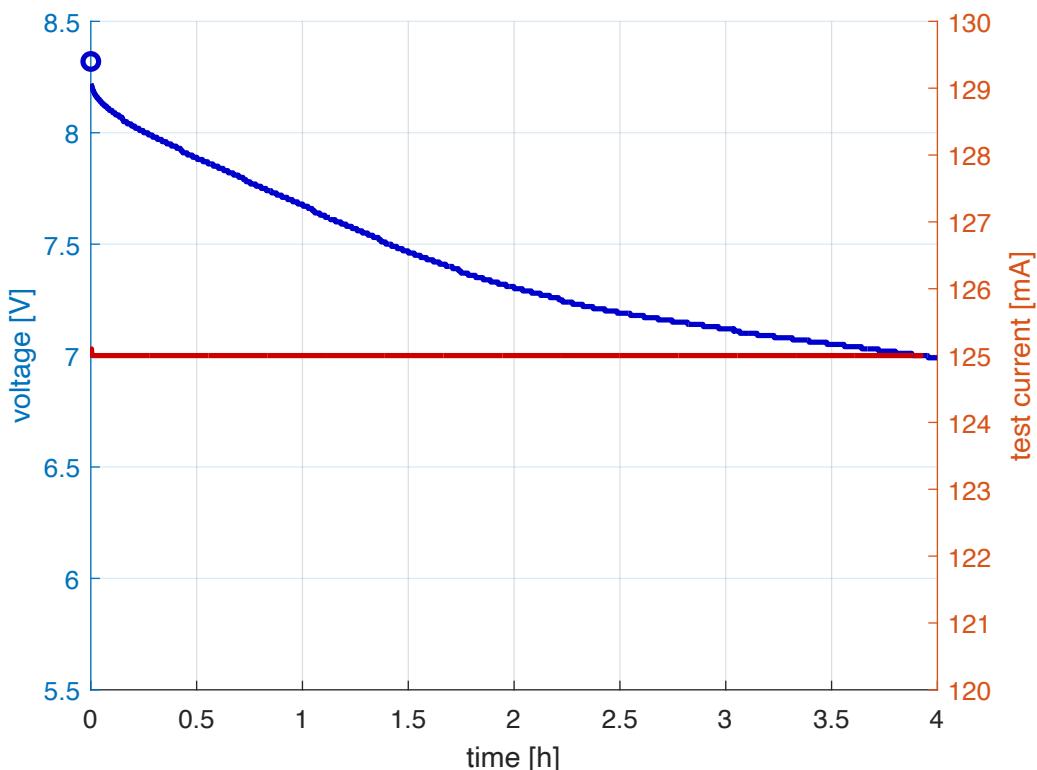


Figure 7: Voltage (blue) and current (red) for a test recording on the 650 mA h LiIon battery pack *Accu A*. The circular marker displays the result of the voltage test (8.32 V).

6. Verification & Validation

6.1. V& V of Test Specifications and Accuracy

Test specifications were checked as follows:

TS-021 The device discharges the battery pack under test constantly with the specified nominal test current I_{Test} .

Section C.4 provides a successful verification for a simulated test discharge.

Section E provides a successful validation for miscellaneous battery types.

TS-023 Discharge is terminated when the nominal test capacity Q_{Test} was drained from the battery under test.

Section E provides a successful validation for a LiIon battery (specified according to MS-003).

Test Procedure PZ-LiIonCheck9V125mV was developed for training purposes and three out of three batteries *Accu A* to *C* were tested successfully. This is documented by the test protocols PP-LiIonCheck9V125mV20220227A & C. For *Accu B* no protocol is available, since data was deleted erroneously. Concluding the data provides a successful validation.

TS-025 Discharge is terminated after the nominal test duration Q_{Test} .

Table 7: Used Test Hardware.

Test Device	Components	First Calibrated On
D22A	board BLi22A & shield A	April 22, 2022 (see KP1-LiIon9V125mA20220222A)
D22B	board BLi22B & shield B	April 22, 2022 (see KP1-LiIon9V125mA20220222B)
Shield	Abbreviation	First Tested On
SLi22A	shield A	March 14, 2022 (see section B.5.1)
SLi22B	shield B	March 14, 2022 (see section B.5.1)
Battery Pack	Abbreviation	First Tested On
Accu A	–	March 18, 2022 (see text)

In the development phase both shields were first mounted onto a development board labeled by A_GF01 for comparing shield on the same board.

Section C.4 provides a successful verification of real time control with accurate timing.

Section E provides a successful validation for miscellaneous battery types.

TS-027 Discharge does not start below the minimum starting voltage V_{Smin} .

Section C.4 provides a successful verification of this limit.

TS-029 Discharge terminates below the minimum end voltage V_{Emin} .

Section C.4 provides a successful verification of this limit.

Section E provides a successful validation for miscellaneous battery types.

TS-031 Ambient temperature Θ_A .

On April 3, 2022 the accuracy of the temperature measurement implemented in test device was assessed. Here the results are summarized briefly. During discharge the recorded temperature was approximately 8 °C above room temperature. As it was confirmed by a reference measurement, the offset was caused by a heating of the LM35 temperature sensor by the power dissipation of the test device. As a practical fix, test protocol PZ-LiIonCheck9V125mV requires monitoring of room temperature by an external thermometer.

In the current firmware implementation, automatic temperature recordings are not visible for the user, since they contain unacceptable errors.

TS-033 A minimal discharge current limit I_{Tmin} was implemented in the firmware.

Section C.3.4 describes the implemented routine in detail.

TS-035 A maximal discharge current limit I_{Tmax} was implemented in the firmware .

Section C.3.4 describes the implemented routine in detail.

TS-037 A maximal stand-by discharge limit I_{0max} was implemented in the firmware .

Section C.3.3 describes the implemented routine.

TS-039 A maximal test voltage limit V_{Tmax} was implemented in the firmware .

Section C.3.4 describes the implemented routine in detail.

Concluding, all test specifications were fulfilled.

Accuracy requirements were checked as follows:

The current reading I_D of the test device fulfills accuracy requirement (see Table 6).

Section C.4 provides a successful verification of the accuracy limit.

Calibration procedure PZ-LiIonCheck9V125mV was developed for training purposes and two out of two devices $D22A$ and $D22B$ were calibrated successfully. This is documented by the test protocols KP-LiIonCheck9V125mV20220227A & B.

The voltage reading V_B of the test device fulfills accuracy requirement (see Table 6).

Section C.4 provides a successful verification of the accuracy limit.

Calibration procedure PZ-LiIonCheck9V125mV was developed for training purposes and two out of two devices $D22A$ and $D22B$ were calibrated successfully. This is documented by the test protocols KP-LiIonCheck9V125mV20220227A & B.

Ambient temperature measurement is not implemented in the current version of test devices (see above).

The measured test capacity Q_{Test} fulfills accuracy requirement (see Table 6).

Section C.4 provides a successful verification accuracy in time and current measurement.

Calibration procedure KZ-LiIonCheck9V125mV was developed for training purposes and two out of two devices $D22A$ and $D22B$ were calibrated successfully. This is documented by the test protocols KP-LiIonCheck9V125mV20220227A & B.

Section E provides a successful validation for miscellaneous battery types.

Concluding, all relevant accuracy requirements were fulfilled.

6.2. Validation of User Requirement

URS-001 Individual, non-destructive test cycles were successfully performed and test data was stored.

Risk-management was applied for guiding test-specifications, hardware and firmware design for ensuring non-destructive testing (see section A).

Test results were obtained for sub-components (see section B.5.1), simulated discharge cycles (see section C.4) and miscellaneous battery types (see section E).

Test Procedure PZ-LiIonCheck9V125mV was used for training purposes and all batteries (*Accu A* to *C*) were tested successfully.

URS-003 The test cycle designed in section 5.1 provides a practical, continuous discharge test setting requiring user interaction only at the beginning and the end of the recording.

Continuous discharge cycles were successfully tested for simulated discharge cycles (see section C.4) and miscellaneous battery types (see section E).

Test Procedure PZ-LiIonCheck9V125mV was used for training purposes and all batteries (*Accu A* to *C*) were tested successfully by continuous discharge cycles.

URS-005 All device tests were performed in a fully automatic fashion including automatic saving of recorded data.

Control and data storage via the USB-port was demonstrated for simulated discharge cycles (see section [C.4](#)), miscellaneous battery types (see section [E](#)) and a series of test recordings [Test Procedure PZ-LiIonCheck9V125mV](#).

URS-007 Dedicated tests for confirming accuracy by the using accurate reference data were successfully performed.

Accuracy was successfully confirmed for simulated discharge cycles (see section [C.4](#)) and calibration of test devices [Calibration Procedure KZ-LiIonCheck9V125mV](#).

Plausible capacity estimates were obtained for miscellaneous battery types (see section [E](#)).

URS-009 The developed design allows for meeting accuracy requirements without trimming.

The design is based on the selection of accurate components based on a calculation of error propagation (see section [B.2](#)) and accurate calculation of physical parameters from AD-conversion (see section [C.1.1](#)).

URS-011 The production and operation of the device meets economical requirements.

The material is within budget (see Appendix [F](#)).

Recording is fully automatic requiring user interaction only at the start and the end of each measurement (see section [5](#)).

Concluding, all user requirements were fulfilled.

7. Summary & Conclusions

This engineering report documents the development, design and testing of the test device *Li-IonCheck9V125mA* according to a V-model development cycle. In parallel, a risk management process was involved for avoiding damage or harm to operators or test batteries and for excluding erroneous test results. Based on this process a practicable nondestructive accelerated test was implemented. Data confirms that the device meets its specifications including accuracy requirements. Furthermore, a practical nondestructive test setting was implemented.

The developed prototypes provide a reliable basis for repeated testing of batteries prior to use in the fictive product. The current documentation contains all information for up-scaling testing – if required – by building additional test units in the future.

8. Documentation & Literature

8.1. Internal Documents

- URS-BioPotHolt 1.0. This is a fictive document for training purposes. Its relevant specs are listed on the slides shown in the lecture
- FS-BioPotHolt 1.0. This is a fictive document for training purposes. Its relevant specs are listed on the slides shown in the lecture
- KZ-LiIon9V125mA. This document describes the calibration procedure for the test devices. It was created by students during training.
 - [] KP-LiIon9V125mA20220422A is the first calibration protocol created for board D22A (created for training)..
 - [] KP-LiIon9V125mA20220422B is the first calibration protocol created for board D22B (created for training)..
- PZ-LiIonCheck9V125mV. This document describes the test procedure for the batteries. It was created by students during training.
 - [] PP-LiIon9V125mA20220427A is the first test protocol created for board *Accu A* (created for training)..
 - [] PP-LiIon9V125mA20220427C is the first test protocol created for board *Accu C* (created for training)..
 - [] *Accu B* was also tested successfully, but the data was deleted erroneously..

8.2. Standards and Guidelines

- American National Standards Institute, ANSI C18.1M Part 1, Portable Primary Cells and Batteries with Aqueous Electrolyte — General and Specifications.
- USB Battery Charging Specification, Revision 1.2, 2010.
- ISO 13485:2016 (E) Medical Devices – Quality Management Systems – Requirements for Regulatory Purposes.

8.3. Datasheets & Test Reports

TCT, Lithium Battery UN38.3 Test Report TCT, Material Safety Data Sheet

References

- [1] R. Zhao, J. Liu, J. Gu, Simulation and experimental study on lithium ion battery short circuit, Applied Energy 173 (2016) 29–39. doi:<https://doi.org/10.1016/j.apenergy.2016.04.016>.

- [2] W. Diao, S. Saxena, M. Pecht, Accelerated cycle life testing and capacity degradation modeling of li₂o₃-graphite cells, Journal of Power Sources 435 (2019) 226830. doi:<https://doi.org/10.1016/j.jpowsour.2019.226830>. URL <https://www.sciencedirect.com/science/article/pii/S0378775319308237>
- [3] M. Qadrda, NickJenkins, J. Wu, Chapter ii-3-d - Smart Grid and Energy Storage - Fundamentals and Applications, in: ? (Ed.), McEvoy's Handbook of Photovoltaics (Third Edition), Science Direct, 2018, pp. 915–928. doi:<https://doi.org/10.1016/B978-0-12-809921-6.00025-2>.
- [4] B. Y. Liaw, R. G. Jungst, G. Nagasubramanian, H. L. Case, D. H. Doughty, Modeling capacity fade in lithium-ion cells, Journal of Power Sources 140 (1) (2005) 157–161. doi:<https://doi.org/10.1016/j.jpowsour.2004.08.017>.
- [5] B. Scrosati, J. Garche, Lithium batteries: Status, prospects and future, Journal of Power Sources 195 (9) (2010) 2419–2430. doi:<https://doi.org/10.1016/j.jpowsour.2009.11.048>.
- [6] W. Kester, Taking the mystery out of the infamous formula, “snr = 6.02n + 1.76db,” and why you should care, Analog Devices, Tutorial MT-001 (10-A) (2008). URL <https://www.analog.com/media/en/training-seminars/tutorials/MT-001.pdf>

Appendices

A. Risk-Management

Risk-Management involves three parts. Operator-Protection (i.e., safety of persons and staff performing the tests), protection of the device under test for damage and the exclusion of incorrect results from the analysis.

A.1. Operator-Protection

During the test of a battery-pack the device under test is **NOT** used as a medical product or equipment thereof. However, the safety of operators (users) must be ensured. Below the identified risks are listed together with the applied measures for risk mitigation. Proper training of operators must be performed.

Risk-101 Hazard of electric shock.

Action A: The test device *LiIonCheck9V125mA v1.a* must be supplied exclusively via its USB-Interface (USB-B-connector). By operation in connection with a USB-compatible laboratory computer the device operates at a safe voltage (5 V) and is galvanically isolated from the power-grid.

- The test device is operated from a 5 V USB-supply (see [4.3](#) and [5.2](#)).

Action B: Operators must be instructed that the use of an alternative power-supply is not allowed.

The implemented test procedure (see [PZ-LiIonCheck9V125mV](#)) contains a proper warning.

Action C: The test device *LiIonCheck9V125mA v1.a* must be designed such, that can be operated without touching the boards or electronic components. *Operators must be instructed accordingly.*

The test device is controlled via a serial USB port from a remote computer (see [5.1](#) and [C.3.2](#)).

The implemented test procedure (see [PZ-LiIonCheck9V125mV](#)) contains a proper warning.

Risk-103 Hazard of fire.

Action A: Limitation of discharge current avoids burning of the battery pack.

- see implemented actions for over current (risk [Risk-203](#)).

A.2. Protection of the Battery-Pack Under Test

[URS-001](#) requires that testing is non-destructive and any damage to the battery-back under test must be avoided. This involves also a reduction of natural wear during discharge. The following hazards were identified and mitigated by the following actions.

Risk-201 Hazard of deep discharge.

Action A: The test capacity Q_{Test} discharged from the device during test is smaller than its nominal capacity. A reasonable buffer must be foreseen.

- Implementation see section [4.2], [RiMi-001].

Action B: The firmware terminates discharge when battery voltage is below a critical level.

- Implementation see [4.2], [RiMi-003] and [RiMi-004].

Action C: During any stand-by phase of the test the residual discharge current must be sufficiently small.

- Implementation see section [B.3], [RiMi-011] and [C.2], [RiMi-035].

Action D: Battery-pack shall be placed inside the test device only for testing purposes. Storage of batteries inside the test device is **not** allowed.

The implemented test procedure (see [PZ-LiIonCheck9V125mV](#))- instructs operators accordingly.

Risk-203 Hazard of short circuit or over current.

Action A: The nominal test current must be within the specified current range of the battery-pack.

- Implementation see section [4.2], [RiMi-005].

Action B: The circuits must be designed such that they provide sufficient current limitation.

- Implementation see section [B.3], [RiMi-015].

Action C: The firmware terminates discharge when a critical current level is exceeded.

- Implementation see [C.2], [RiMi-031].

Action D: As a complimentary measure to action B, a fuse should be foreseen.

- Implementation see section [B.3], [RiMi-017].

Risk-205 Hazard of unintended charging.

Action A: The shield-board was designed such that unintended charging is not possible.

- Implementation see section [B.3], [RiMi-019].

Risk-207 Hazard of wear.

Action A: A single test discharge must have essentially negligible battery life-cycle.

Clarification: For the regular life-cycle of a battery-pack in the fictive product *BioPotHolter 1.0* wear occurs during regular use. Thus, wear must be analyzed in context with regular use.

[RiMi-001] (comment B, section [4.2]) provides a literature based rational for negligible wear for the chosen test specifications.

A.3. Test Reliability

There are two types of hazards originating from the process of testing. On the first hand goods of substandard quality (i.e., battery-failure) must not pass the test, since they may cause a malfunction or insufficient function of the product. On the other hand – for avoiding economical loss – samples of good quality must not fail the test.

Risk-301 Insufficient capacity of battery-pack under test.

Action A: The accuracy of capacity assessment must allow for reliable exclusion of devices of insufficient capacity.

