

Neues Leben für alte Batterien

- Autor
- Zielgruppe
- Neues Leben für Batterien
- Testkriterien
- Batterietypen
- Idee einer Prüfanlage
- Prüfanlage Funktionen
- Mechanik
- Schiefe Ebene
- Revolver
- Messvorrichtung
- Batterie vorhanden?
- Lichtschranke
- Sortiereinrichtung
- Aktoren
- Revolver/Schrittmotor
- Revolver justieren
- Elektronische Bauteile
- Schaltung
- Arduino-Pinbelegung
- Eingangsspannung
- Stromversorgung
- Stromstärken
- Analoge Messwerte
- Schutzschaltung
- Programmierung
- Funktionen I & II
- Libraries
- SD-Karte
- Inbetriebnahme
- Anwender-Interface
- Prüfanlage Funktionen
- Ergebnisse & Schlusswort

Autor

| | |
|-----------|--|
| Name | Dipl.-Ing. (FH) Enno Klatt |
| Lehre | Kraftfahrzeugmechaniker |
| Studium | Maschinenbau, Apparatebau, Verfahrenstechnik |
| Stationen | <ul style="list-style-type: none">• Projektingenieur (Meeresbergbau)• Datenverarbeitung (IBM-PC XT)• Schulungsleiter (Öl- und Gasbrenner)• IT-Consultant (Expo Hannover)• Lehrer in Informatik, Technik, Physik• Lehrbeauftragter CAD (Hochschule Hannover) |

Zielgruppe

| | |
|--------------------------------------|--|
| SchülerInnen | Projekt im Fach Technik. |
| Kerncurriculum für die Oberschule | Handlungsbereich 2: Energie und Technik. |
| Unterricht | <ul style="list-style-type: none">• Lernen durch Handeln.• Methode Versuch und Irrtum.• Handwerkliche Fähigkeiten.• Mechanik, Elektronik, Programmierung.• ... |
| Material | <ul style="list-style-type: none">• Halbzeuge der UMT-Technik. Arduino, Steckbrett, diskrete Bauelemente.• Sensoren und Aktoren.• ... |
| Pilotanlage | Die vorgestellte Lösung ist eine von vielen! |
| Projekt-Dateien | https://github.com/EKlatt/AAA-Tester |
| UMT-Technik | https://technik-lpe.de/technik/ |

Neues Leben für Batterien

Schulprojekt
Taschenlampe

μTaLa
LED-Taschenlampe mit
AAA-Batterie.

AATiS Bausatz

AS332

Geschichte

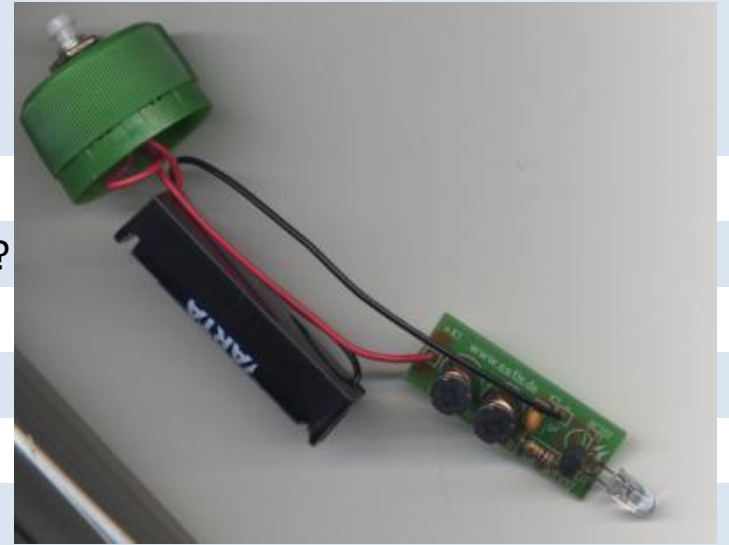
- Gebrauchte Batterien nutzen?
- Ausreichende Kapazität?
- AA-Batterien (AAA-Batterien)
- Funktionstest mit Voltmeter
- Mindestens 1,0 V
- Schülerversuche

Projektidee:

Wolfgang Lipps

Automatisierung

Prüfanlage für einen automatischen Betrieb.



Testkriterien

Stromaufnahme μTaLa : $I \approx 10 \text{ mA}$

Auslegung Lastkreis:

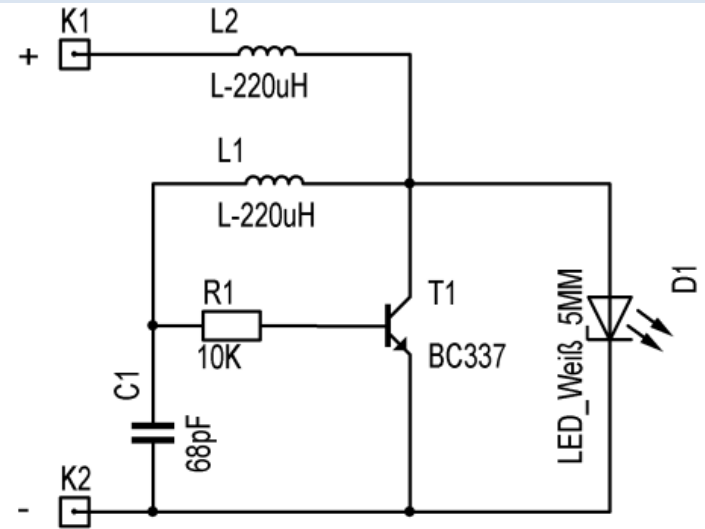
Mindest-Spannung $U = 1 \text{ V}$

Last-Widerstand $R = U / I$
 $R = 100 \Omega$

Angenommene
Messdauer 30 s

Entscheidung Test-Spannung $> 1,0 \text{ V}$,
dann weitere Verwendung.

Quelle [Microsoft Word - \$\mu\text{TaLa}\$ -Aufbau \(aatis.de\)](https://aatis.de)



Batterietypen

| | |
|--------------|-------------------------------|
| AA-Batterie | Alkali-Mangan, Zink-Kohle |
| Durchmesser | 13,5 mm bis 14,5 mm |
| Länge | Länge von 49,2 mm bis 50,5 mm |
| Nennspannung | 1,5 V |



| | |
|--------------|-------------------------------|
| AAA-Batterie | Alkali-Mangan, Zink-Kohle |
| Durchmesser | 9,5 mm bis 10,5 mm |
| Länge | Länge von 43,3 mm bis 44,5 mm |
| Nennspannung | 1,5 V |



Form
der
Fläche

| | |
|------|--|
| Link | https://de.wikipedia.org/wiki/Mignon_(Batterie) https://de.wikipedia.org/wiki/Micro_(Batterie) |
|------|--|

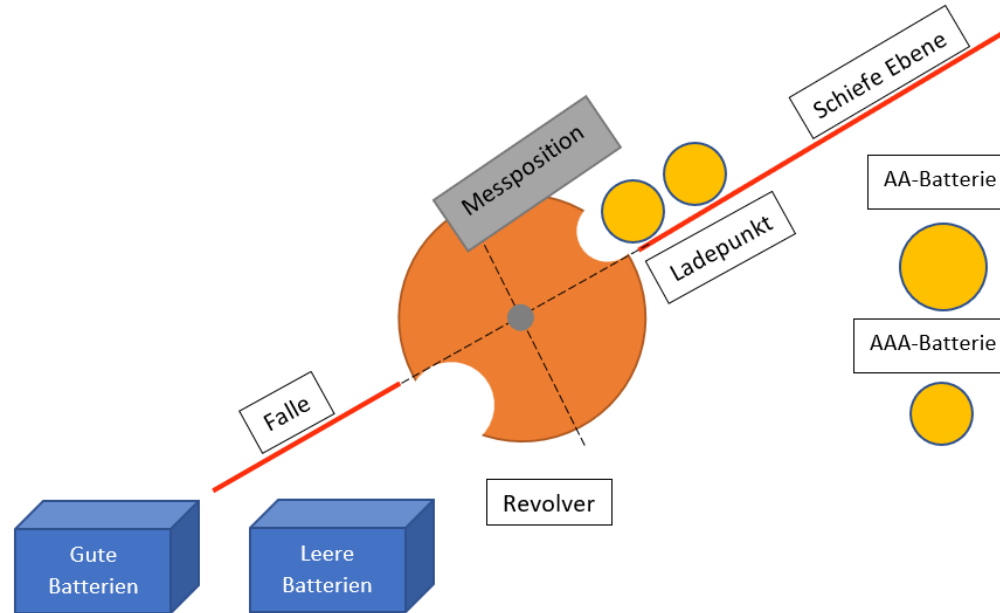
Idee einer Prüfanlage

Batteriezufuhr

„Schiefe Ebene“

Prüfanlage:

- Schiefe Ebene
- Revolver
- Messvorrichtung
- Falle
- Sortierbehälter
- Arduino
- Steckbrett



Länge der schiefen Ebene

ca. 600 mm (60 AAA-Batterien mit ≈ 10 mm \varnothing)

Prüfanlage Funktionen

| | |
|------------|--|
| Einrichten | 0. Prüfvorrichtung vorbereiten. |
| Zuführen | 1. Batterie automatisch zuführen. 2. Batterie zur Messposition bewegen. |
| Messen | 3. Ist eine Batterie vorhanden? 4. Spannung prüfen. 5. Haben wir eine gute oder leere Batterie? 6. Batterie zur Sortierung bewegen. |
| Sortieren | 7. Gute Batterie in Behälter „Gute Batterie“. 8. Leere Batterie in Behälter „Leere Batterie“. |
| | 9. Weiter mit Schritt 1. |

Mechanik

Mechanische Aufgaben?

- Bau einer „schiefen Ebene“ mit einer glatten Oberfläche.
- Optimieren der Neigung.
- Bau eines Batteriewechsers (Revolver).
- Entwickeln einer Messvorrichtung für die Batteriespannung.
- Erkennen einer fehlenden Batterie (Endschalter/Lichtschranke).
- Bau einer Sortiereinrichtung (Falle).



Schiefe Ebene



Anforderungen

- Sicherer Transport der Batterien.
- Seitliche Führungen für AAA- und AA-Batterien.
- Einfache Bestückung.

Material

PVC-Hartschaumplatte 3 mm dick.

Neigung optimieren

- Sicheres Nachrutschen der Batterien.
- Der Revolver darf nur eine Batterie mitnehmen.
- Den Schrittmotor nicht überlasten.

Empfehlung

Die Neigung der schiefen Ebene minimieren.
Neigung: ca. 20° zur Horizontalen.

Revolver

Anforderungen Batteriewechsler

- Sicherer Transport der Batterien.
- Anzahl der Bauteile und Antriebe minimieren.
- Einfacher Aufbau.
- Sichere Mitnahme einer Batterie.
- Einhalten der Messposition.
- Sicherer Transport zur Sortierung.

Randbedingungen

- Maße der AAA- und AA-Batterien.
- Lage der schiefen Ebene.
- Aufbau der Messeinrichtung.
- Lage der Falle.

Material

3D-Druck oder Hartholz-Zylinder.

Konstruktion

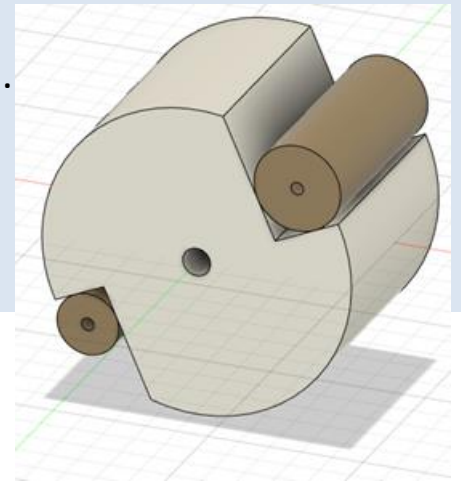
Nutformen für AAA- und AA-Batterien.