The implemented test device allows for measurement at sufficient accuracy (see section [5.3], Table [6]).

Risk-303 Erroneous test due to faded battery capacity.

Action A: At the beginning of each test battery, voltage is checked for confirming sufficient pre-charging.

- A proper cycle starting voltage level was specified (see [4.2], [RiMi-003]).
- Firmware terminates the test if voltage was too low (see [C.3.3]).

Risk-305 Unpredictable events (e.g., failure of device components) may occur.

Action A: The following parameters are monitored by the firmware.

- Test current upper and lower limit (see [RiMi-031] and [RiMi-033]).
- Stand-By discharge (see [RiMi-035]).
- Maximal test voltage (see [RiMi-037]).

Action B: The firmware enters a save mode if a check fails.

- The save mode is an infinite loop. It sets hardware permanently to stand-by (no discharge, see [RiMi-041]).

B. Shield Design

This section describes the design of the shield board *Shield-LiIonCheck9V125mA v1.a* in three steps. First, circuit design and calculation are described. Next, the shield board PCB layout created using the *Autodesk Eagle v.9.6.2* layout editor is presented. Finally, test results for two shield prototypes are listed.

B.1. Circuit Design & Calculation

In this section capital letters are used for indicating time independent values (e.g., DC voltage U). Lower case letters are used for time dependent parameters (e.g., transient voltage $u(t)$). The calculated design parameters were transferred to the layout and are listed in the bill of materials (Table 8).

The shield board *Shield-LiIonCheck9V125mA v1.a* is mounted on an *Arduino UNO (r3)* board (see section 3). The shield enables measurement and control of battery discharge and displays the actual operational state. Thus, it hosts the following functional units:

- A controllable current sink allowing for accurate selection of the discharge current.
- An accurate reference voltage for A/D conversion.
- A temperature sensor for assessing ambient temperature.
- A red/green duo-LED and an I^2C -interface for driving the LCD.

B.1.1. Design Current Sink

Figure 8 illustrates the concept of the adjustable current sink. We first compute parameters by analytical formulas and then verify the results by an *LTspice* simulation.

Operating point: The operating point was chosen such, that at the nominal discharge current reasonable voltage levels are obtained. Furthermore, overheating of all parts was prevented. Near the operating point variations in component parameters (e.g., tolerances or temperature drift) are compensated by firmware control. According to specification, at the operating point the nominal discharge current $I_{Test} = 125 \text{ mA}$ (FS-021) is drained from the battery. We can assume at first approximation, that this current essentially drains to the emitter I_E . The resistor in the emitter circuit was chosen by $2 \times 10 \Omega$. Thus, the emitter voltage U_E equals approximately 2.5 V. It is, thus, fairly centered within the 5 V USB-supply (see FS-033). The power dissipation P_{RE} in the emitter circuits resistors is obtained by:

$$P_{RE} = U_E I_E = 2.5 \text{ V} \times 125 \text{ mA} \simeq 310 \text{ mW}. \quad (1)$$

Two $1/4$ Watt 10Ω resistors in series were used for safely dissipating this power. The larger part of the total power is dissipated by the transistor Q_N . We use the upper bound of the battery voltage $V_{Bmax} = 8.4 \text{ V}$ (MS-005) for computing the maximal power P_Q dissipated by the transistor

$$P_Q = (V_{Bmax} - U_E) I_E = (8.4 \text{ V} - 2.5 \text{ V}) \times 125 \text{ mA} \simeq 738 \text{ mW}. \quad (2)$$

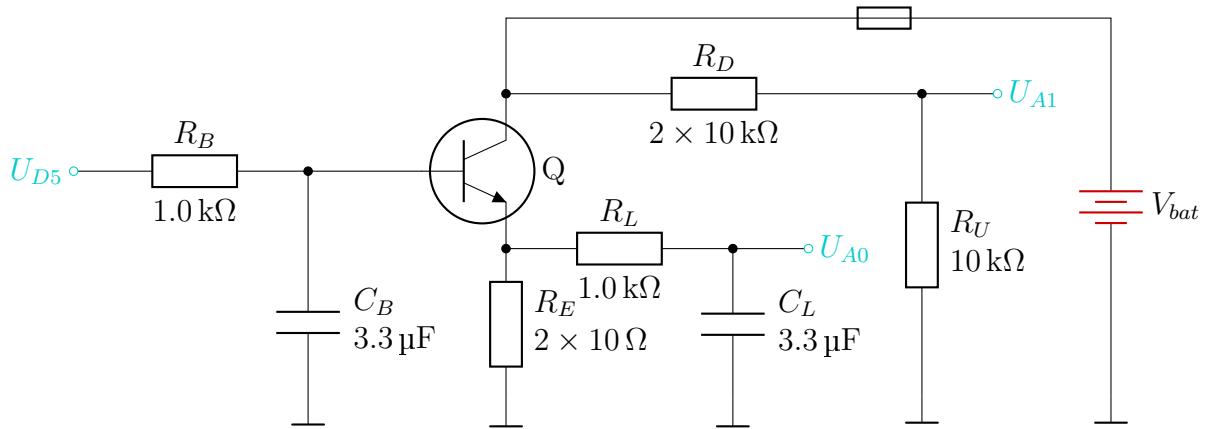


Figure 8: Basic circuit of the adjustable current sink. Via the PWM pin D5 of the *Arduino UNO* board the control voltage V_{D5} is applied to the basis resistor R_B of transistor Q. The capacitor C_B smoothies the PWM-signal. The current gain of the transistor allows for approximately linear control of the discharge current in the collector circuit via the control voltage V_{D5} . Emitter current is measured via the voltage drop in the emitter resistor. A second low-pass R_L, C_L further reduces the ripple. Its output is connected to analog-pin A0. The voltage of the the battery is assessed via the voltage divider R_D, R_U and Pin A1.

An npn-transistor of type BD139 was chosen for the circuit. Due to its TO-225 package (variant of TO-126) it can handle a power dissipation of up to 1.25 W without the need of an additional heat sink (see datasheet). The base-emitter voltage drop U_{BE} was assumed to be 0.6 V (typical value for a silicon bipolar junction transistor). Furthermore, we assumed the current gain $\beta = 124$ (estimated from Figure 1 in the datasheet). Choosing 1.0 kΩ for the base resistor R_B , we obtain for the input voltage U_{D5} at the operating point

$$U_{D5} = I_E \left(\frac{R_B}{\beta + 1} + R_E \right) + U_{BE} = 125 \text{ mA} \times \left(\frac{1.0 \text{ k}\Omega}{124 + 1} + 20 \Omega \right) + 0.6 \text{ V} \simeq 4.1 \text{ V}. \quad (3)$$

Thus, the input voltage at the operation point amounts to approximately 80 % of the nominal 5 V USB supply voltage. For the given 8-bit resolution of the Arduino AD-conversion this relates to an integer value of $0.8 \times (2^8 - 1)$, which is near 200. The step-size of the discrete control allows for adjusting input voltage with a relative step size of 0.5 % which is sufficiently accurate. Furthermore, the operating point is about 20 % below the nominal supply voltage which ensures a sufficiently large control range.

Remaining Ripple: Arduino allows for adjusting input voltage by pulse width modulation (PWM). Thus, soothing by low-pass filters is required. By selecting pin D5 of the *Arduino UNO* for control, PMD is accomplished at a frequency of $f_{D5} = 980 \text{ Hz}$ (i.e., period $T_{D5} = 1.02 \text{ ms}$). This is twice the frequency compared to most of the pins which eases removal of ripple. At the operation point, the PWM-signal gets low for 18 % of each cycle, which corresponds to an off-interval of 184 μs. During this off-phase the capacity C_B at the transistor base discharges via the resistors $R_B // (\beta + 1)R_E$. We obtain for the time constant τ_B

$$\tau_B = [R_B // (\beta + 1)R_E]C_B = (1.0 \text{ k}\Omega // 125 \times 2 \times 10 \Omega) \times 3.3 \mu\text{F} \simeq 2.4 \text{ ms}. \quad (4)$$

Assuming an approximately linear discharge during the low phase, we obtain an estimate for the percentage of variation of the base voltage

$$\frac{\Delta U_B}{U_B}[\%] = \pm \frac{1}{2} \frac{T_{off}}{\tau_b} \times 100\% = \pm \frac{184 \mu s}{2 \times 2400 \mu s} \times 100\% = \pm 3.83\%. \quad (5)$$

Therefore, base voltage ripple amounts to $\pm 3.83\%$ of its mean. This ripple is transmitted in a near 1:1 fashion to the emitter. It provides also an estimate for the residual ripple in the discharge current. This ripple is of relatively high frequency and sufficiently small for having negligible effect on battery capacity assessment. However, a further low-pass is used for accurate assessment of the mean emitter voltage. Reasonably, the parameters of the second low-pass were chosen identical as for the basis circuit low-pass. We obtain for the corner frequency f_L

$$f_L = \frac{1}{2\pi R_L C_L} = \frac{1}{2\pi \times 1.0 \text{ k}\Omega \times 3.3 \mu\text{F}} = 48.2 \text{ Hz}. \quad (6)$$

The ratio of corner frequency f_L and fundamental frequency f_{D5} provides an estimate for the ripple in the measured voltage

$$\frac{\Delta U_{D5}}{U_{D5}}[\%] \simeq \pm \frac{f_L}{f_{D5}} \frac{\Delta U_B}{U_B}[\%] = \pm \frac{48.2 \text{ Hz}}{980 \text{ Hz}} \times 3.83\% = \pm 0.19\%. \quad (7)$$

The true ripple is even smaller as estimated by eq. (7), since the low pass damps harmonics n by a factor n ($n = 2, 3, \dots$, neglected for simplifying computation). The residual ripple in the measured mean is acceptable for accurate assessment of the mean emitter voltage. Below, an *LTspice*-simulation was performed providing a verification of the estimated ripple.

Delay: The two low-pass filters cause a delay in the transmission of the control voltage U_{D5} to the emitter output U_{A0} . The largest change in U_{D5} occurs in the power-on phase when it is switched from zero to the operating point. Since we aim for accurate measurements with deviations well below $\pm 1\%$ a sufficiently long delay must be considered for obtaining a sufficiently stable reading. Here, the eight-fold of the time constant τ_B provides 20 ms as a reasonable estimate. Note, that in a test setting of 4 h duration, this delay does not impose any restriction. Section C.1 describes the implementation of this delay in the firmware.

LTspice-Simulation: *LTspice* (Version XVII(x64) (17-0.27.0) Analog Devices, Inc.) was used for providing a verification of the computations performed above. The *LTspice*-circuit model was created according to Figure 8 and saved to file LiIonCheck9V125mA_v1a.asc (stored as a read-only). A *spice*-model for transistor BN139 was obtained from github.com. The 4.1 V operation voltage at the control pin (eq. (3)) was modeled by a PWM-signal as described above. Battery voltage was set to its nominal value $V_{BN} = 7.4 \text{ V}$ (MS-011). Figure 9 displays three voltages for the power-on phase. As can be seen, the simulated ripples are slightly smaller as estimated above. Furthermore, also the delay in the output voltages agrees with the estimations above. The simulated steady state mean discharge current of the battery was 125.5 mA with a residual ripple of $\pm 5.0 \text{ mA}$. Thus, the *LTspice*-simulation confirms the results of the analytic computation.

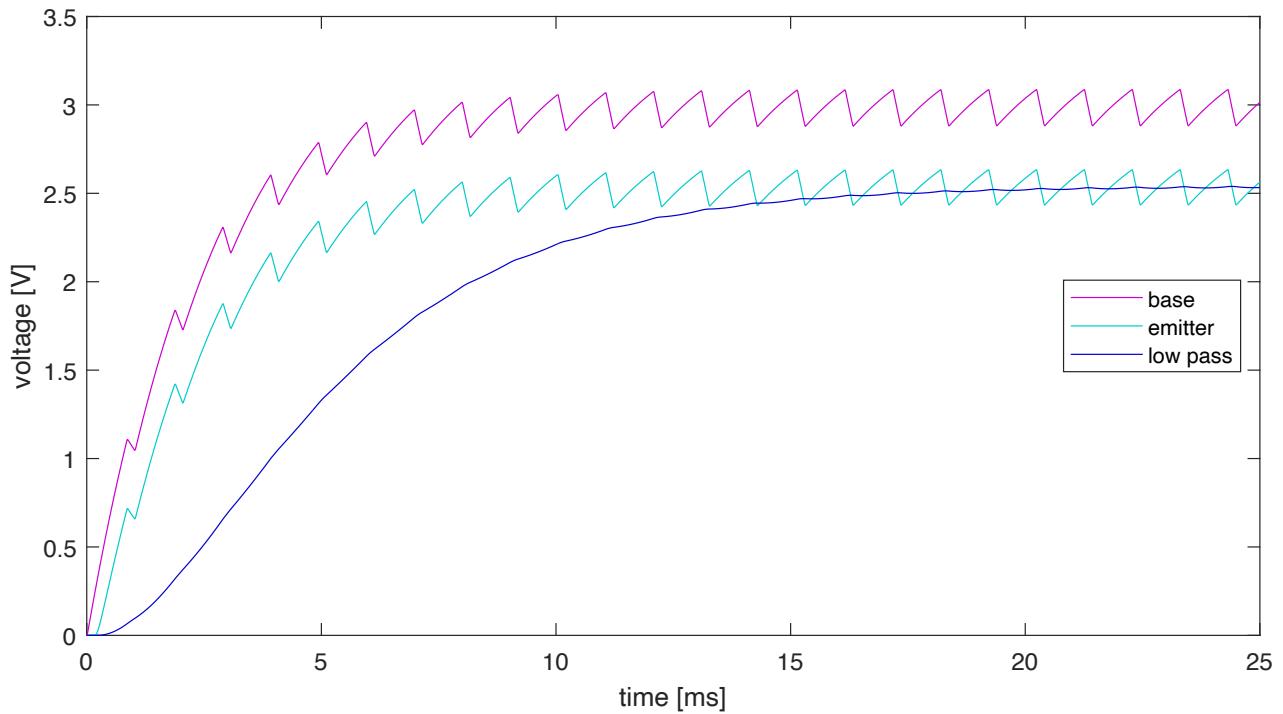


Figure 9: *LTspice*-simulation of the voltages at the base, the emitter and the low-pass output of the controlled current sink for the power-on phase (PWM-modulated control pin). At the end of the simulated interval, signals settled near the periodic activity. The ripple in the base voltage was $\pm 3.43\%$. At the output of the second low-pass the residual ripple was reduced to $\pm 0.14\%$.

B.1.2. Design Reference Voltage

When aiming to obtain high accuracy of AD-conversion for *Arduino* boards, the use of an external reference voltage is recommended. The TL431 shut regulators are a family of integrated circuits provided by various manufacturers, which (by using a voltage divider) allow for reference adjustment in a wide range. Internally, TL431 circuits use a fixed voltage V_{ref} of 2.5 V. By using parts of “grade B” (TL431B) a narrow tolerance of $\pm 0.5\%$ is obtained for the internal voltage. Figure 10 A) depicts the chosen design. The voltage divider resistors were chosen as $R_{S1} = 1 \text{ k}\Omega$ and $R_{S2} = 2 \times 1 \text{ k}\Omega$ yielding the reference voltage V_{Cref}

$$V_{Cref} = V_{ref} \left(1 + \frac{R_{S1}}{R_{S2}} \right) = 2.5 \text{ V} \times \left(1 + \frac{1}{2} \right) = 3.75 \text{ V}. \quad (8)$$

The chosen reference voltage equals to 75 % of the nominal supply voltage V_{5V} . This is sufficiently large for obtaining good accuracy in the voltage range of interest. The values were chosen such, that the voltage divider can be built using three resistors of equal type. Since R_{S1} was chosen at a relatively low value, voltage drop generated by the gate current I_{ref} (typ. a few μA according to the datasheet) is in the order of a few mV and appears negligible.

The current flow across the voltage divider amounts to $3.75 \text{ V} / (3 \times 1 \text{ k}\Omega) = 1.25 \text{ mA}$. The serial resistor R_{SV} was chosen by 330Ω . Thus, at the lower limit of the USB supply voltage ($V_{5Vmin} = 4.5 \text{ V}$, see FS-035) the current flow across the serial resistor becomes $(4.50 - 3.75) / 330 \Omega = 2.27 \text{ mA}$.

This provides sufficient current for operating the shut and the divider over the entire specified range of supply. When using an external reference voltage, the AREF-pin must be configured by firmware for proper use (see section C.3). The resistor R_{SR} provides a basic current limitation in the case of incorrect configuration and protects the internal reference of the *Arduino* board. It was set to a relatively low value of 470Ω for ensuring that the voltage drop across R_{SR} becomes small (the AD-converter input resistance is approximately $30\text{k}\Omega$ according to the documentation of the *Arduino UNO r3* board). We obtain for the final reference voltage

$$V_{Aref} = V_{Cref} \frac{30\text{k}\Omega}{30\text{k}\Omega + R_{SR}} = 3.75\text{V} \times \frac{30\text{k}\Omega}{30\text{k}\Omega + 470\Omega} = 3.69\text{V}. \quad (9)$$

Section B.2.1 provides an error estimation for this voltage reference.