Empfehlung

3D-Druck

Link

<https://github.com/EKlatt/AAA-Tester>
siehe Ordner „3D“.



Messvorrichtung

Anforderungen

- Messen der elektrischen Spannung im Leerlauf und unter Last.
- Soll sich der Bauform der AAA- und AA-Batterie anpassen.
- Einen guten Kontakt an den Polen gewährleisten.

Material

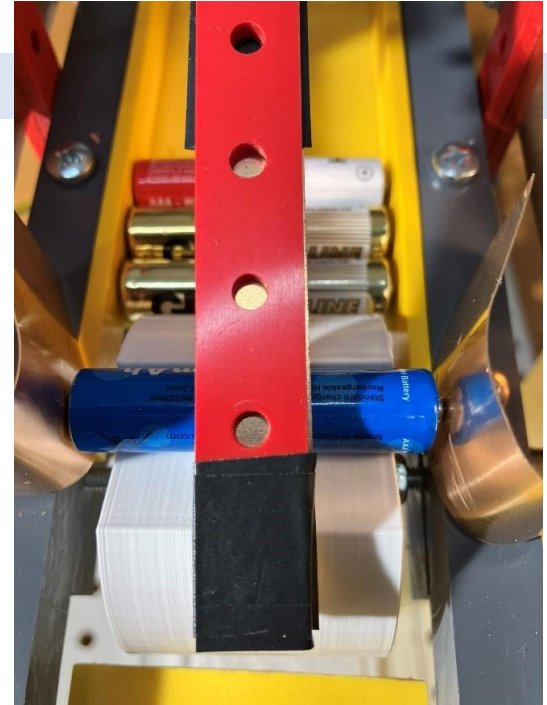
Idee: Blechstreifen aus Federbronze.

Herstellung

Blechstreifen von Hand geformt.

Probleme

- Formgebung von Hand.
- Blechstreifen dürfen das Zuführen der Batterie nicht behindern.
- Automatische Anpassung an die beiden Bauformen.
- Kontaktschwierigkeiten in Folge unterschiedlicher Bauformen, insbesondere des Minus-Pols.



Batterie vorhanden?

Anforderungen

- Die Prüfanlage soll automatisch betrieben werden.
- Solange eine Batterie vorhanden ist soll geprüft werden.
- Ist keine Batterie vorhanden, soll der automatische Betrieb anhalten.

Randbedingungen

- Durchmesser der AAA- oder AA-Batterien.
- Lage des Revolvers.

Material

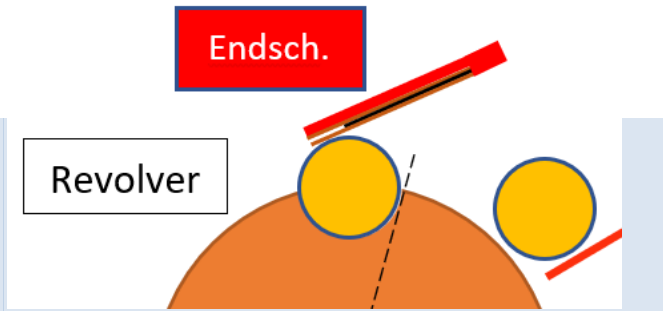
- UMT-Lochstreifen
- Kontakte aus Federbronze
- Isolierband

Probleme

- Abstand Batterie Endschalter je nach Batterietyp.
- Automatische Anpassung.
- Lage des Revolvers.
- Kontaktmechanismus

Optimierung

- Endschalter mit zu kurzem Kontaktweg.
- Alternative Lichtschranke!



Lichtschanke

Anforderungen

- Die Prüfanlage soll automatisch betrieben werden.
- Solange eine Batterie vorhanden ist soll geprüft werden.
- Ist keine Batterie vorhanden, soll der automatische Betrieb anhalten.

Randbedingungen

- Durchmesser der AAA- oder AA-Batterien.
- Lage des Revolvers.

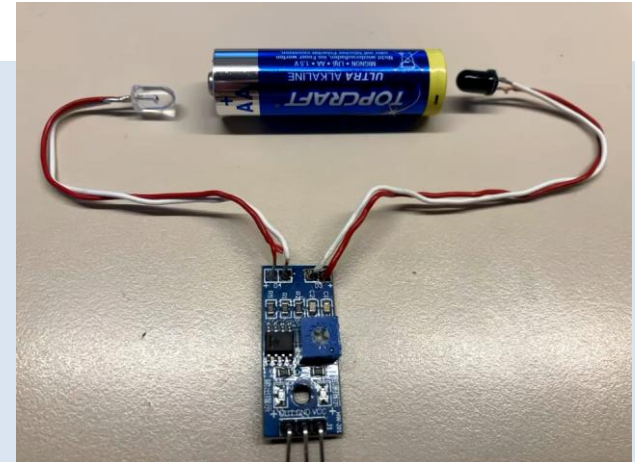
Bauteil

Abstandssensor Modul einstellbar
Digitaler Ausgang (0/1): „1“ entspricht Batterien bereit.