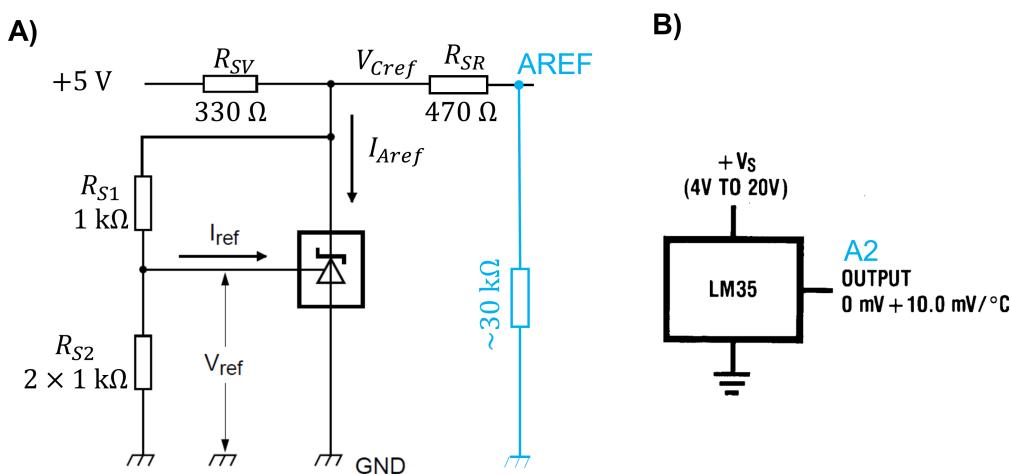


Figure 10: *Panel A)* Adjustment of the reference voltage V_{Aref} using the voltage divider R_{S1} , R_{S2} . The serial resistor R_{SV} limits the current flow to the shunt regulator. The output resistor R_{SR} provides a protection for the circuit behind the AREF-pin of the *Arduino UNO* board (see text). The diagram was based on a drawing from the NXP Semiconductors datasheet and adopted for illustrating the actual design. *Panel B)* Measurement of ambient temperature by operating the LM35 from the 5 V supply. By directly connecting its output to the A2-pin (see Table 4) temperature can be measured between 0 °C to 150 °C (see datasheet).

B.1.3. Room Temperature Sensor

Room temperature is assessed by the integrated circuit *LM35* (see Figure 10 B). The drawing was copied from the datasheet of the *LM35*. The circuit converts positive temperature in °C into an analog output with the slope $10\text{mV}^{\circ}\text{C}^{-1}$. Thus, the ambient temperature 20 °C corresponds to 0.20 V. This value is in the lower range of AD-conversion. However, no further amplification was foreseen (not necessary for meeting accuracy requirements, see TS-031).

B.1.4. Duo-LED

A 5 mm red/green duo-LED was chosen for indicating the phases in the test cycle as described in section 5.1. As outlined in section 5.2 a component with a common cathode must be used.

Red/green duo-LEDs are available from multiple suppliers with comparable data. A SSL-LX5099IGW (Lumex Inc.) duo-LED was used for the prototype and computation is based on its specifications. The device emits at a peak wavelength of 635 nm (red) and 565 nm (green) requiring a forward voltage of $V_{RD} = 2.0\text{ V}$ and $V_{GN} = 2.2\text{ V}$, for red and green respectively. By using a current limitation resistor $R_{RD} = 470\Omega$ for the red color, the current I_{RD} at the nominal supply level (see Table 4.3) is then obtained from

$$I_{RD} = \frac{V_{5V} - V_{RD}}{R_{RD}} = \frac{5.0\text{ V} - 2.0\text{ V}}{470\Omega} = 6.4\text{ mA}. \quad (10)$$

This value is well below the maximal forward current (25 mA) of each color and sufficiently large for obtaining a clearly visible optical output. The same resistor value was also chosen for the green channel, since it operates at a comparable forward voltage.

B.2. Tolerances

For meeting the accuracy requirements defined in Table 6, proper tolerances must be foreseen. In this section the overall uncertainty for each of the three measurement channels (discharge current, battery voltage and ambient temperature) is estimated such, that the specified tolerances are met. At the end of Appendix B, Table 8 summarizes the accuracy requirements of all components.

B.2.1. Uncertainty in Reference Voltage

The accuracy of the reference voltage can be estimated from the sum of uncertainty of individual components in the circuit (Figure 10 B).

Uncertainty Shunt Regulator: The chosen TL431BCLPG regulator provides a maximal uncertainty of $\pm 0.5\%$.

Uncertainty Adjustment Resistors: According to eq. (9) the reference voltage is adjusted by resistors R_{S1} and R_{S2} . For obtaining sufficient accuracy a tolerance of $\pm 0.1\%$ was chosen for these resistors. For computing propagation of uncertainty we assume that R_{S1} was at the upper/lower bound of of tolerance (i.e., ± 0.001) and R_{S2} was at the lower/upper bound (i.e., ∓ 0.001). With this assumptions we obtain for the adjustment term of eq. (9)

$$\left(1 + \frac{1}{2} \frac{1 \pm 0.001}{1 \mp 0.001}\right) = 1.500 \pm 0.001. \quad (11)$$

Adjustment resistors provide a relative uncertainty of $\pm 0.001/1.5 = \pm 0.07\%$. Thus, the total uncertainty in the reference voltage was estimated by $\pm 0.57\%$.

B.2.2. Uncertainty in AD-conversion

The 10 bit AD-converter of the ATmega328P μ -controller provides an uncertainty of $\pm 2\text{-LSB}$ (datasheet). Thus, we obtain the absolute uncertainty of AD-conversion ΔV_{AD}

$$\Delta V_{AD} = \pm V_{Aref} \frac{2}{2^{10} - 1} = \pm 3.75\text{ V} \times \frac{2}{1023} = \pm 7.33\text{ mV}. \quad (12)$$

B.2.3. Uncertainty in Battery Data

Assessment of discharge current and battery voltage is based on recording of analog input voltage on pins A0 and A1 (see C.1.1).

Uncertainty in Pin-Voltage: First, the uncertainty of AD-conversion enters calculation. Since, the current sink circuit in Figure 8 was designed such that voltages at pins A0 & A1 are typically near 2.5 V, the relative error of AD-conversion can be estimated by $\pm 7.33\text{ mV}/2.5\text{ V} = \pm 0.29\%$. Furthermore, also the uncertainty of the reference voltage (see above) must be considered. Summing up both uncertainties the total uncertainty in voltage is estimated by $\pm 0.86\%$.

Uncertainty in Discharge Current: In addition to pin-voltage U_{A0} the emitter resistors (see C.1.1) enter the calculation of the discharge current. For obtaining sufficient accuracy a tolerance of $\pm 0.1\%$ was chosen for these resistors. Summing up both uncertainties, the total uncertainty in discharge current was estimated by $\pm 0.96\%$. This value is below the specified accuracy limit in Table 6.

Uncertainty in Battery Voltage: In addition to pin-voltage the uncertainty in the transfer function of the voltage divider R_D, R_U enters the calculation. For obtaining sufficient accuracy, a tolerance of $\pm 0.1\%$ was chosen for these resistors. This yields also an estimate for the uncertainty in the transfer function. Summing up both uncertainties the total uncertainty in battery voltage was estimated by $\pm 0.96\%$. This value is below the specified accuracy limit in Table 6.

B.2.4. Uncertainty in Temperature

According to the data sheet the output voltage of the LM35 provides an uncertainty of $\pm 0.5\text{ }^\circ\text{C}$ at room temperature. Furthermore, at its nominal voltage-to-temperature gain of $10\text{ mV }^\circ\text{C}^{-1}$ the uncertainty in AD-conversion introduces an additional uncertainty of $\pm 7.33\text{ mV}/10\text{ mV }^\circ\text{C}^{-1} = \pm 0.7\text{ }^\circ\text{C}$. Thus, the total uncertainty in temperature is estimated by $\pm 1.2\text{ }^\circ\text{C}$. This value is below the specified accuracy limit in Table 6.

B.3. Risk Mitigation

The following actions for risk mitigation were foreseen in the circuits described above .

RiMi-011 High impedance measurement of battery voltage. The voltage divider R_D, R_U contains sufficiently large resistor values ($2 \times 10\text{ k}\Omega$ und $10\text{ k}\Omega$). Thus, in the stand-by phases of the test, deep discharge is avoided (Risk-201). For the chosen resistors, a residual discharge current of approximately 0.3 mA is obtained at the upper limit of the battery voltage $V_{Bmax} = 8.4\text{ V}$ (see Table 1). Thus, over 12 h of stand-by a maximum of 3.6 mA h (i.e., 0.7% of the test capacity Q_{Test}) gets drained. This value is acceptable.

RiMi-015 Physical limitation of the current. The resistor R_E in combination with the transistor Q and its base resistor R_B limit the discharge current (Risk-203). In a worst-case fault condition (failure of control), control voltage U_{D5} may be assumed at the upper limit of the specified USB supply voltage 5.5 V . Performing identical calculation as applied for the operating point voltage we obtain the failure current by $(5.5\text{ V} - U_{BE})/(R_E + R_B/\beta)$.

For the parameters listed above, the maximal failure current is estimated by 175 mA. This value is acceptable.

RiMi-017 Fuse. Complimentary to [RiMi-015](#) a 160 mA fuse is foreseen on the board. It prevents for short-circuit discharge in the unlikely situation of a failure in physical current limitation (e.g., short circuit in resistor or transistor). This risk mitigation measure protects the battery in case of a damaged board.

RiMi-019 Unintended charging of the battery. The maximal specified USB-supply voltage ($U_{5Vmax} = 5.5$ V) is below the lower limit of the specified battery test voltage ($U_{Emin} = 5.8$ V). This renders unintended charging impossible ([Risk-205](#)). Furthermore, even in the chase of extremely unlikely failure conditions the transistor Q and the resistor R_D limit current flow to the battery under test.

B.4. Eagle Circuit & Board Layout

The shield was created using *Autodesk EAGLE Version 9.6.2*. Figure [11](#) shows the circuit diagram. Figure [12](#) depicts the shield layout. A pin-header (SV5) was foreseen for driving the I²C LCD. The four pins were connected such, that a ribbon cable can be used for connection with the I²C module shown in Figure [5](#). The bill of materials is listed in Table [8](#).

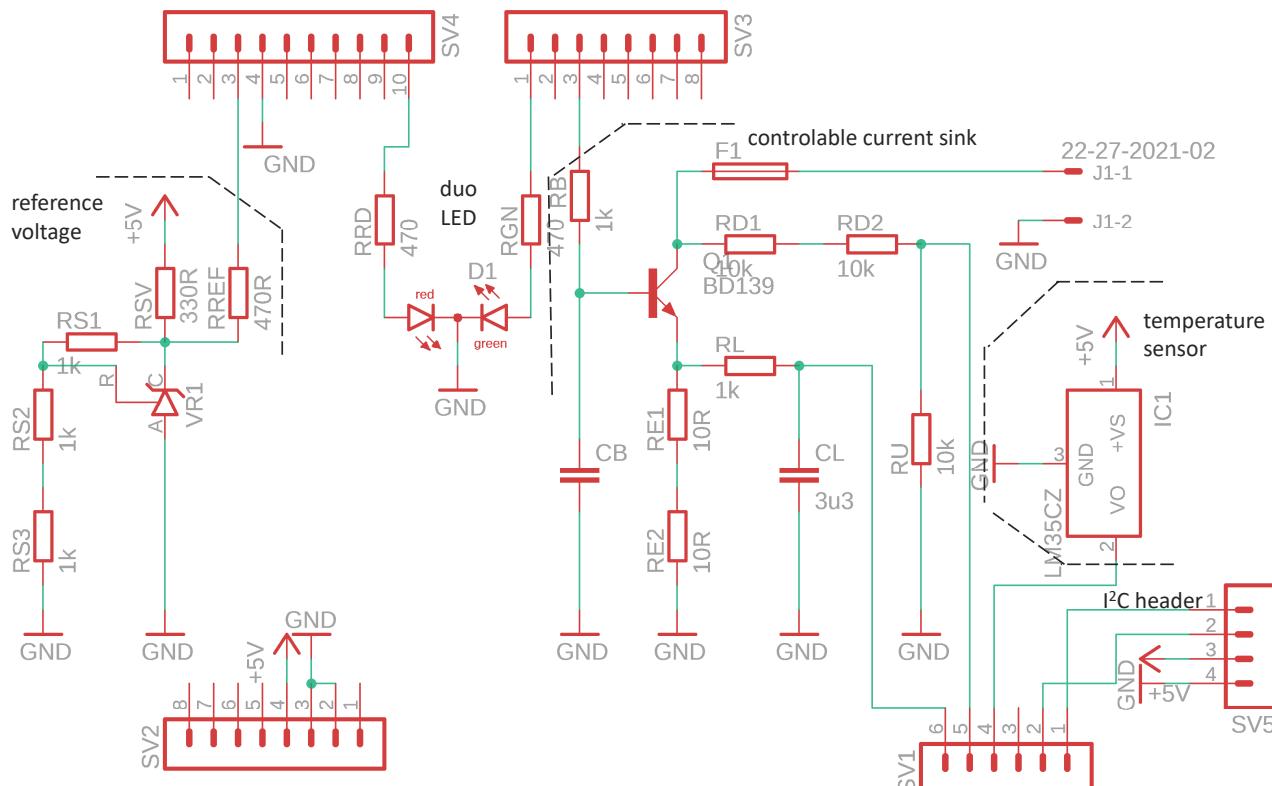


Figure 11: *EAGLE* circuit diagram of the shield. The sub-circuits as described in sections [B.1.1](#) to [B.1.4](#) are indicated

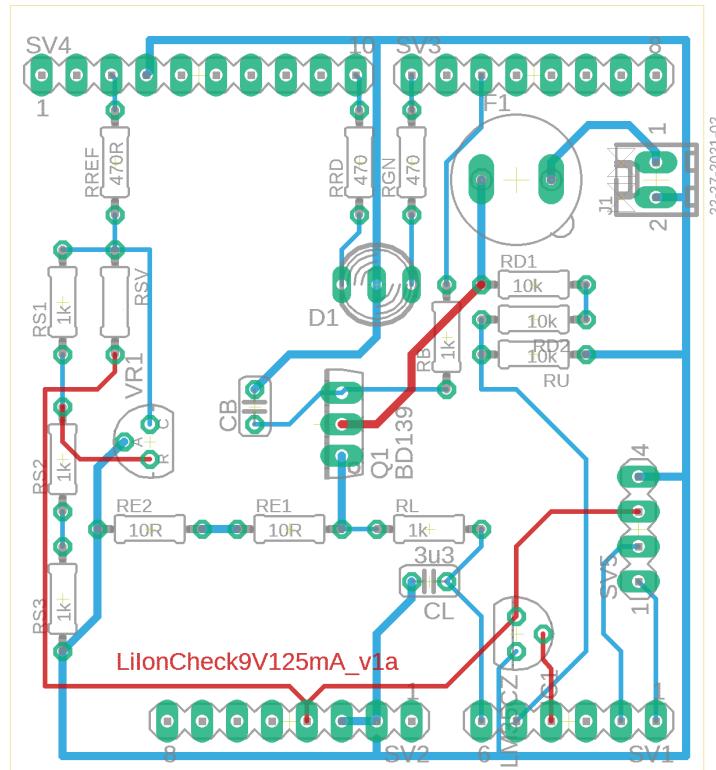


Figure 12: Layout of the shield. Pin headers SV1 to SV4 were placed such that the shield can be connected with the *Arduino UNO r3* board.

B.5. Shield Prototypes

Between March 10 and 14, 2022 Two shield prototypes were built according to the design described in section [B.4]. A bread board was used for testing the following components prior to use: item 7 - npn-transistor, item 8 - duo LED and item 9 - voltage reference. In total five transistors were tested. They displayed a relatively large variation in current gain ranging from ≈ 80 to ≈ 110 (transistors at room temperature, parameter variation of BD139). For shield A a transistor with a high current gain was chosen, while for shield B a transistor with a low gain was chosen for investigating the influence of this parameter variation on test devices. Due to a delay in the delivery of item 10 - temperature sensors were added later and tested on April 4, 2022. Figure [13] depicts the two shield prototypes.

B.5.1. Test Results

The following tests were performed on March 14 and 15, 2022 for testing the shields **prior** to connection with the *Arduino UNO r3* board.

Current Sink Operating Point: The concept of this test was to supply PIN D5 of the shield with the nominal operation point voltage U_{B5} (see section [B.1]) and to measure the discharge current I_D . Thus, the laboratory power supply [LaPS-001] was used for providing the current at a voltage level near the nominal battery voltage V_{BN} (see [MS-001]). Multimeter [MuMe-003] was connected in series to the power supply and used as an ampere-meter and multimeter [MuMe-001] was connect parallel to the terminal block of the shield and used as a volt-meter

Table 8: Bill of Materials – Shield.

Item #	Part	Value/Name	Key Feature(s)	Quantity
1	resistor	10 Ω	$\pm 0.1\%$, 250 mW	2
2	resistor	470 Ω	–	3
3	resistor	330 Ω	–	1
4	resistor	1 k Ω	$\pm 0.1\%$	5
5	resistor	10 k Ω	$\pm 0.1\%$	3
6	capacitor	3.3 μF	tantal electrolyte	2
7	npn-transistor	BD139	circuit design (see section B.1.1)	1
8	duo LED	red, green	common cathode (see section 5.2)	1
9	voltage reference	TL431BCLPG	$\pm 0.57\%$ (see section B.2.1)	1
10	temperature sensor	LM35	$\pm 1^\circ\text{C}$	1
11	fuse	TR5 housing	160 mA (RiMi-017)	1
12	terminal block	–	2.54 mm, 2 pins	1
13	SIL socket	–	2.54 mm, up to 10 pins	4
14	SIL pin header	–	2.54 mm, 4 pins	1

(see Figure 14). Here, the voltage drop over the ampere meter, does **not** enter the analysis. The operating point was simulated by a PWM-signal as described in Figure 22.

Due to the power dissipation in the transistor, the current gain displays some variation with the warming of the transistor. Thus, a stop-clock-app on a smartphone was used for measuring discharge parameters over time. Due to the fast heating of the transistor in the first few seconds after power-on accurate an initial reading was documented.

As can be seen from Table 9, Shield A displays a larger discharge current as compared to shield B. This is due to the selection of the individual transistors described in the first paragraph of section B.5. Thus, the experimental setting allows for investigating individual variation of transistor current gain. The deviation of the discharge current from its nominal value $I_{Test} = 125 \text{ mA}$ is always below $\pm 15\%$. Thus, in the assembled test device PI-control can remove the individual offset of each shield.

The oscilloscope Osci-001 was used for measuring the residual ripple at the base of the BD-139 transistor. The peak-to-peak amplitude was approximately 250 mV at a mean base voltage of 3.3 V. This corresponds to a relative base-ripple of $\pm 3.8\%$. This result is in good quantitative agreement with the analytical estimate in eq. (5) and the estimate of the LTspice-simulation (Figure 9). The residual ripple at the output of the emitter-circuit low-pass was too small for accurate measurement (few mV peak-to-peak, covered by noise).

Reference Voltage: The the laboratory power supply LaPS-001 was used for supplying the shield via the 5 V pin and the neighbouring ground pins (clamp type probes). Another clamp type probe was connected to the V_{CREF} pin of resistor R_{REF} . Supply voltage was varied over the specified USB-voltage range (see Table 4.3). The results are listed in Table 10. At the lowest USB-voltage, shield B displayed a deviation of -1.5% for the nominal reference voltage (3.75 V, see section B.1.2). All other measurements were approximately -0.4% below the nom-

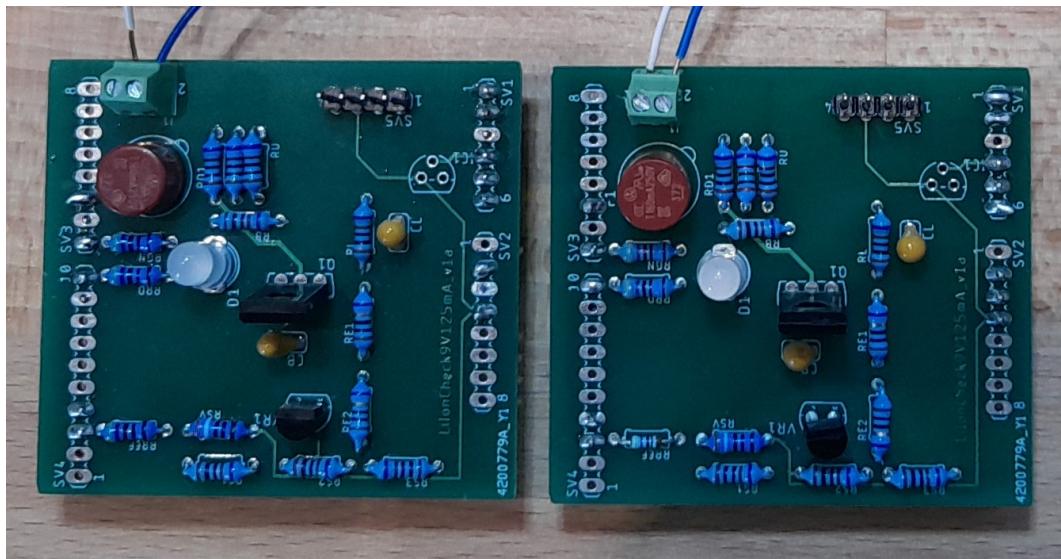


Figure 13: Shield prototypes A (left) and B (right). The LM35 was not available at the time of first assembly (see text).



Figure 14: Test of the current sink (Shield A) at the operating point (see text). The shield was supplied via the terminal block and a clamp type probe was used for connecting pin D5.

inal value and, thus, well within the expected range of uncertainty (see [B.2.1](#)). The specified accuracy of [MuMe-003](#) (FLUKE 179) is $\pm 0.16\%$.

The observed deviation at the lower USB-supply level was due to a slightly too high value used for R_{SV} in the shield prototypes (390Ω). In the actual design it was reduced to $R_{SV} = 330\Omega$ (see [B.1.2](#)). The deviation of shield B affects only the lower boarder of the USB-supply range (chosen with a wide buffer in Table [4.3](#)). Furthermore, also this boarder deviation is only approx. -1% below the other measurements. This deviation, therefore, does **not** impose a noticeable restriction to the use of shield B.

B.5.2. Interpretation of Shield Test

The measured discharge parameters (voltage and current) are in good qualitative and quantitative agreement with the computed design parameters. Thus, the shield-prototypes are

Table 9: Discharge Current at Operation Point.

time	current I_D mA	voltage U_B V	current I_D mA	voltage U_B V
	Shield A		Shield B	
<5 s	≈ 122	7.64	≈ 106	7.67
1 min	130.8	7.63	114.8	7.66
2 min	132.9	7.62	116.9	7.65
5 min	134.5	7.62	118.5	7.65
10 min	134.8	7.62	118.6	7.65
15 min	134.9	7.60	118.7	7.65
fuse R_F	0.82 Ω		0.87 Ω	
multimeters:	I_D ... MuMe-003	U_B ... MuMe-001	R_F ... MuMe-002	

Table 10: Reference Voltage V_{CREF} .

supply voltage U_{USB} V	V_{CREF} Shield A V	V_{CREF} Shield B V
5.52	3.739	3.735
5.00	3.738	3.734
4.60	3.737	3.733
4.50	3.737	3.692
multimeters:	U_{CREF} ... MuMe-003	U_{USB} ... MuMe-001

suitable for assessing battery capacity when used in the test device *LiIonCheck9V125mA v1.a*.

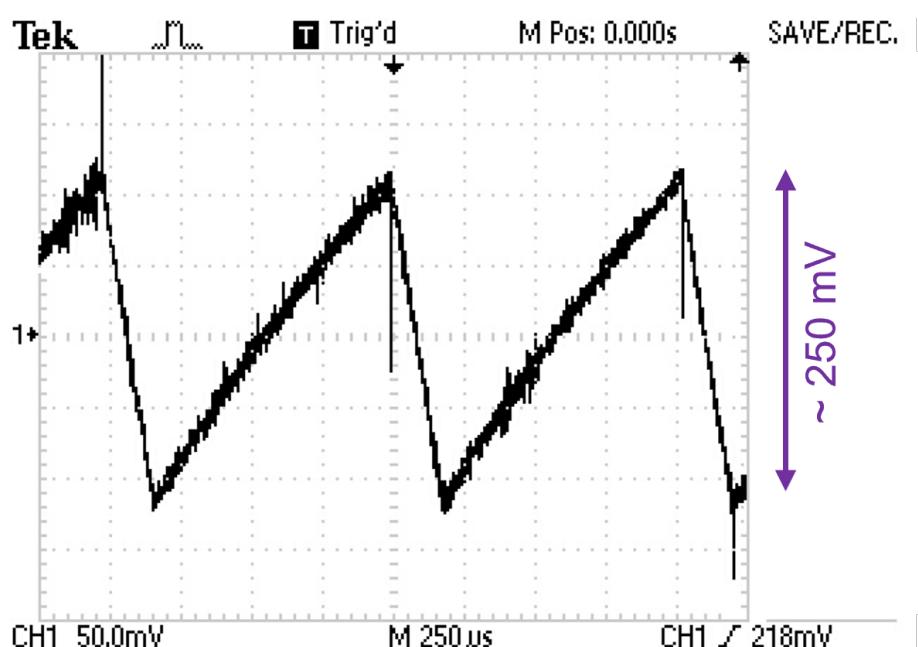


Figure 15: Residual ripple (AC-component) measured at the base of the BD139 transistor (see text).

C. Firmware Design

C.1. Controller Design

In this section capital letters are used for indicating time independent values (e.g., DC voltage U). Lower case letters are used for time dependent parameters (e.g., transient voltage $u(t)$).

C.1.1. Analog Input – Physical Parameters

The AD-converter on the *ATmega328P* μ -controller (*Arduino UNO* board) hosts a 10-bit AD-converter. Thus, each single reading is an integer number N_x in the range from zero to $2^{10} - 1 = 1023$. The firmware (see C.3) operates the AD-converter with the external reference voltage $V_{Aref} = 3.69\text{ V}$ (see B.1.2). The voltage U_x of an individual reading is then defined by

$$U_x = \frac{N_x}{1023} \times V_{Aref}. \quad (13)$$

The step size of AD-conversion is $3.69\text{ V}/1023 = 3.61\text{ mV}$. For all three analog input channels (see Table 4) the respective voltages U_{A0} to U_{A2} are obtained from eq. (13).

Computations of the discharge current and the battery voltage are based on the circuit of the current sink shown in Figure 8.

Discharge current The (near-DC) emitter voltage is obtained from U_{A0} . The emitter current is then U_{A0}/R_E . A correction by the current gain factor $(\beta-1)/\beta$ is needed for obtaining the collector current reassembling the first discharge pathway. The second (high impedance) discharge pathway is the voltage divider $2R_U, R_U$. The current across the divider circuit is obtained by U_{A1}/R_U . Thus, the total discharge current I_D is obtained from

$$I_D = \frac{\beta-1}{\beta} \frac{U_{A0}}{R_E} + \frac{U_{A1}}{R_U}. \quad (14)$$

Battery Voltage The collector voltage at the input of the voltage divider equals $3 \times U_{A1}$. In series to the collector the resistance of the fuse R_F generates a small voltage drop $I_D R_F$. Thus, the battery voltage V_B is obtained from

$$V_B = 3 \times U_{A1} + I_D R_F. \quad (15)$$

For the fuses used in the shield prototypes we estimated $R_F = 0.85\Omega$ from measurements (see Table 9).

Temperature According to the specification of the LM35 temperature sensor (see Figure 10 B) ambient temperature Θ_A can be computed from U_{A2} (see Table 4) as follows

$$\Theta_A = \frac{U_{A2}}{0.01\text{ V }^\circ\text{C}^{-1}}. \quad (16)$$

The step-size of AD-conversion (see above) yields a resolution of $0.361\text{ }^\circ\text{C}/\text{step}$ for temperature.

C.1.2. PI-Control

Controller implementation by firmware allows for flexible design. The control of the discharge current $i_D(t)$ is based on the concept depicted in Figure [16] A. In section [B.1.1] the operating point was computed. At the set-point (i.e., at the predefined constant control voltage U_{D5}) the current sink operates at the specified discharge current I_{Test} . However, due to unavoidable parameter variation the actual discharge current $i_D(t)$ may display some offset from the set-point. We can write for the time dependent control voltage $u_{D5}(t)$

$$u_{D5}(t) = U_{D5} + \Delta u_{D5}(t). \quad (17)$$

Thus, control voltage is essentially determined by the constant, relatively large operating point voltage U_{D5} and a small time dependent voltage-offset $\Delta u_{D5}(t)$. This eases adjustment of current. For the current sink circuit shown in Figure [8], the voltage-offset $\Delta u_{D5}(t)$ can be estimated for a given discharge current $i_D(t)$ by

$$\Delta u_{D5}(t) \simeq \left(R_E + \frac{R_B}{\beta} \right) [I_{Test} - i_D(t)]. \quad (18)$$

Based on eq. [18] a classical proportional-integral (PI) feedback control was implemented (see Figure [16] A). The P-component of the control loop allows for quick reduction of the control-offset, while the I-component allows for continuously reducing control-offset to zero. The following notation is used for describing PI-control at discrete time steps ΔT_C :

- The abbreviation $r_C = R_E + R_B/\beta$ is introduced for the control loop resistance in eq. [18].
- At each time step t_m discharge current $i_D(t_m)$ is denoted by i_{Dm} . Analogously, the voltage-offset $\Delta u_{D5}(t_m)$ is denoted by Δu_{D5m} .
- The abbreviation P_m is used for the P-component at each time step.
- The abbreviation Σ_m is used for the I-component at each time step.

We aim for a current control which operates sufficiently fast, while avoiding oscillations (stable control). It is a good compromise designing PI-control such, that at each time step the control offset is approximately halved. We can then write for the P-component

$$P_m = \frac{r_C}{2} (I_{Test} - i_{Dm}). \quad (19)$$

For the I-component we aim for maintaining the effect of the P-component in the actual time step in the next time step. We can write

$$\Delta u_{D5m} = \Sigma_{m+1} = \Sigma_m + P_m \quad \dots \text{with } \Sigma_1 = 0. \quad (20)$$

Here, eqs. [19] and [20] define a weighted summation of the control-offset at each time step and, thus, an approximation of integration. Since the sum of P-component and I-component is per definition the output of a PI-controller, Σ_{m+1} equals the offset voltage Δu_{D5m} in the actual time step. Figure [16] B illustrates schematically the convergence of the discharge current with repeated control cycles. Since control offset is approximately halved at each step, PI-control approximately behaves like a first order linear system with a time constant of $\Delta T_C / \ln(2)$.

Figure [16] B shows the idealized behavior of the control. Real world parameter variations (in particular: warming of the transistor during operation, drop of battery voltage during discharge) are compensated by this control. Furthermore, the application of the controller at relatively short time steps “averages out” the discrete PWD-steps in U_{D5} over time. Thus, the average discharge current can be accurately controlled.