vorher

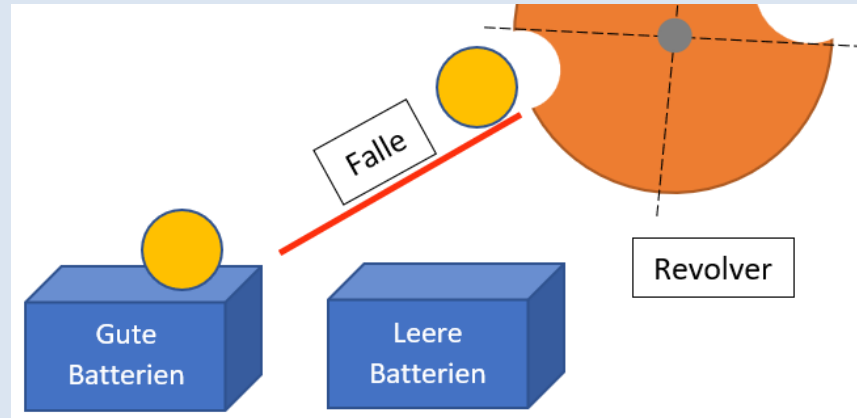


Nach Umbau

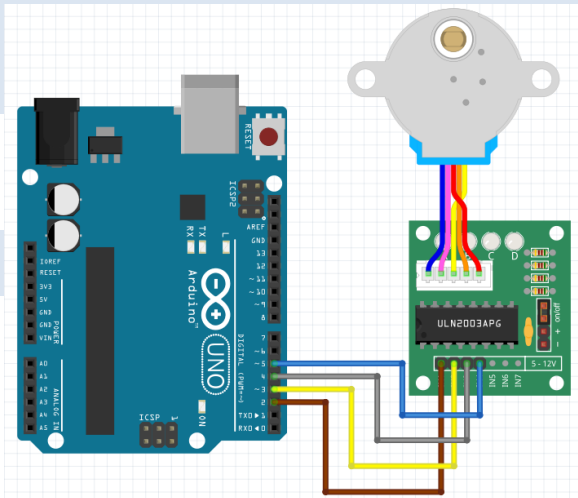



Sortiereinrichtung

| | |
|-----------------|---|
| Anforderungen | Einsortierung in einen Behälter für gute oder leere Batterien. |
| Randbedingungen | <ul style="list-style-type: none">• Einfacher Mechanismus.• Sichere Sortierung.• Einfacher Antrieb. |
| Material | <ul style="list-style-type: none">• PVC-Hartschaumplatte 3mm dick.• UMT-Vierkantstab (12 mm x 12 mm).• Servo. |
| Probleme | Lage (Neigung) zum Revolver. |



Aktoren

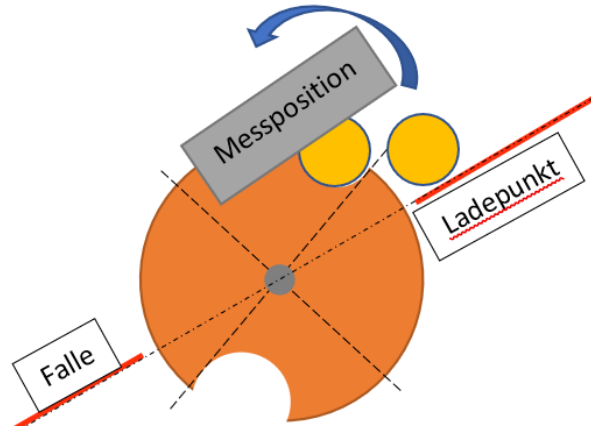
| | | |
|-----------------------|---|---|
| Schrittmotor | 28BYJ-48 Stepper Motor |  |
| Schritte 1. Umdrehung | 2048 ca. 220 mA | |
| Pinbelegung | <pre>Stepper stepper(..., Arduino Pin motor_pin_1, (2) motor_pin_2, (4) motor_pin_3, (3) motor_pin_4) (5)</pre> | |
| Servo | SG90 |  |
| Max. Moment | 18 Ncm (1.8 kgf·cm) | |
| Pinbelegung | <pre>#define servoPin (7)</pre> | |

Revolver/Schrittmotor

Ladepunkt -> Messposition

Messposition -> Falle

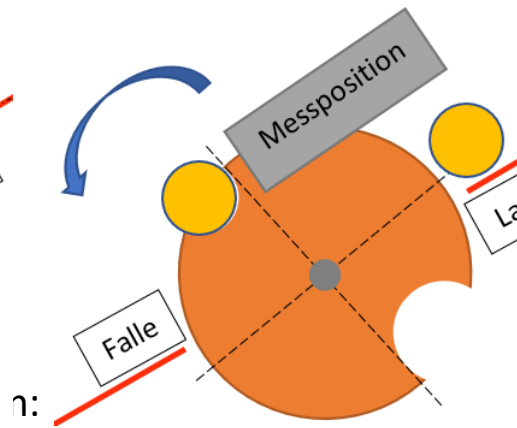
Falle -> Ladepunkt



390 Schritte

Terminal: „s:390“

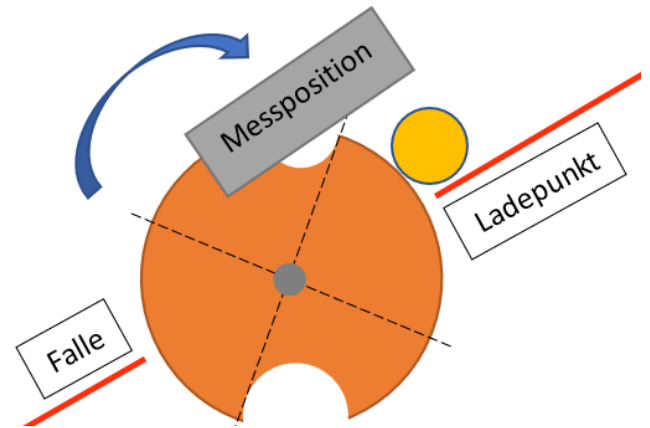
Sketch: stepsLoadTest



350 Schritte

Terminal: „s:350“

Sketch: stepsTestTrap



- (390 + 350) Schritte

Terminal: „s:-740“

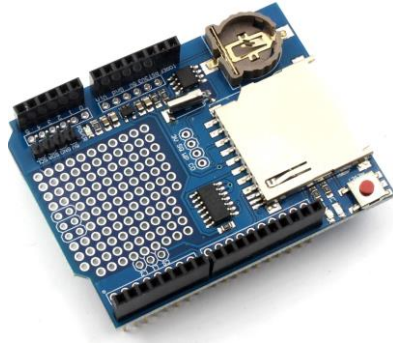
Siehe auch Parameter in der „config.ini“ auf der SD-Karte.