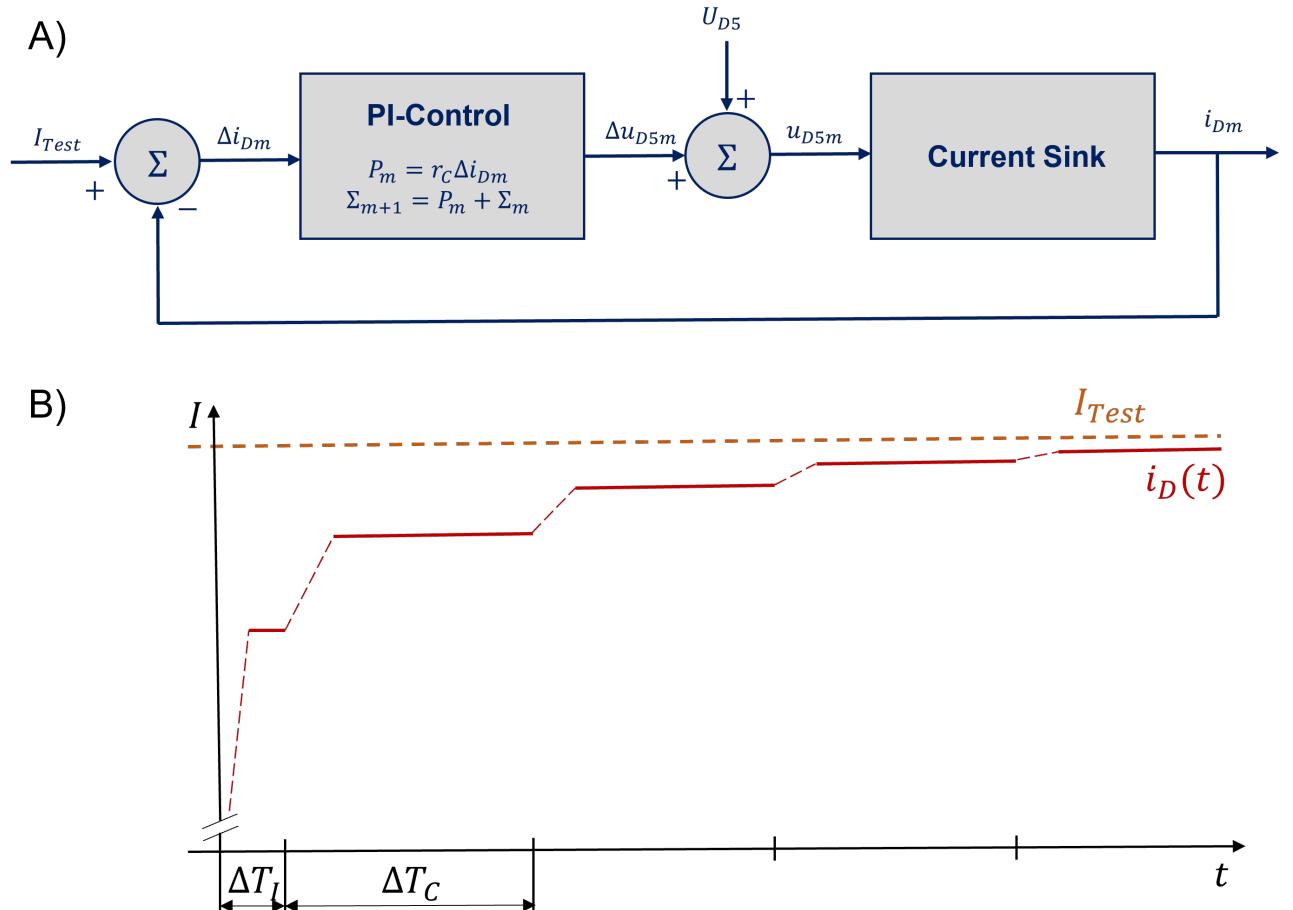


Figure 16: *Panel A:* Schematic of the implemented PI-control feedback loop. The nominal test current I_{Test} inputs as the set-value and the control offset is computed by comparison with the actual discharge current i_{Dm} (time step m). The PI-controller computes the updated voltage offset Δu_{D5m} . The voltage offset is added to the operation point control voltage U_{D5} . The actual control voltage u_{D5m} drives the current sink and adjusts the discharge current. *Panel B:* Iterative adjustment of the discharge current at power-up of the discharge phase. At $t = 0$ the system is turned on and control voltage is set to the operation point U_{D5} . Within the initialization time ΔT_I the discharge current approaches a stable value. It typically displays a small deviation from the set point (note that the y -axis is depicted with an offset from zero). A first control step is performed and after a short transition phase control offset is approximately halved. Control steps are repeated at a control interval ΔT_C and at each step control offset is approximately halved.

C.2. Risk Mitigation

RiMi-003 and RiMi-004 are voltage limits defined along with the test specifications and must be implemented in the firmware. Below more parameters limits are defined which must be incorporated into software.

RiMi-031 Maximal test discharge I_{Tmax} . Discharge must abort when a critical discharge level is exceeded (Risk-203). The limit was set at 20 % above the nominal test current (see TS-035).

- This check must be performed in the discharge phase (see Figure 3).

RiMi-033 Minimal test discharge I_{Tmin} . Discharge at insufficient current causes improper testing. The battery should be protected from unnecessary discharge in fault conditions for avoiding wear (Risk-207). The limit was set at 20 % below the nominal test current (see TS-033).

- This check must be performed in the discharge phase (see Figure 3).

RiMi-035 Maximal stand-by discharge I_{0max} . The battery must be protected from deep discharge (Risk-203). RiMi-011 provides counter-actions to stand-by discharge enforced by hardware. For the extremely unlikely condition of an hardware failure an complementary firmware counter-action is foreseen. The limit was set to approximately the three-fold of the maximal stand-by current obtained from hardware design. This avoids that “noisy” measurements terminate the test erroneously (see TS-037).

- This check must be performed in the waiting phase and during the voltage test (see Figure 3).

RiMi-037 Maximal test voltage V_{Tmax} . It should be possible to use the test device for testing alternative battery types ahead of potential use in the fictive product. They may deliver a voltage which is slightly above the specified upper limit for the actual battery type (V_{Bmax} , see MS-005). For avoiding thermal damage to the current sink unit (see Figure 8) test voltage may exceed V_{Bmax} only by approximately 20 % (see TS-039).

- This check must be performed in the discharge phase (see Figure 3).

RiMi-041 If a safety relevant check fails the firmware must be transferred into a *safe mode*.

- The safe mode is the phase at the end of the test (see Figure 3). It turns the current sink off permanently.

C.3. Firmware Implementation

The *Arduino Integrated Development Environment (IDE, version 1.8.13)* was used for firmware implementation. The open source *Arduino IDE* is available for free download which makes it attractive for this training project. However, it should be noticed that the very limited debugging tools of the *Arduino IDE* impose some restrictions for firmware development, testing and technical documentation. These restrictions were partially overcome by a step like development of the code. In each step one single routine was developed, added to the code and tested. Here, a detailed description of the code is provided. The listing of the code is then provided in section

C.5 (Listing **1**).

The header includes the libraries used for driving the liquid crystal display and defines a data structure for driving the duo-LED. The following groups of variables need global definition:

Constant parameters holding safety relevant test specifications (see Table **2**) or the definition of the current sink operating point (see section **B.1.1**) are defined globally. Thus, unique values are globally defined for this central design specifications. Variable names use at least five characters and are identical as in Table **2** for supporting unique identification.

The two variables used for driving the two output devices are defined globally: one object `lcd` for the liquid crystal display (class `LiquidCrystal_I2C.h`) and variable `lcd` for the duo-LED (structure `LEDblnk`).

Variable `TestCheck` which is used for aborting the test (if checks fail) must be defined globally (used in `setup()` and `loop()`).

Array `hms` holding the run-time of the actual test phase in hours-minutes-seconds format must be defined globally (used in `setup()` and `loop()`).

The implemented routines are described in a “top down” fashion. First (“top level”), the “main program” is described by its set-up routine and infinite loop. Next, the subroutines for the three test phases (defined in section **5.1**) are described. The *Voltage Test* and *Discharge Phase* both require accurate measurement of battery data. For reducing AD-conversion errors, oversampling is performed for achieving process gain **[6]**. Furthermore, routines for data assessment and control are listed. Finally (“bottom level”), the output functions are described.

C.3.1. Routines `setup()` and `loop()`

The *Arduino IDE* always requires two routines which substitute the “main program”: a `setup()` routine executed a single time at program start, and a `loop()` routine executed with infinite repetitions. Since test phases (in particular the discharge phase) must run only once, they were implemented in the `setup()` routine. The *safe mode* (see **RiMi-041**) was implemented in the infinite `loop()` routine, which is called after regular or irregular termination of a test phase. This design ensures that a test cannot start a second time after a failed check.

Listing **1** contains the `setup()` routine. Program execution starts with three blocks which set up the firmware for interaction with the shield:

- Firstly, for using the reference circuit described in section **B.1.2**, firmware is set the external mode. Note that the use of the internal mode might cause a physical damage to the *Arduino UNO* board (resistor R_{SR} provides only an untested protection). Thus, the internal reference must **not** be used.
- Secondly, the input-/output pin settings are defined such that firmware settings agree with wiring on the shield. While most of these settings are identically with the *Arduino* default settings, pin-modes are always set explicitly for ensuring proper function. For the input pins the *Arduino* “pull-up option” is **not** used. Instead, care was taken to uniquely define input levels by hardware connections on the shield.

- Thirdly, the control voltage U_{D5} is set to zero (ensure, that there is no unintended battery discharge after power on).

Next, initialization of outputs is performed. The serial USB-interface is set to its default rate (9600 baud). A higher speed is not required (every eight seconds two values are transferred). Furthermore, the LCD-display is made ready for use.

The three phases defined in section 5.1 are then executed (routines described below). A failed check during the voltage test skips the discharge phase. After the three tests, the LED color is set to red. Blinking is turned on if at least a single test failed.

The *safe mode* at the end of the three phases is implemented by a short infinite-loop structure. At each iteration the control voltage U_{D5} is set to zero (avoid unintended discharge). Battery data is recorded and displayed by the LCD together with the standby time. A delay of one second is foreseen for incrementing stand-by time hms properly and for generating a blinking interval for the LED.

C.3.2. Routine RunWaitingPhase01

First, control voltage U_{D5} is set to zero (avoid unintended discharge, redundant action) and the LED is set for blinking in green. A while loop is used for checking if a battery is inserted. If this is not the case, at each loop iteration the LCD prompts a message (insert Battery) and displays the measured data. The lowest specified battery voltage V_{Bmin} (see MS-007) was used as a threshold. Thus, also a partially discharged battery can be tested for documentation purposes.

After inserting a battery, the program waits for receiving input via the USB serial port and a message is prompted at the LCD (press ENTER). Thus, the user can start the test by pressing the ENTER key on a external computer (see Risk-101).

C.3.3. Routine RunVoltageTest01

First, control voltage U_{D5} is set to zero (avoid unintended discharge, redundant action) and the LED is set for continuous yellow. For reducing discretization errors, battery data is measured eight times and the mean value is transmitted via the serial board (process gain [6]). The eight measurements are organized in a for-loop. At each iteration actual battery data is displayed by the LCD display. Each iteration takes (approximately) 1 s. After loop termination, mean values are computed and checks (current & voltage limits) are performed. Furthermore, the mean values are printed via the USB-serial output for documenting the results of the stand-by (idle mode) battery test.

C.3.4. Routine RunDischargePhase01

This function was developed for execution in real time with accurate timing. It applies oversampling for reducing discretization effects. Floating point variables were used for implementing the calculations described in C.1.1 and C.1.2. According to the online *Arduino* documentation, division is the arithmetic float operation of longest duration. It takes less than 40 µs at the ATmega328P processor operating at 16 MHz.

First the LED is set for blinking in yellow. Next the control voltage U_{D5} is set to the operation point for starting discharge. A delay of 20 ms is foreseen for obtaining a steady-state emitter voltage in the current sink (see Figure 9). Note that this extremely short “turn-on” phase is negligible with respect to the 4 h test time T_{Test} (see TS-025).

The test time T_{Test} (i.e., duration of discharge) is specified by 4 h and an accuracy of $\pm 0.25\%$ (see section 5.3) is requested. Thus, the test time is $14\,400 \pm 36$ s. The following design measures were chosen for meeting this requirement: Three loops were used for implementing the three time intervals defined in the flow-chart of the test (Figure 3). While operations (instead of the delay command) were used for accurate control of timing in each loop. Buffers were foreseen for handling the required computation time of operations. Thus, the three nested loops can be described as follows:

- The innermost loop (loop counter k) runs the PI-control of the discharge current I_D (eight iterations per second). For reducing discretization error (AD-conversion and DA-conversion), the innermost moving average is computed after the eight repetitions. Here, the 32 bit float representation allows for process gain of the 10 bit AD-conversion value. At each iteration the loop interval was reduced from 125 ms to 120 ms yielding a buffer of 8×5 ms = 40 ms for operations in the middle loop.
- The middle loop (loop counter m) computes the mean for the innermost moving average and drives the LCD. Again, eight iterations are performed, for further reducing discretization errors by another outermost moving average. At each iteration the loop interval was reduced from 1000 ms to 990 ms yielding a buffer of 8×10 ms = 80 ms for operations in the outer loop.

At the end of the loop the limits TS-029, TS-033, TS-035 and TS-039 are checked.

- The outer loop (loop counter n) computes the mean for the outermost moving average and transfers the data via the USB serial port. Thus, the transferred data is the mean of $8 \times 8 = 64$ measurements performed over each interval of 8 s duration. As it follows directly from the law of large numbers, stochastic errors (such as, e.g. discretization) are reduced by a factor of eight by this design. The outer loop interval was set to exactly 8 s for obtaining accurate timing.

In total 1800 repetitions of the outer loop are performed for obtaining a nominal test duration T_{Test} of 4 h. Note that the applied buffers slightly prolong every last repetition in the middle loop and the innermost loop. Thus, the average loop cycles are equal to their definition in Figure 3.

C.3.5. Routines for Measurement & Control

PIcontrol01: The routine updates the PI-controller at each iteration as described in section C.1.2. Pointer notation is used for inputting & returning control voltage U_{D5} . In fault conditions (or for unexpected input), the computed control voltage may display (too) large deviations for the predicted operating point. Thus, control deviation is limited to ± 0.9 V from the operating point.

getBatteryData01: The routine reads the voltages U_{A0} (emitter) and U_{A1} (voltage divider at collector). It computes the discharge current I_D and the battery voltage U_B by applying the formulas listed in section C.1.1 using pointer notation.

getTemperature01: The routine reads the voltage U_{A2} (LM35 output). It computes the temperature as described in section C.1.1 and returns the temperature value.

C.3.6. Output Routines

LCDcountHMS01: The routine prints the run time of test phases in “hours-minutes-seconds” format (hh:mm:ss) on the LCD display. Ahead of calling the function the cursor must be set to the left-most digit (0, 0 in this application, Figure 5). The routine is called with the actual run time (global variable) and an integer increment which is updated at each call. First, the actual time stamp is displayed. Then the timer is updated. When setting the increment to +1 the timer counts upwards. When setting the increment to -1 the timer counts backwards (negative time is not allowed). Logic operations are used for restricting minutes and seconds digits to the interval [0 59].

LCDprintData01: The routine prints the actual ambient temperature Θ_A , discharge current I_D and battery voltage U_B on the LCD display, by using the layout (cursor positions) depicted in figure 5. Printing of temperature can be avoided by inputting a negative temperature value (the circuit depicted in Figure 10 B does not allow for recording sub-zero temperature).

LEDblink01: The routine drives the duo-LED by using a data structure of type LEDblnk. If blinking is turned on, it flips “on/off-mode” with every call. If mode is on for the current call, then the LED is set to the selected color (red ... 'r', yellow ... 'y' or green ... 'g'). Only one of these three color code characters can be used.

C.4. Test Results

The tests performed in this section provide the V&V (verification & validation) of the functionality of the firmware in combination with the hardware. Testing comprises: the verification of execution of program steps (depending on user input or recorded data), the verification of the accuracy of the recordings and demonstration of proper control of the discharge current. Figure 17 depicts the test circuit. Laboratory power supply LaPS-001 provided an adjustable test voltage V_{Test} . Multimeter MuMe-003 was used as an ampere meter measuring the test current (as the surrogate for the discharge current I_D). Multimeter MuMe-001 was used as a volt meter for measuring the surrogate of the battery voltage U_B . The circuit was connected to the terminal block of the shield, which was mounted onto an *Arduino UNO r3* board. A single *Arduino UNO r3* board label as A_GF01 was used for development and V&V. Via an USB-connection the *Arduino UNO* was connected with a laboratory computer and the shield was connected to the LCD (see section 5.2).

Simulated Discharge: Figure 18 depicts the scheme applied for testing of all four operation phases. Before starting the test, the laboratory power supply was turned off. Thus, when starting the program the device was tested to prompt (insert BATTERY) via the LCD (initial waiting phase). Next, the power supply was turned on and test voltage was set to 9.0 ± 0.2 V. Now, device was expected for prompting (press ENTER). A serial monitor was opened on the laboratory computer and recording was started by pressing enter. Now, the device was expected for performing the voltage test (8 sec duration).

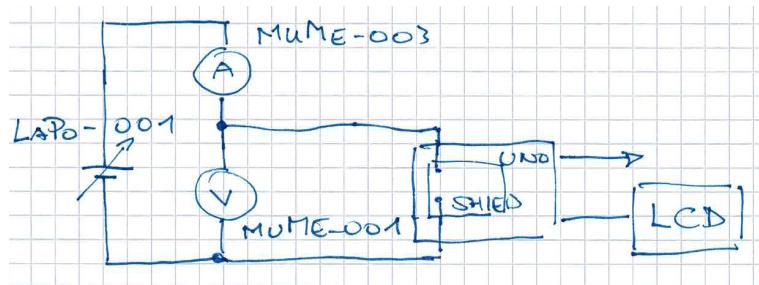


Figure 17: Test circuit for firmware V&V (see text).

At the transition of the voltage test to the discharge phase a stop-clock application was started on a smart phone app. This provided an independent timer for testing the timing of the discharge phase. Upon the device starts discharge, the battery voltage (i.e., the voltage on the terminal blocks of the shield) drops by some tenths of V (voltage drop across the ampere meter). Then, every two minutes the test voltage was reduced manually on the power supply for coarsely simulating a battery discharge behavior. A step size of approximately 0.5 V was chosen. After 10 min voltage had dropped to near 6 V (i.e., slightly above the allowed discharge limit V_{Emin} , see **TS – 029**) and it was not further reduced. Between the steps at odd minutes (i.e., 1, 3, ... up to 11) multimeter readings were taken (stable conditions in the middle of the steps). Ampere meter **MuMe-003** provides an average function. Thus, current readings were averaged over at least four seconds for reducing small current fluctuations due to PI-control.

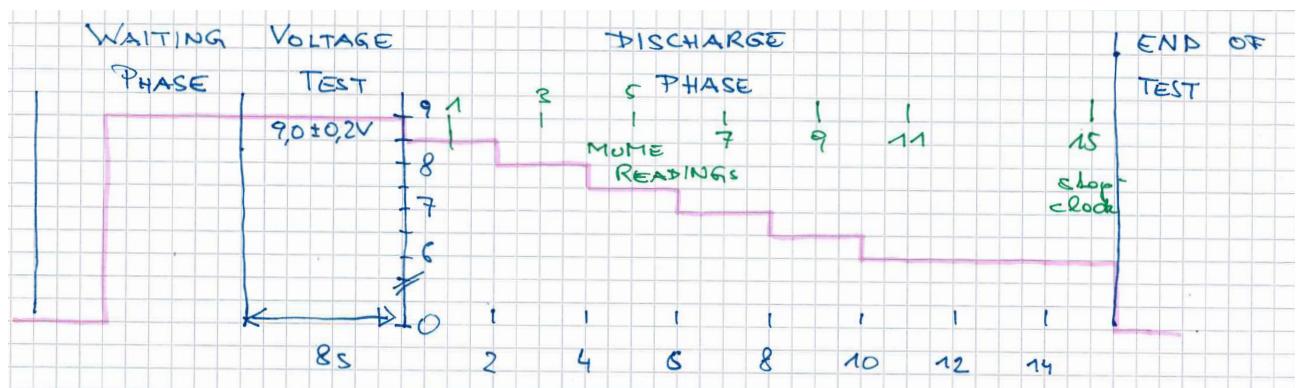


Figure 18: Scheme for testing of all four operation phases (see text).

At the time point 15 min:0s on the stop-clock (i.e., at 900 s), run time was simultaneously read from the LCD-display for verifying timing of the program execution. After the time point 15 min:30 s the test voltage was quickly reduced using the power supply for provoking a voltage drop below V_{Emin} . Here, the device was expected to interrupt discharge and the branch to the infinite loop at the end of test (*safe mode*).

For the test of both shields *A* & *B* the program properly executed all test cycles. In particular, the following observations were made:

- The device prompted (insert BATTERY) at start and moved over to (press ENTER) upon turning the test voltage on.

- [–] The LED was blinking green in the waiting phase.
- For all tests, the Voltage Test started when pressing ENTER. The timer on the LCD provided the count-down to the start of the discharge phase.
- [–] The LED was continuously yellow during the Voltage Test.
- After timeout of the Voltage Test discharge started automatically. The LCD displayed discharge time.
- [–] The LED was blinking yellow during the Discharge Phase.
- When reducing test voltage below its lower limit, the device switched to the *safe mode*. The LCD now displayed the time after the end of test.
- [–] The LED was blinking red, indicating a premature termination of the test.

Thus, the program properly controls the execution of all test phases.

Furthermore, data was correctly transmitted via the USB serial-port. Figure 19 displays current and voltage during the discharge phase. For both shields, PI-control adjusted current precisely to its nominal value $I_{Test} = 125 \text{ mA}$. When testing shield A it was possible to adjust battery voltage in steps. When testing shield B it was not possible to accurately adjust voltage on the laboratory power supply LaPS-001 (the current limit of the power supply was erroneously set too close to the nominal test current). However, it was possible to simulate a coarse transition of the test voltage from high to low levels and steady readings were obtained at “odd minutes”. Thus, the quality of both tests was acceptable.

Table 11 lists the relative deviations (firmware data vs. multimeters) observed for battery voltage U_B and discharge current I_D . They are well within the specified accuracy limits (see Table 6). For both shields the deviation in 900 s run-time between LCD and stop-clock was -1 s (i.e., -0.11%). Thus, also this value was within the specified accuracy limit.

Table 11: Relative Deviations during Simulated Discharge.

Component	U_B max	U_B mean	U_B Min	I_D max	I_D mean	I_D Min
Shield A	-0.24%	-0.44%	-0.53%	0.08%	0.01%	-0.08%
Shield B	-0.25%	-0.45%	-0.71%	0.48%	0.39%	0.32%

Voltage Test: The test scheme shown in Figure 18 provided a single voltage test for each shield. For investigating proper function of the voltage test at a lower voltage level (i.e., for insufficiently charged batteries) a second test was performed. Here, the test voltage was set to $7.50 \pm 0.05 \text{ V}$. This value was chosen slightly below the lower limit of the starting voltage V_{Smin} . Thus, discharge must not start after the voltage test and program execution must directly branch to the end-of-test (*safe mode*). The test circuit was the same as depicted in Figure 17 – except for using MuMe-001 as the ampere-meter and MuMe-003 as the volt meter. The reason for exchanging multimeters was taking advantage of the 4 mA current range of the APPA 105R for the small currents at stand-by. For the second test, program execution skipped discharge as expected for both shields. Table 12 lists the results.

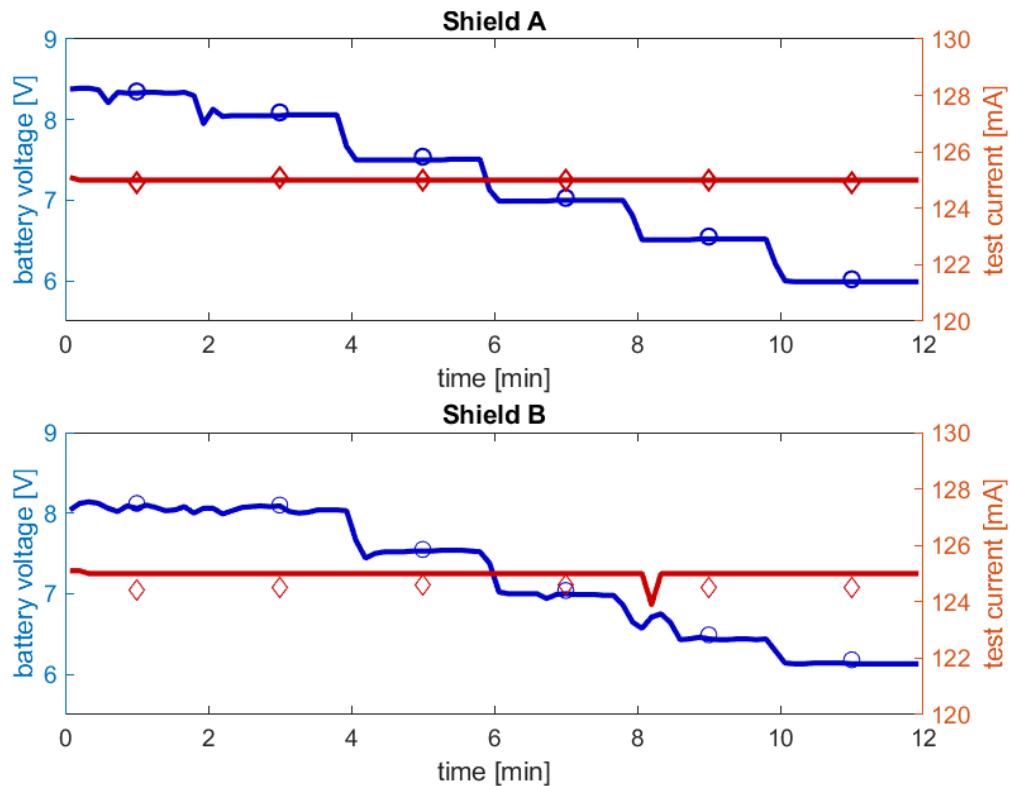


Figure 19: Current (red) and voltage (blue) during the discharge phase measured for shield A *top* and shield B *bottom*. Continuous line represent the data transmitted via the USB-port (*Ardunio* data). The markers represent the control measurements obtained from the multimeters (see text).

Deviations in voltage were in the range of -0.2% to zero and, thus, well within the specified range. The measured currents were in the expected range (load is composed by the $30\text{ k}\Omega$ serial resistance of the voltage divider circuit R_D, R_U (see Figure 8). USB data captured these small currents at the limit of its resolution.

Thus, the acquired data fulfills the accuracy requirements.

The USB-supply voltage delivered by the laboratory computer to the shield was 5.05 V (MuMe-002, same reading for both shields).

Table 12: Test Results for Voltage Test.

Component	U_B USB	U_B MuMe	I_D USB	I_D MuMe
Shield A	9.02 V	9.03 V	0.3 mA	0.31 mA
	7.50 V	7.50 V	0.3 mA	0.251 mA
Shield B	9.08 V	9.09 V	0.3 mA	0.31 mA
	7.48 V	7.50 V	0.3 mA	0.251 mA

C.5. Listing

Listing 1: Arduino Sketch

```

1 //*****
2 // Firmware for Test Device LiIonCheck9V125mA
3 // see Engineering Report ER-LiIonCheck9V125mA actual version for
4 // detailed description.
5 // (C) Gerald Fischer, 2022/04/01
6 *****
7 #include <Wire.h>
8 #include <LiquidCrystal_I2C.h>
9 typedef struct {
10   char color; // 'r'...red, 'g'...green, 'y'... yellow
11   bool on;    // true ... on, false ... off
12   bool blnk; // true ... blink, false ... continuous
13 } LEDblnk;
14 //
15 // Global Constants
16 //
17 const float ITest = 0.125; // [A] spc. test current, see TS-021
18 const float ITmax = 0.150; // [A] max. test current, see RiMi-031
19 const float ITmin = 0.100; // [A] min. test current, see RiMi-033
20 const float VTmax = 10.0; // [V] max. test voltage, see RiMi-035
21 const float VSmin = 8.0; // [V] min. voltage start, see TS-027
22 const float VEmin = 5.8; // [V] min. voltage end, see TS-029
23 const float UD5op = 4.1; // [V] controll voltage, operating point
24 //
25 // Global Variables
26 //
27 LiquidCrystal_I2C lcd(0x27, 16, 2);
28 LEDblnk LED;
29 bool TestCheck; // true = pass; false = fail
30 int hms[3]={0,0,0}; // global array hour-minutes-second
31 *****
32 void setup() {
33   // First - set internal reference
34   analogReference(EXTERNAL);
35   // Second - define pin-modes
36   pinMode(A0, INPUT); // analog pin A0 -> U_Emitter
37   pinMode(A1, INPUT); // analog pin A1 -> U_collector/3
38   pinMode(A2, INPUT); // analog pin A2 -> Temperature
39   pinMode(5, OUTPUT); // digital pin 5 -> PWM control voltage
40   pinMode(7, OUTPUT); // digital pin 7 -> LED green
41   pinMode(8, OUTPUT); // digital pin 8 -> LED red
42   // Second - set discharge current to zero - safety
43   analogWrite(5,0); // set pin D5, control voltage to zero
44   // Set up USB

```

```
45 Serial.begin(9600); // 9600 default boud rate
46 // init LCD-Display
47 lcd.init();
48 lcd.backlight();
49 // Wait
50 RunWaitingPhase01();
51 // Test Discharge
52 TestCheck=RunVoltageTest01();
53 // Test Discharge
54 if (TestCheck == true){
55     TestCheck=RunDischargePhase01();
56 }
57 // Set Time Counter zero for infinite loop
58 for (int k=0; k<3; k++){hms[k] = 0;}
59 // Set up LED for infinite loop
60 LED.color = 'r';
61 LED.on = true;
62 if (TestCheck == true){LED.blnk = false;}
63 else {LED.blnk = true;}
64 }
65 //*****
66 // Main loop -> SAFE MODE !
67 //*****
68 void loop() {
69     float Temp;
70     float IDch; // discharge current
71     float UBat; // battery voltage
72     //
73     analogWrite(5,0); // pin D5, set discharge current to zero - safety
74     lcd.clear();
75     LCDcountHMS01(1); // print time counter and increment by one
76     Temp = getTemperature01();
77     getBatteryData01(&IDch , &UBat);
78     LCDprintData01(Temp, IDch, UBat);
79     LEDblink01(&LED);
80     delay(1000);
81 }
82 //*****
83 // Run Waiting Phase
84 //*****
85 void RunWaitingPhase01(){
86     float IDch; // discharge current
87     float UBat; // battery voltage
88     //
89     analogWrite(5,0); // pin D5, set discharge current zero - safety
90     // set LED style
```

```

91   LED.color = 'g';
92   LED.on = true;
93   LED.blnk = true;
94   // wait for battery inserted
95   getBatteryData01(&IDch , &UBat);
96   while(UBat < VEmin){
97     LCDprintData01(0.0, IDch, UBat); // print current & voltage, Temp
98     =0
99     lcd.setCursor(0,0); // reposition cursor at origin
100    lcd.print(" _insert_BATTERY_ ");
101    LEDblink01(&LED);
102    delay(500);
103    getBatteryData01(&IDch , &UBat);
104  }
105  // wait for pressing Enter
106  while (!Serial.available()){
107    LCDprintData01(0.0, IDch, UBat); // print current & volt.,Temp=0
108    lcd.setCursor(0,0); // reposition cursor at origin
109    lcd.print(" _press_ENTER_ ");
110    LEDblink01(&LED);
111    delay(500);
112    getBatteryData01(&IDch , &UBat);
113  }
114 //*****
115 // Run VoltageTest
116 // returns true when voltage test was correctly completed
117 //*****
118 bool RunVoltageTest01(){
119   float nSumID=0.0; // sum for moving average current
120   float nSumUB=0.0; // sum for moving average voltage
121   float nMavID; // value of moving average current
122   float nMavUB; // value of moving average voltage
123   float Temp; // ambient temperature
124   float IDch; // discharge current
125   float UBat; // battery voltage
126   // init time counter & LED
127   hms[0]=8; // array hour-minutes-second start = 8 sec
128   hms[1]=0;
129   hms[2]=0;
130   LED.color = 'y';
131   LED.on = true;
132   LED.blnk = false;
133   //
134   analogWrite(5,0); // pin D5, set discharge current zero - safety
135   for (int n=0; n<8; n++){
136     // measure & sum up - but do no control!

```

```
137     getBatteryData01(&IDch , &UBat);
138     Temp = getTemperature01();
139     nSumID = nSumID + IDch;
140     nSumUB = nSumUB + UBat;
141     // LCD-print
142     lcd.clear(); // clear display and position cursor at 0,0
143     LCDcountHMS01(-1); // display time and decrement
144     LCDprintData01(Temp, IDch, UBat);
145     LEDblink01(&LED);
146     delay(980); // approx 1 s - 20 ms for remaining execution
147 }
148 // battery data - compute mean
149 nMavID = nSumID/8.0;
150 nMavUB = nSumUB/8.0;
151 // write to USB
152 Serial.print(nMavUB,2);
153 Serial.print(" ");
154 Serial.println(1e3*nMavID,1); // convert current to mA
155 // check & return test result
156 if (nMavUB < VSmin) {return false;} // RiMi-003
157 if (nMavID > 1e-3) {return false;} // RiMi-035
158 return true;
159 }
160 //*****
161 // Run Discharge Phase
162 // returns true when discharge was correctly completed
163 //*****
164 bool RunDischargePhase01(){
165     const float TnLoop = 8000; // [ms] duration m loop
166     const float TmLoop = 990; // [ms] duration m loop
167     const float TkLoop = 120; // [ms] duration k loop
168     float Temp; // ambient temperature
169     float Tpmin=100.0; // init with high Temperature
170     float Tpmax= 0.0; // init with low Temperature
171     float IDch; // discharge current
172     float UBat; // battery voltage
173     float UD5; // control voltage pin 5
174     long Tn; // time zero n-loop
175     int hms[3]={0,0,0}; // local array hour-minutes-second
176     //
177     LED.color = 'y';
178     LED.on = true;
179     LED.blnk = true;
180     // set operation point
181     UD5 = UD5op; // init operation point
182     analogWrite(5,round(UD5*51)); // pin D5, 51 = (2^8-1)/5 [V]
183     delay(20);
```

```
184 // outer USB n-loop
185 Tn = millis(); // get time zero n-loop
186 for (int n=0; n<1800; n++) {
187     while (millis()-Tn < n*TnLoop) {}
188     float mSumID=0.0; // sum for moving average current
189     float mSumUB=0.0; // sum for moving average voltage
190     float mSumTp=0.0; // sum for moving average temperature
191     float kMavID; // value of moving average current
192     float kMavUB; // value of moving average voltage
193     float kMavTp; // value of moving average temperature
194     long Tm; // time zero m-loop
195     // middle LCD m-loop
196     Tm = millis(); // get time zero m-loop
197     for (int m=0; m<8; m++) {
198         float kSumID=0.0; // sum for moving average current
199         float kSumUB=0.0; // sum for moving average voltage
200         float kSumTp=0.0; // sum for moving average temperature
201         long Tk; // time zero k-loop
202         while (millis()-Tm < m*TmLoop) {}
203         Tk = millis(); // get time zero k-loop
204         for (int k=0; k<8; k++) {
205             while (millis()-Tk < k*TkLoop) {}
206             // measure & control
207             getBatteryData01(&IDch, &UBat);
208             PIcontrol01(IDch, &UD5);
209             Temp = getTemperature01();
210             analogWrite(5, round(UD5*51)); // pin D5, 51 = (2^8-1)/5 [V]
211             // sum for average
212             kSumID = kSumID + IDch;
213             kSumUB = kSumUB + UBat;
214             kSumTp = kSumTp + Temp;
215         } // -> end k-loop
216         // k average, m summation
217         kMavID = kSumID/8.0;
218         mSumID = mSumID + kMavID;
219         kMavUB = kSumUB/8.0;
220         mSumUB = mSumUB + kMavUB;
221         kMavTp = kSumTp/8.0;
222         mSumTp = mSumTp + kMavTp;
223         // LDC-print
224         lcd.clear(); // clear display and position cursor at 0,0
225         LCDcountHMS01(1); // increment time counter by one
226         LCDprintData01(kMavTp, kMavID, kMavUB);
227         LEDblink01(&LED);
228         // 4 safety checks must be passed
229         if (kMavUB < VEmin || // RiMi-004
230             kMavUB > VTmax || // RiMi-037
```

```

231         kMavID < ITmin || // RiMi-033
232         kMavID > ITmax) // RiMi-031
233     {
234         analogWrite(5,0); // fail, set voltage zero at D5
235         return false;
236     }
237 } // -> end m-loop
238 // write UB & ID to USB
239 Serial.print(mSumUB/8.0,2);
240 Serial.print("L");
241 Serial.println(mSumID/8e-3,1);
242 // check temperature, store min or max
243 Temp = mSumTp/8.0;
244 if (Temp < Tpmin) {Tpmin = Temp;}
245 if (Temp > Tpmax) {Tpmax = Temp;}
246 } // -> end n-loop,
247 // discharge completed
248 analogWrite(5,0); // pin D5, reset discharge current to zero
249 // write min/max Temperature to USB
250 Serial.print(Tpmin,1);
251 Serial.print("L");
252 Serial.println(Tpmax,1);
253 // all tasked done, return test result
254 return true;
255 }
256 //*****
257 // Read analog input A0 (emitter volt.) and A1 (collector volt./3).
258 // -> Compute discharge current and battery voltage (see ER
259 // Appendix C.1.1.).
260 // Input/Output:
261 // *ID_p ... discharge current (pointer)
262 // *uB_p ... battery voltage (pointer)
263 //*****
264 void getBatteryData01(float *ID_p , float *uB_p){
265     const float A0toIC = 1.79e-4; //(124/125)*3.69[V]/1023/20[Ohm]
266     const float A1toIU = 3.61e-7; //3.69[V]/1023/10[kOhm]
267     const float A1toUC = 1.08e-2; //3*3.69[V]/1023
268     const float RF = 0.85; //fuse resistance in [ohm]
269     float rdA1; // analog read A1
270     float IC; // collector current
271     float IU; // current across resistor RU
272     float UC; // collector voltage
273     // read and compute
274     IC = A0toIC*float(analogRead(A0));
275     rdA1 = float(analogRead(A1));
276     IU = A1toIU*rdA1;
277     UC = A1toUC*rdA1;