Revolver justieren

| | |
|-----------------------|--|
| Achtung | Die jeweilige Revolver-Nut und die schiefe Ebene müssen fluchten. |
| Mechanische Anpassung | |
| Länge Batterie | Die Breite der schiefen Ebene durch entfernen/einlegen von ungelochten UMT-Vierkantstäben (12 mm x 3 mm) anpassen. |
| Revolver Nut | Der Revolver hat zwei Nuten. Eine Nute für AAA- und AA-Batterie. |
| Justierung | Vor Beginn eines Prüflaufs muss die entsprechende Nut mit dem Ladepunkt gegenüber der schiefen Ebene ausgerichtet werden. |
| Rechter Taster | Einmalige Betätigung lässt Revolver/Schrittmotor um 10 Schritte gegen den Uhrzeigersinn drehen. |
| Terminal s-Befehl | Sinnvolle Werte: von (+1) bis (+2048) und (-1) bis (-2048). Befehl: „s:Schritte“ Gegen den Uhrzeigersinn: „s:10“. Im Uhrzeigersinn: „s:-10“ |

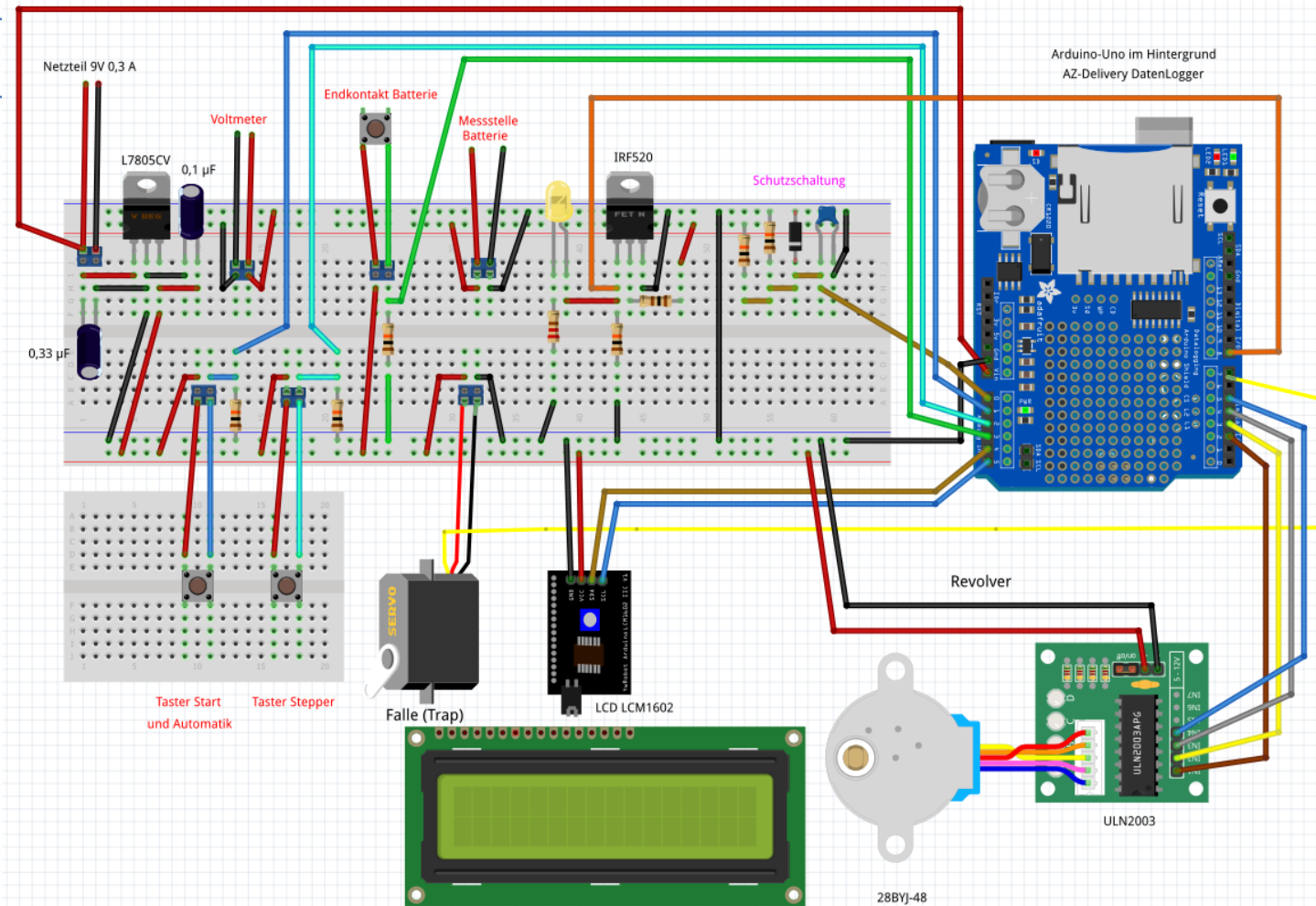
Elektronische Bauteile

| | |
|-----------------|---|
| Mikrokontroller | Arduino-UNO mit AVR® 8-bit Mikrokontroller. |
| Datenlogger | Data Logger Shield mit RTC. Das Shield wird auf den Arduino aufgesteckt. |