```

```
278     *ID_p = IC+IU;
279     *UB_p = UC+RF * IC;
280 }
281 //*****
282 // Read analog input A2 (LM35 output) and return temperature
283 // see ER Appendix C.1.1
284 //*****
285 float getTemperature01(){
286     const float A2toTemp = 0.361; // 100 [K/V] x 3.69 [V] / 1023
287     return A2toTemp * float(analogRead(A2));
288 }
289 //*****
290 // compute PI-control (// see ER Appendix C.1.2
291 // Input
292 //      IDch ... discharge current (control reading)
293 // Input/Output:
294 //      *UD5_p ... control voltage at pin D5 (pointer)
295 //*****
296 void PIcontrol01(float IDch, float *UD5_p){
297     const float RC_2 = 14.03; // 0.5*(20 [Ohm]+1000[Ohm]/124[beta])
298     *UD5_p = *UD5_p + RC_2*(ITest-IDch); // PI-control step
299     if (*UD5_p < 3.2) {*UD5_p = 3.2;} // check lower limit
300     else if (*UD5_p > 5.0) {*UD5_p = 5.0;} // check upper limit
301 }
302 //*****
303 // LCD display of actual time in hours-minutes-second (hh:mm:ss)
304 // format & increment of time-count.
305 // Input:
306 //      increment ... positive or negative (i.e., decrement) integer
307 //      NOTE: decrement negative time is not correctly displayed
308 //*****
309 void LCDcountHMS01(int increment){
310     const int msmax = 59; // maximal count for minutes and seconds
311     // print time in hh:mm:ss format
312     for (int n=2;n>=0;n--){
313         if (hms[n]<10) {lcd.print('0');} // print "0 to 9" as "00 to
314         // 09"
315         lcd.print(hms[n]);
316         if (n>0) {lcd.print(':');}
317     }
318     // increment/decrement counter and maintain hour,min,sec format
319     hms[0]=hms[0]+increment;
320     for (int m=0;m<2;m++){
321         if (hms[m]>msmax){ // increment minutes/seconds maximal count
322             hms[m+1]=hms[m+1]+1;
323             hms[m]=0;
324         }
325     }
```

```
324     if (hms[m]<0) {      // decrement minutes/seconds minimal count
325         hms[m+1]=hms[m+1]-1;
326         hms[m]=msmax;
327     }
328 }
329 */
330 //*****LCD print of actual data. Cursor positions according to ER
331 // Section 5.2 (Figure 4)
332 // Input:
333 //   Temp      ... ambient temperature [oC]
334 //   IDch      ... discharge current [A]
335 //   UBat      ... battery voltage [V]
336 /*
337 void LCDprintData01(float Temp, float IDch, float UBat){
338     // print Temperature
339     lcd.setCursor(10, 0); // ... symbol 11 in first line.
340     if (Temp > 0.0){ // only positive temperature is allowed
341         lcd.print(String(Temp, 1));
342         lcd.print((char)223);
343         lcd.print("C");
344     }
345     // print current
346     lcd.setCursor(0, 1); // ... 1st symbol in 2nd line.
347     lcd.print(String(1e3*IDch, 1)); // convert to mA
348     lcd.print("mA");
349     // print voltage
350     lcd.setCursor(11, 1); // ... symbol 12 in 2nd line.
351     lcd.print(String(UBat, 2)); // directly in V
352     lcd.print("V");
353 }
354 */
355 //*****drive duo-LED
356 // Input:
357 //   *LED_p      ... pointer to LED structure (set LED on/off)
358 //*****LEDblink01(LEDblink *LED_p){
359 void LEDblink01(LEDblink *LED_p){
360     // start with both LEDs being turned off
361     digitalWrite(7,LOW);
362     digitalWrite(8,LOW);
363     // check blink
364     if (LED_p->blnk == true){
365         if (LED_p->on == true){LED_p->on = false;}
366         else                      {LED_p->on = true;}
367     }
368     // turn LED on?
369     if (LED_p->on == true){
```

```
371     if (LED_p->color == 'g' || LED_p->color == 'y') {
372         digitalWrite(7, HIGH);
373     }
374     if (LED_p->color == 'r' || LED_p->color == 'y') {
375         digitalWrite(8, HIGH);
376     }
377 }
378 }
```

D. Inspection, Measuring, and Test Equipment (IMTE)

In an industrial environment IMTE records are typically maintained by a proper structure within a quality management system (see section 7.6 in ISO 13485). For training purposes this document provides IMTE records within this Appendix. Since no external calibration was available, IMTE devices were tested in combination with each other as described below. Results were checked for plausibility and compared to available manufacturer specifications.

D.1. Multimeters

The three multimeters listed in Table 13 were used. Figures 20 and 21 show the result of a voltage test measurement and a current test measurement. During all test recordings it was verified that none of the multimeters displayed a warning for a low battery level (batteries supplying the multimeters).

According to their datasheets both the FLUKE 179 and the APPA 105R provide an accuracy of $\pm(0.1\% + 2 \text{ digit})$ for the voltage recording. This corresponds to an absolute error of $\pm 0.03 \text{ V}$ at the test voltage of 8.4 V. For the MM1.2 an accuracy of $\pm(0.5\% + 2 \text{ digit})$, i.e., $\pm 0.06 \text{ V}$ is specified.

For the current measurement the APPA 105R is specified for an accuracy of $\pm(0.4\% + 2 \text{ digit})$ (i.e., $\pm 0.7 \text{ mA}$ at 130 mA). The FLUKE 179 is specified for $\pm(1.0\% + 3 \text{ digit})$ (i.e., $\pm 1.6 \text{ mA}$ at 130 mA).

All observed deviations were below the specified limits. This justifies the assumption, that all multimeters MuMe-001, MuMe-002 and MuMe-003 fulfill their specified accuracy. However, multimeters displaying a low battery warning must not be used since their accuracy is not defined.

Table 13: Multimeters in Use.

multimeter ID	Device Type	serial nr.	label
MuMe-001	APPA 105R	82407652	GF private
MuMe-002	BENNING MM 1-2	n.a.	GF private
MuMe-003	FLUKE 179	11950279	UMIT Inv.Nr. 157/0910

D.2. Other Test Devices

The following other test devices were used:

LaPS-001 Laboratory Power Supply: BASETech Model: BT-153 (serial number 0391)

- adjustable DC voltage/current,
- tested in combination with multimeters.



Figure 20: Simultaneous measurement of voltage using the three multimeters (see Table [13]) MuMe-001 to MuMe-003 (from left to right) in parallel. A fully charged LiIon-battery block was used for providing a test voltage near the upper bound of the specified battery voltage (see Table [1], $V_{Bmax} = 8.4 \text{ V}$). Largest vs. smallest reading displays a deviation of approximately 0.2 %.

FuGe-001 Function generator: FeelTech (China) Model: FY3200S-24M (serial number 190532003853)

- generation of oscillatory test signals,
- tested in combination with oscilloscope Osci-001 prior to use.

Osci-001 Oscilloscope: Tektronix Model: TDS 1002B (serial number C010942), UMIT Inventar-Nummer 0018/06

- measurement of oscillatory test signals,
- tested in combination with function generator FuGe-001 prior to use.

ThCp-001 Thermo-Couple Electronic Thermometer: Greisinger Model: GTH 1170 (labeled “Gerald privat”)

- a calibration was performed on April 13, 2020. The deviation from zero in an ice-water bath was +0.1 °C.

All other test devices were tested prior to use. The oscilloscope was tested in combination with the function generator by simulating the operation point control voltage U_{D5} of the current sink (see Figure [22]).

Thus, proper function was verified for all test devices.



Figure 21: Simultaneous measurement of current using the three multimeters (see Table 13) MuMe-001 to MuMe-003 (from left right) in series. The laboratory power supply LaPS-001 was used for providing a test current near the specified nominal current of the battery block (see Table 1, $V_{BN} = 130 \text{ mA}$). The two readings with four digits resolution (MuMe-001 & MuMe-003) display a deviation of approximately 0.4 %.

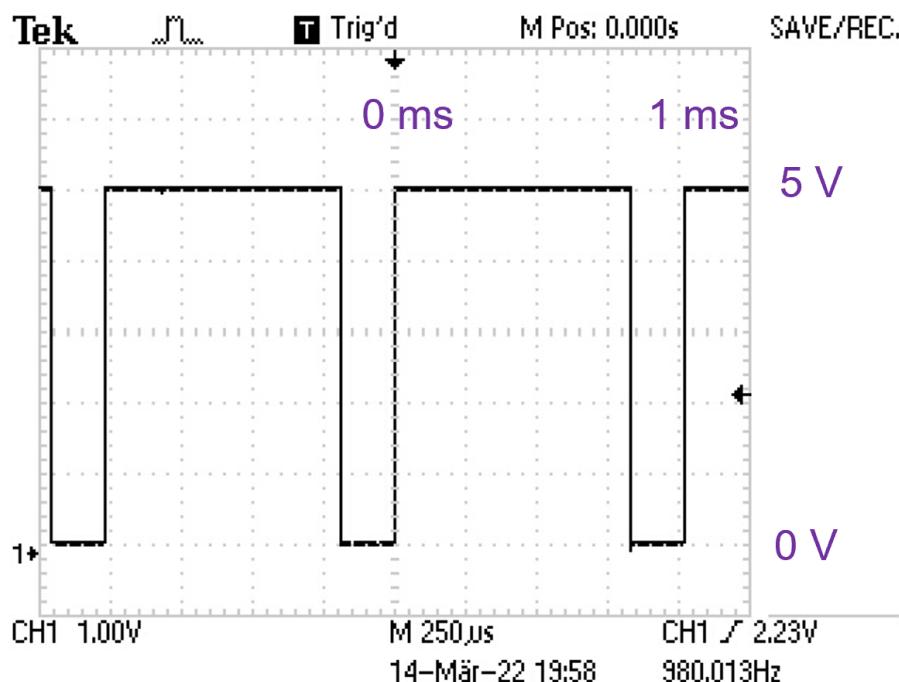


Figure 22: For testing the function generator FuGe-001 in combination with the oscilloscope Osci-001 the control voltage at the operation point of the current sink (see section B.1.1) was simulated. Thus, the output of the function was set to a square wave of 5 V amplitude with a frequency of 980 Hz and a duty cycle of 82 % (i.e., 4.1 V mean voltage). The recording of the scope properly reflects this setting.

E. Miscellaneous Battery Types

Different types of 9 V-blocks were tested for studying the performance of the developed test device. The *Arduino UNO r3* board labeled as *A_GF01* in combination with the shield labeled as *SLi22A* was used for test performed in this section. Measurements were carried out at room temperature between 20 °C and 25 °C.

E.1. Primary Batteries

E.1.1. Zinc-Carbon Battery

Historically, zinc-carbon batteries were developed at the end of the 19th century. They are the first type of primary cells brought into commercial use. However, their performance is limited. They were included into the analysis as an example of a battery type which is expected to fail the test. *VARTA* offers the zinc-carbon battery-model *SUPERLIFE* which was used for the test. Since no detailed datasheet was available for this battery model, reference data was extracted from the ANSI C18.1M Part 1 standard for 9 V zinc-carbon batteries (6F22). Highest battery voltage is 10.5 V (new cell without load). The highest test load described in the standard is approximately 15 mA (620Ω connected to the battery). This is approximately a factor of eight below the nominal test current of the *LiIonCheck9V125mA*. According to the standard, battery performance drops to 80 % after 12 month of storage.

A sample of type *SUPERLIFE* was used for the test performed on March, 17 2022. From the label at the bottom of the sample, storage time was identified by 10 month. A first test discharge was performed. Battery voltage dropped quickly and after approx. 140 s the test device stopped the discharge automatically since, voltage dropped below the cut-off limit V_{Emin} (see [23]). Since it was observed, that voltage increased again in the stand-by phase after the test, a second discharge was performed after a waiting period of 20 min. Again, discharge was terminated prematurely. For both discharge cycles, usable capacity at $I_{Test} = 125\text{ mA}$ was only about 5 mA h.

As it was verified from the data transferred via the USB serial-port, the test device accurately controlled the discharge current. Furthermore, the test device correctly controlled the execution of the test phases including a correct premature termination due to insufficient battery capacity.

E.1.2. Alkaline Battery

In the 1980ies alkaline batteries came into widespread use. While this battery type was initially promoted by the manufacturer *Duracell*, nowadays a broad spectrum of suppliers provide alkaline batteries. The tested battery was an *Energizer* model *MAX* 9 V block. For this model, a datasheet was available providing estimates for capacity (see Figure [24]). Figure [25] shows the results of the test discharge performed on March, 17 2022. From the label at the bottom of the sample, storage time was identified by 9 month. Near 2.8 h a small local minimum is observed in the voltage trace. In alkaline cells oxidation during discharge is split into two half reactions which can cause a local minimum. This effect caused also a recovery of voltage back to 7.68 V within 12 h after discharge.

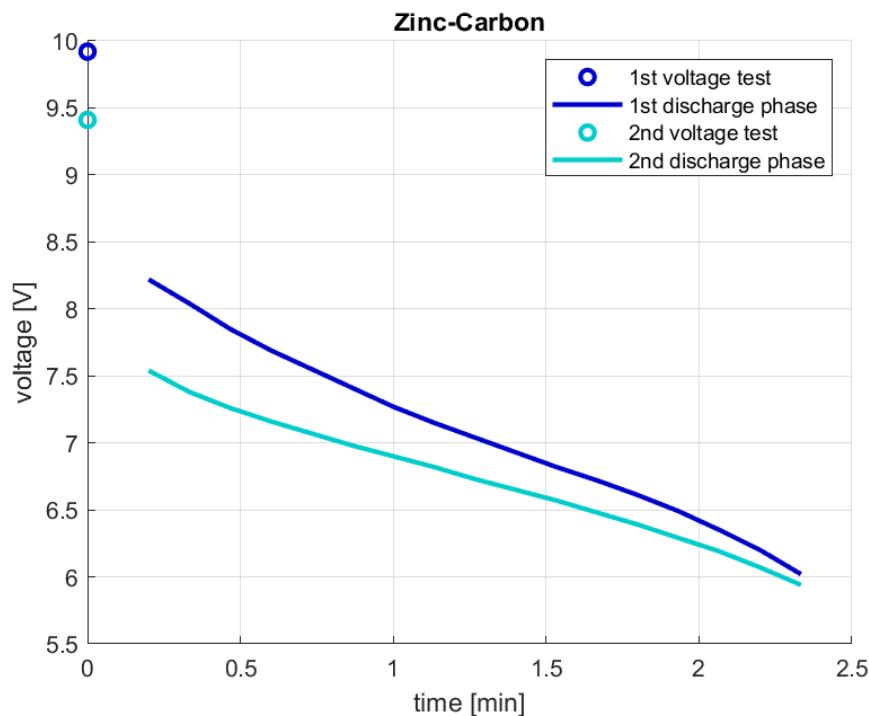


Figure 23: Results for two discharge cycles of a zinc-carbon primary cell (time scale in minutes, see text).