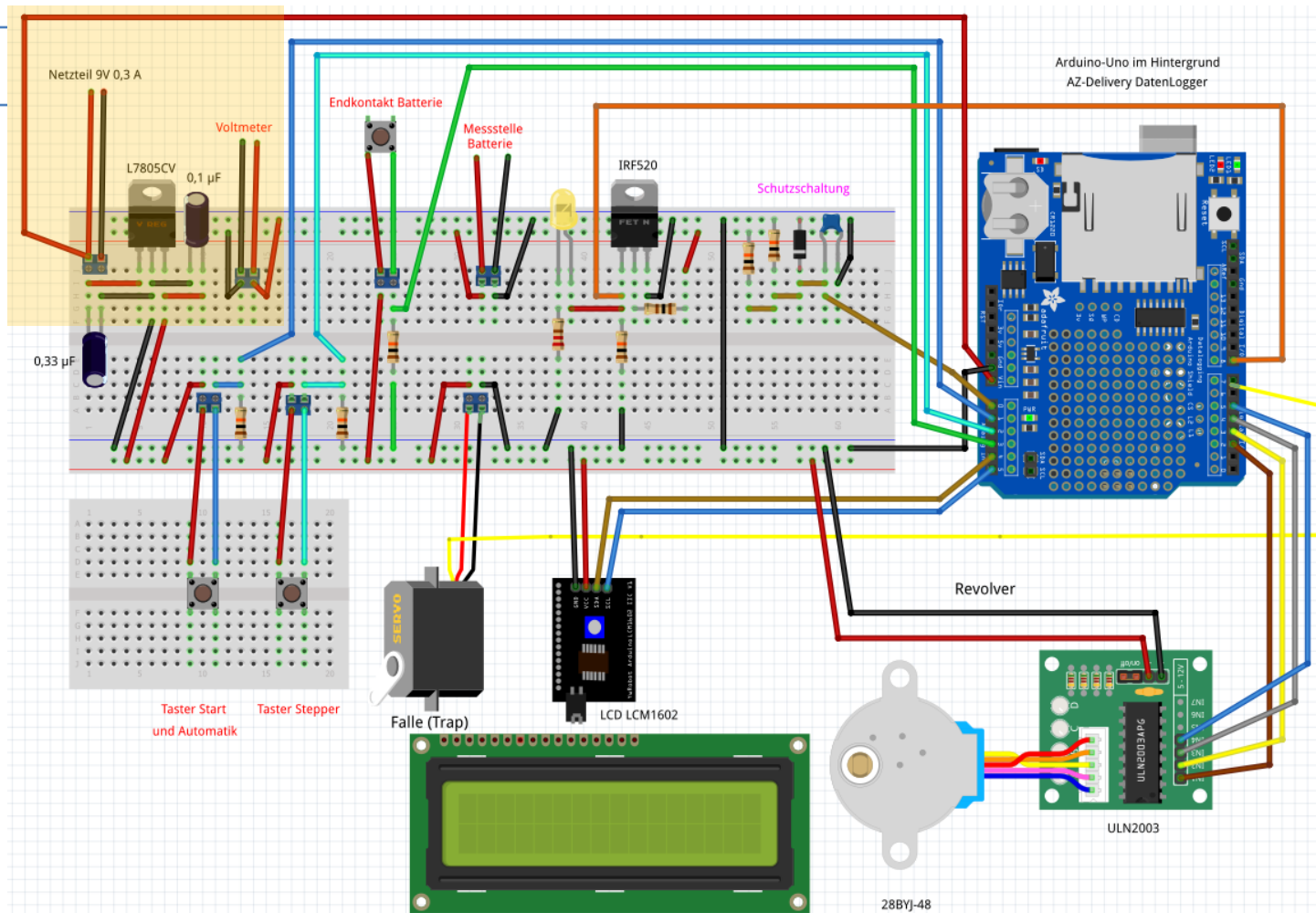


| | |
|----------------------|--|
| | SD-Karte |
| LCD | LCD 16 x 2 mit LCM1602 |
| MOSFET | FET N-channel IRF520 |
| Diskrete Bauelemente | LED, Taster, Widerstände, Diode, Kondensator |

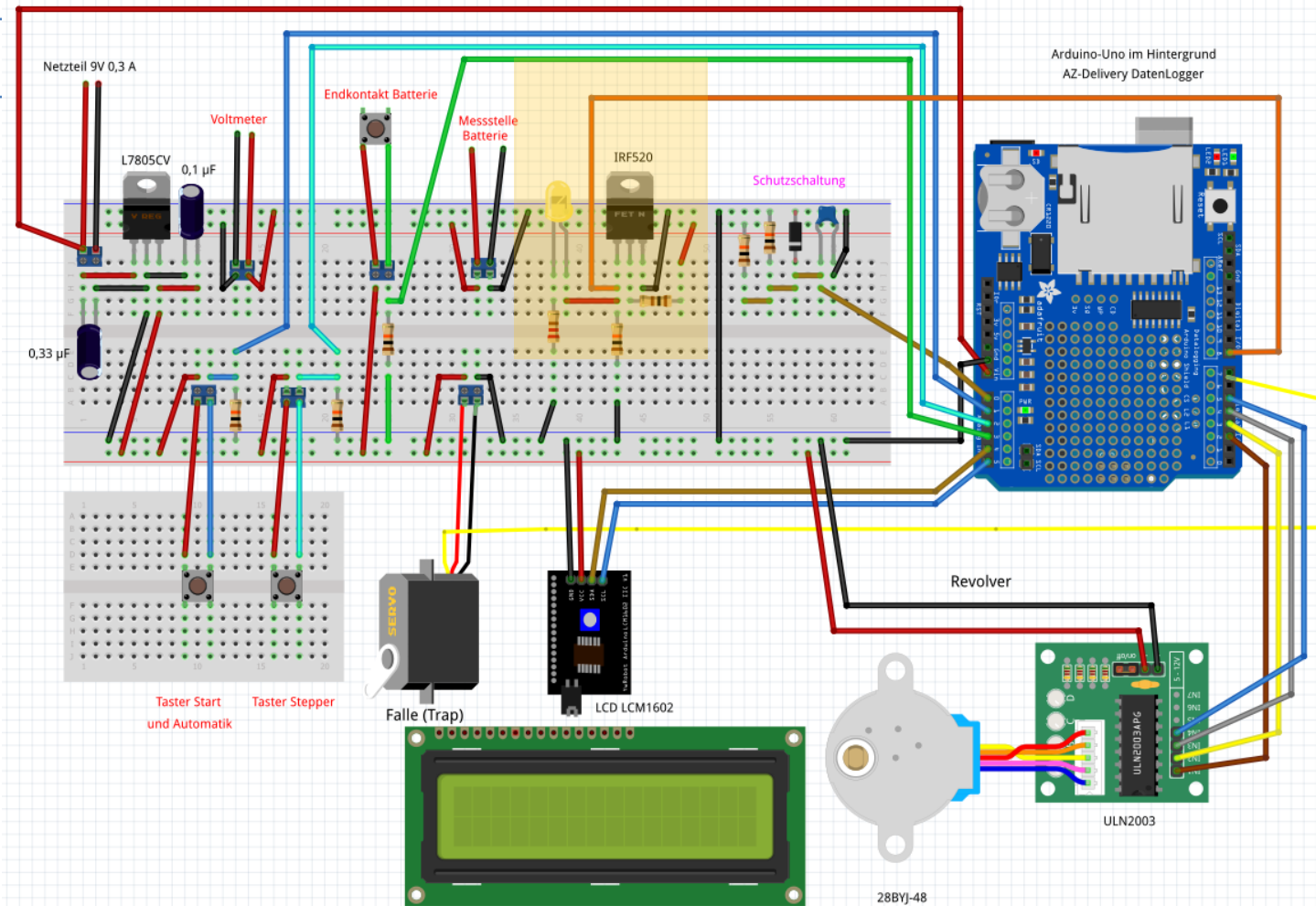
Schaltung



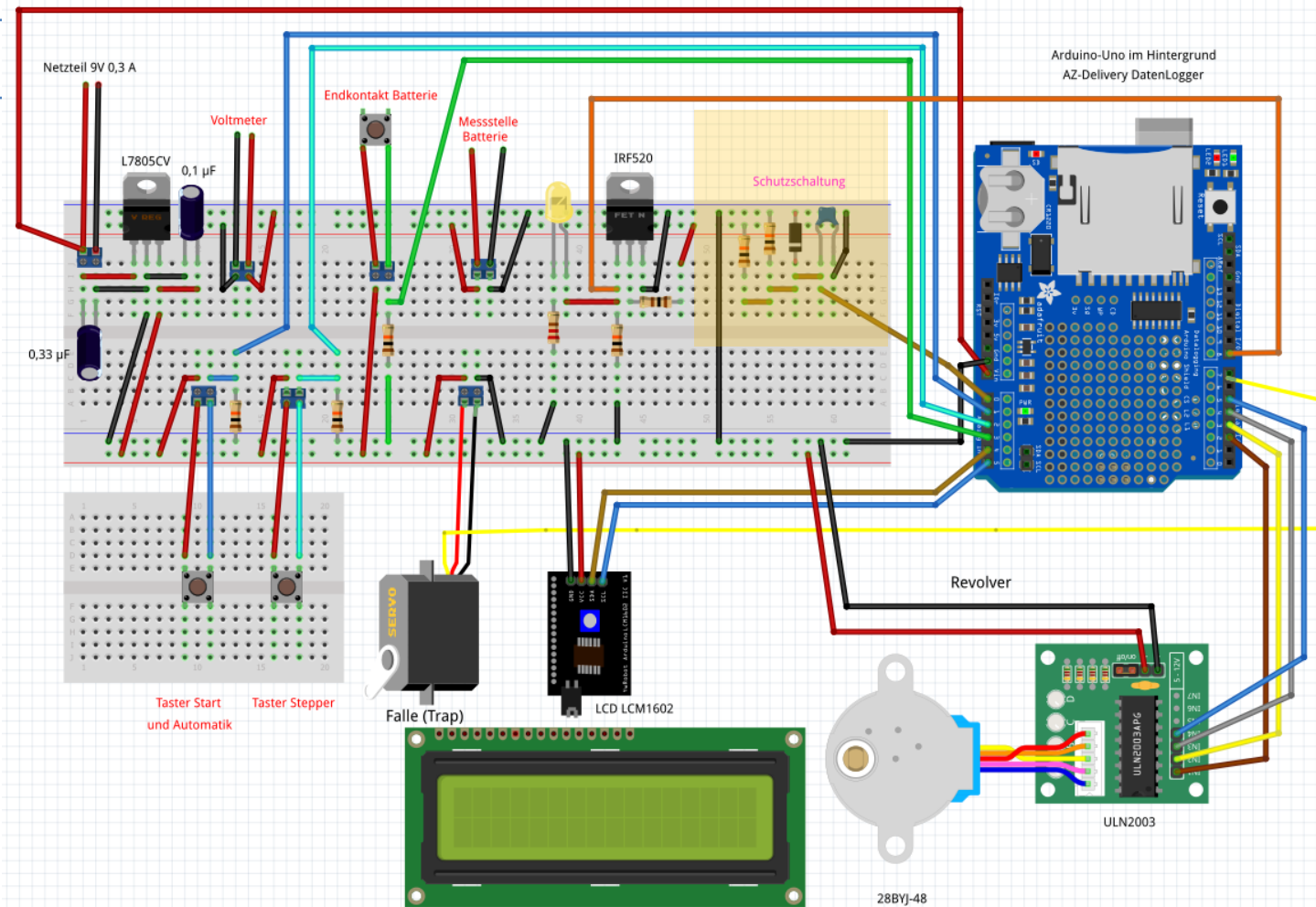
Schaltung



Schaltung



Schaltung



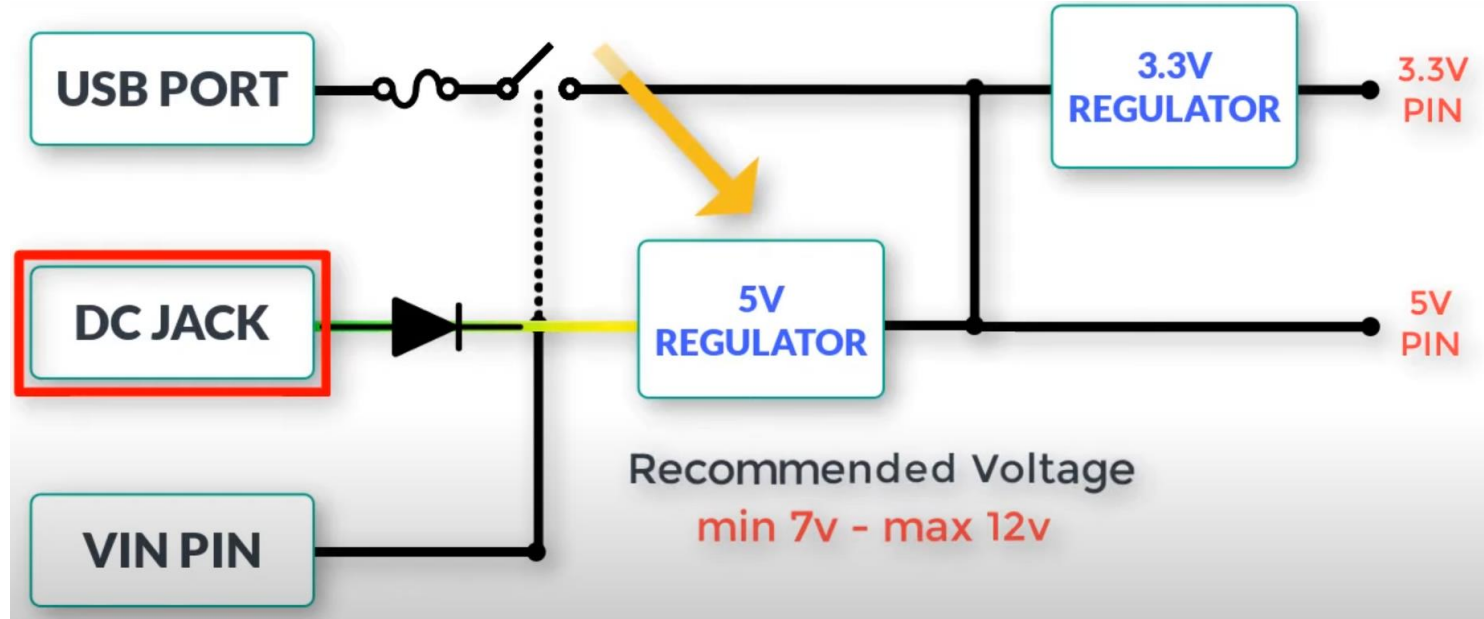
Arduino-Pinbelegung

| Arduino Pin | | Bauteil | Sketch |
|-------------|------|--|--|
| 13 | SCK | DatenLogger | |
| 12 | MISO | DatenLogger | |
| 11 | MOSI | DatenLogger | |
| 10 | SS | DatenLogger | |
| 9 | | frei | |
| 8 | | Lastkreis Ein/Aus MOSFET | stressPin |
| 7 | | Servomotor | servoPin |
| 6 | | frei | |
| 5, 4, 3, 2 | | Schrittmotor | motor_pin_4, motor_pin_2, motor_pin_3, motor_pin_1 |
| GND | | Allgemein GND | |
| VIN | | Externe Spannung | |
| A0 | | Spannungsmessung | voltagePin |
| A1 | | Taster Start | operationPin |
| A2 | | Taster Schrittmotor | stepPin |
| A3 | | Endschalter Batterie | stopAutoPin |
| A4 | SDA | LCD 2x16, DatenLogger I ² C | LCD SDA, DatenLogger SDA |
| A5 | SCL | LCD 2x16, DatenLogger I ² C | LCD SCL, DatenLogger SCL |

Eingangsspannung

Welchen Anschluss?

USB Port / DC Jack / VIN



Link

<https://www.youtube.com/watch?v=3rbn0pNoGa8>
<https://www.programmingelectronics.com/power-arduino/>

Stromversorgung

Arduino Stromversorgung

| | |
|---------------------|---|
| DC-Jack (DC-Buchse) | 7 V bis 12 V (mit Verpolungsschutz) empfohlen. |
| USB | 5 V (meist kleiner). |
| VIN | 7 V bis 12 V (ohne Verpolungsschutz). |
| 5 V Pin | Keinesfalls machen! |

Steckbrett Stromversorgung

| | |
|---------|--|
| 5 V Pin | 5 V (reguliert über den Spannungsregler) |
| VIN | 7 V bis 12 V (in Abhängigkeit von der DC-Buchse). Nicht empfohlen, da Sensoren und Aktoren meist 5 V benötigen. |