Again, the test device accurately controlled the discharge current including a premature termination of the test due to insufficient battery capacity. The measured capacity of the sample was 476 mA h. This is in good agreement with data extracted from the *Energizer MAX* datasheet.

E.1.3. Li-Primary Battery

Lithium Manganese Dioxide ($\text{Li}-\text{MnO}_2$) battery-packs came in widespread use in 9 V blocks due to their extremely low self discharge which renders them attractive in smoke detectors. However, they perform also well at relatively high load. Furthermore, they offer a higher capacity compared to alkaline-cells and operate in a wide range of ambient temperature. A 9 V block contains three cells. The tested battery was an *Energizer* model *L522* 9 V block. For this model, a datasheet was available, which allowed for estimating the capacity of the block by approximately 700 mA h. Figure 26 shows the results of the test discharge performed overnight on March, 18 and 19, 2022. From the label at the bottom of the sample, storage time was identified by 6 month. The measured test capacity was in good agreement with the capacity extracted from the data sheet.

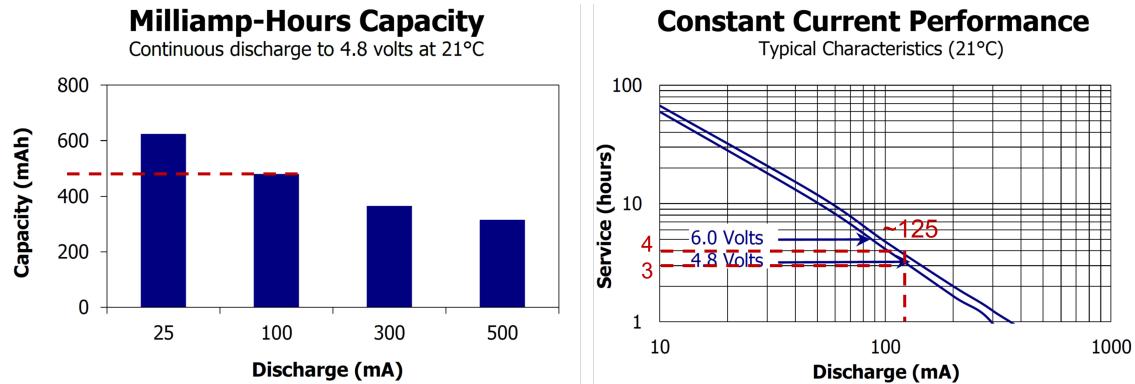


Figure 24: Two capacity estimates taken from the datasheet of the *Energizer MAX 9 V* block. The red lines were inserted for data extraction. *Left*: at 100 mA capacity is estimated somewhat below 500 mA h. *Right*: at 125 mA between 3 h to 4 h of service are expected, yielding a range of 375 mA h to 500 mA h for capacity.

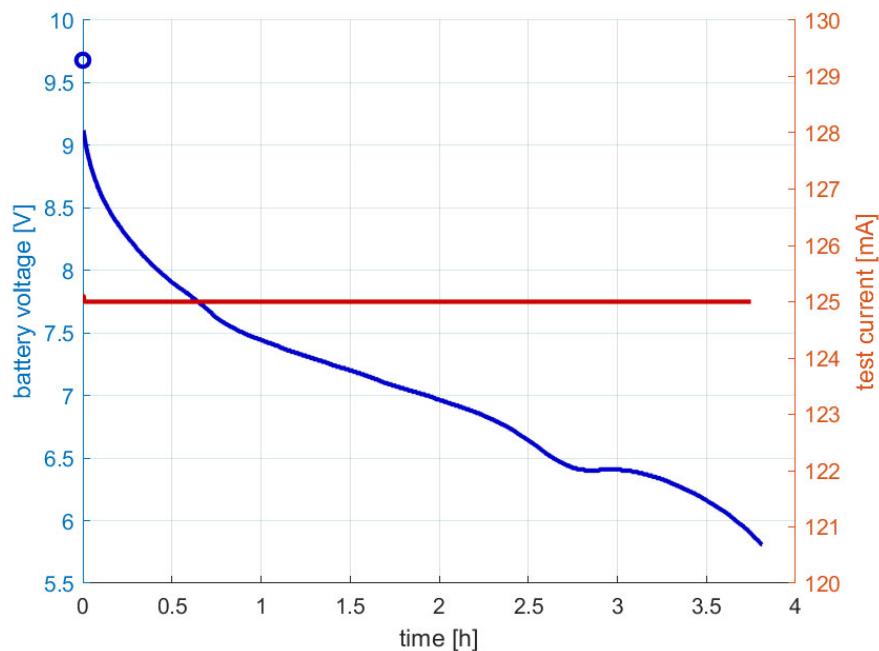


Figure 25: Results of the test discharge of the *Energizer MAX 9 V* block. The test device terminated discharge after 3 h:48 min:50 s, since voltage dropped below V_{Emin} . The measured test capacity was 476 mA h.

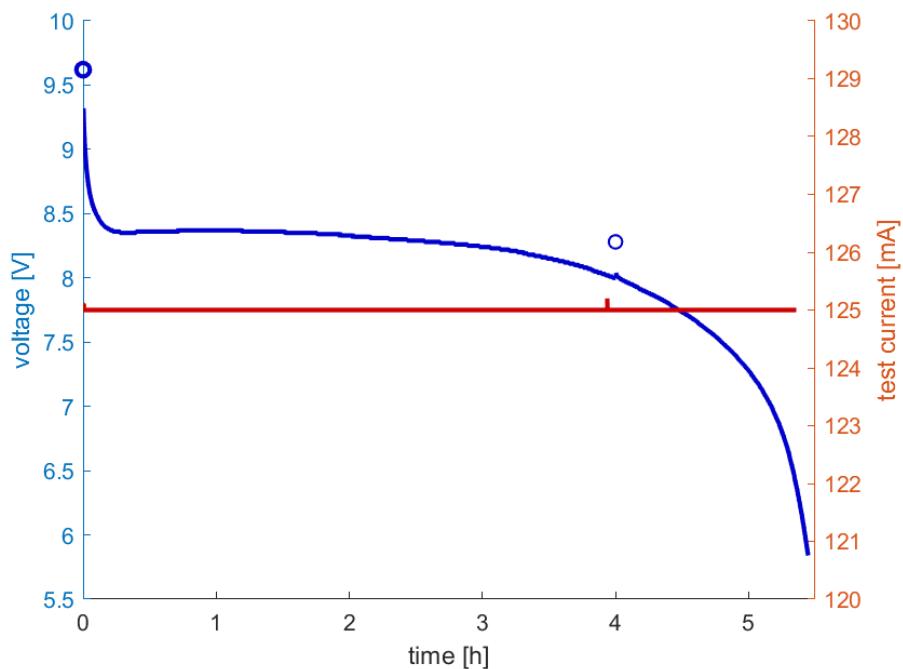


Figure 26: Results of the test discharge of the *Energizer L522 9 V* block. The test device terminated discharge after 4 h according to specification. Due to the high battery voltage at this termination it was possible to start a second test after only a few seconds until voltage dropped below V_{Emin} . The total discharge time of both tests was 5 h:26 min:52 s. The measured test capacity was 681 mA h.

E.2. USB-Rechargeable 9 V Block

Within recent years USB-rechargeable 9 V blocks became available. These are double cell LiCoO₂ polymer battery packs with polymer anode. They contain identical LiIon-cells as described in section 4.1 but are additionally equipped with a USB-socket for charging (see Figure 27). They contain circuits for controlling charging and a deep discharge protection. This allows for easy use of such battery-packs. However, the publically available technical documentation for such devices is poor. Therefore, the test device was used for measuring capacity of a USB-rechargeable 9 V block.



Figure 27: The USB-Rechargeable 9 V LiIon-block can be charged via a USB-micro socket on its bottom. A capacity of 650 mA h is printed onto the package. A duo LED indicates charging (red) and full battery (blue).

The tested block was labeled by *vhbw*, 9V, 650mA h, Li-ion (Model No.: BW888201291). According to the information provided at the web-site of *Conrad Elektronik* (checked on March 17, 2022) its minimum capacity is specified by 630 mA h. The battery was fully charged prior to test (blue-LED light on the bottom of the package). The voltage test after charging is depicted in Figure 20. Figure 28 depicts the test result. Surprisingly, the test terminated prematurely approximately one minute ahead on the end of the regular test interval T_{Test} . The Internal protection circuit shut down the battery when voltage dropped to about 6.1 V. During this shut-down the measured voltage was zero. After removal from the test device the shut-down was removed by recharging the battery again.

Again, the test device accurately controlled the discharge current including a premature termination of the test due to insufficient battery capacity. The measured capacity of the sample was 495 mA h. Thus, the test clearly demonstrated that the actual capacity is 24% below the value printed onto the battery.

E.3. Summary 9 V Blocks

Figure 29 allows for comparison of all tests performed in various battery types. 9 V zinc-carbon blocks are not suitable for operation at the test load. The alkaline block and the USB rechargeable 9 V block both clearly fail the test. The 650 mA h LiIon battery pack labeled by Accu A passes the test (see 5.4). Also the Li-MnO₂ battery passes and allows even for a second discharge phase due to its high voltage level.

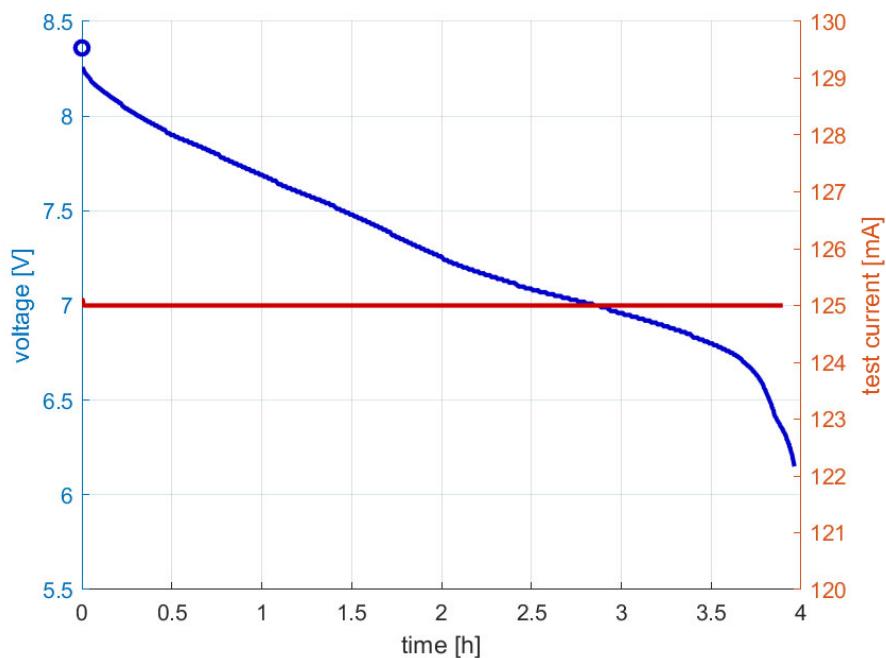


Figure 28: Results of the test discharge of the USB-rechargeable 9 V block. The test device terminated discharge after 3 h:58 min:56 s, since voltage dropped to zero (shut-down of the battery by its internal protection circuit). The measured test capacity was 495 mA h.

Concluding, the designed text device allows not only for reliable testing of the LiCoO_2 intended for use in the **fictive product** – moreover it allows for accurate comparison with other types of 9 V blocks.

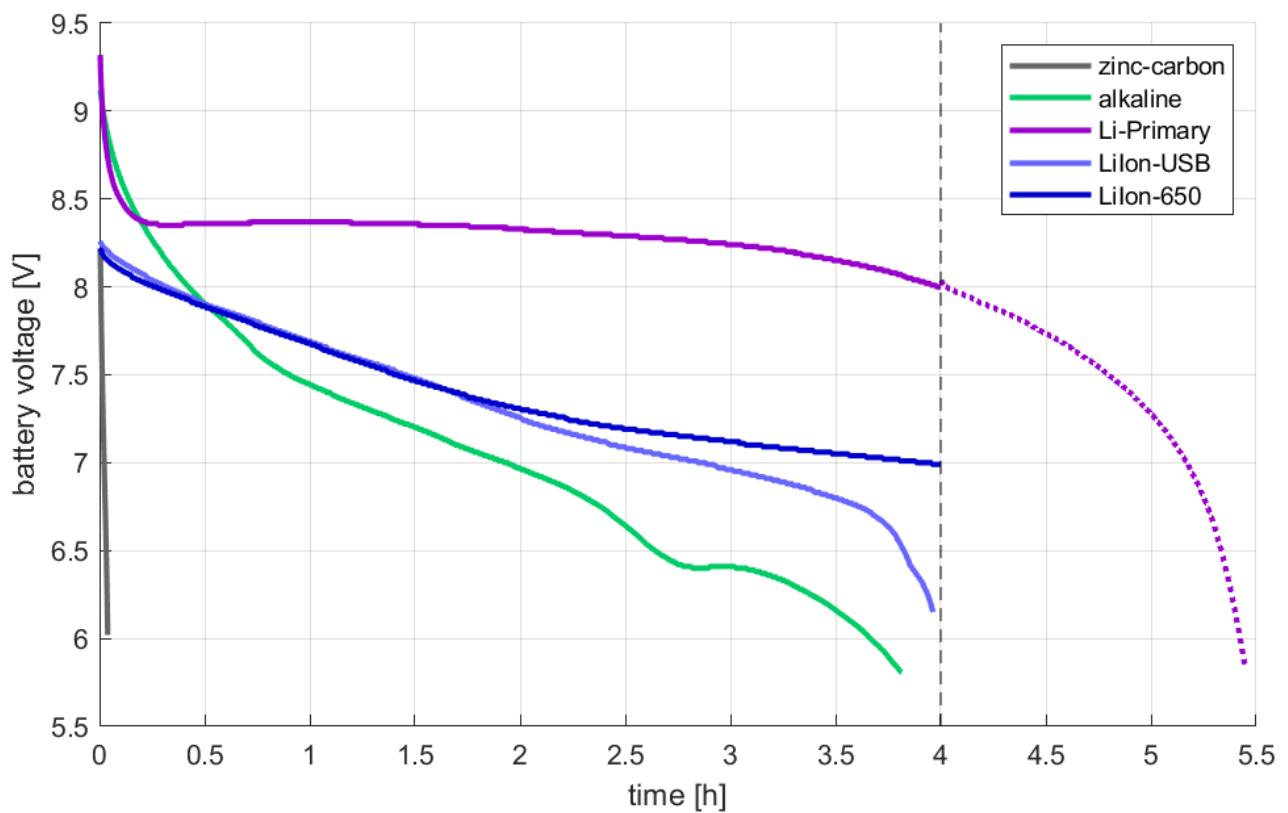


Figure 29: Test results for miscellaneous types of batteries (see text).

F. Material Cost

This section provides an overview on material cost. The document is intended for training purposes. The names of suppliers are not disclosed. The Numbers listed in Table 14 are the prizes paid during the development in 2022 (in Euro). It contains also cost for shipping. Here, materials for building five units were ordered. In the Table shipping cost were divided per five.

Table 14: Prizing of Materials – Test Device *LiIonCheck9V125mA v1.a*.

Item #	Part	Short Name	Price per Unit
<i>BOM for Device (see Tab. 5)</i>			
1	main board	<i>Arduino UNO r3</i>	24 Euro
2	shield board	<i>Shield</i>	5 Euro
3	liquid crystal display	<i>16 × 2 digits</i>	7 Euro
4	battery holder	<i>9 V block</i>	4 Euro
5 to 8	lab material	parts in stock	5 Euro
9	mounting plate	PVC	7 Euro
–	USB-cable	B-type	8 Euro
–	BOM for shield (see Tab. 8)		10 Euro
Total			70 Euro

Based on the experience obtained when building the prototypes a trained operator can assemble one unit in about four working hours. This estimate does not involve time for final testing.

Concluding, the developed design meets the requirement defined in URS-011.