| Link | https://docs.arduino.cc/hardware/uno-rev3 |
| | How do I power my Arduino? The Pi Hut |

Stromstärken

Arduino Stromversorgung

| | |
|---------------------|------------------------------|
| USB | 500 mA (resettable polyfuse) |
| DC-Jack (DC-Buchse) | 500 mA bis 1 A |
| VIN | 500 mA bis 1 A |
| 5V Pin | Keinesfalls machen! |

Pins

| | |
|----------------------------------|---|
| pro Pin | 20 mA (absolute maximale Stromstärke 40 mA) |
| Σ digitale & analoge Pins | 200 mA (d. h. maximal 10 LEDs mit 20 mA) |

Verwendung als Spannungsquelle

| | |
|--------|---|
| 5V Pin | 500 mA (800 mA, abhängig vom Spannungsregler) |
| VIN | 1 A in Abhängigkeit von der DC-Buchse. Nicht empfohlen! |

Analoge Messwerte

| | |
|-----------------------|---|
| Anforderungen | Erfassen der Spannungen im Bereich < 2 V. |
| Randbedingungen | <ul style="list-style-type: none">Spannung einer neuen Batterie $\leq 1,6$ V.Testspannung von 1,0 V. Sketch: <code>compareVoltage = 1.0;</code>Arduino Referenzspannung 1,1 V. Sketch: <code>analogReference (INTERNAL)</code> |
| Spannungsteiler | Wegen der Referenzspannung von 1,1 V ist eine Spannungsteilerschaltung erforderlich. |
| AREF-Pin | Am AREF-Pin kann die tatsächliche Referenzspannung gemessen werden. Sketch: <code>AREF_Voltage = 1.07</code> |
| Analog-Digitalwandler | Umwandlung der Spannung (hier 0 bis 1,1 V) in die digitalen Werte von 0 bis 1023. |
| Berechnung im Sketch | <code>readAnalog = AREF_Voltage/1023 * (readAnalog) * 2;</code> |

Schutzschaltung

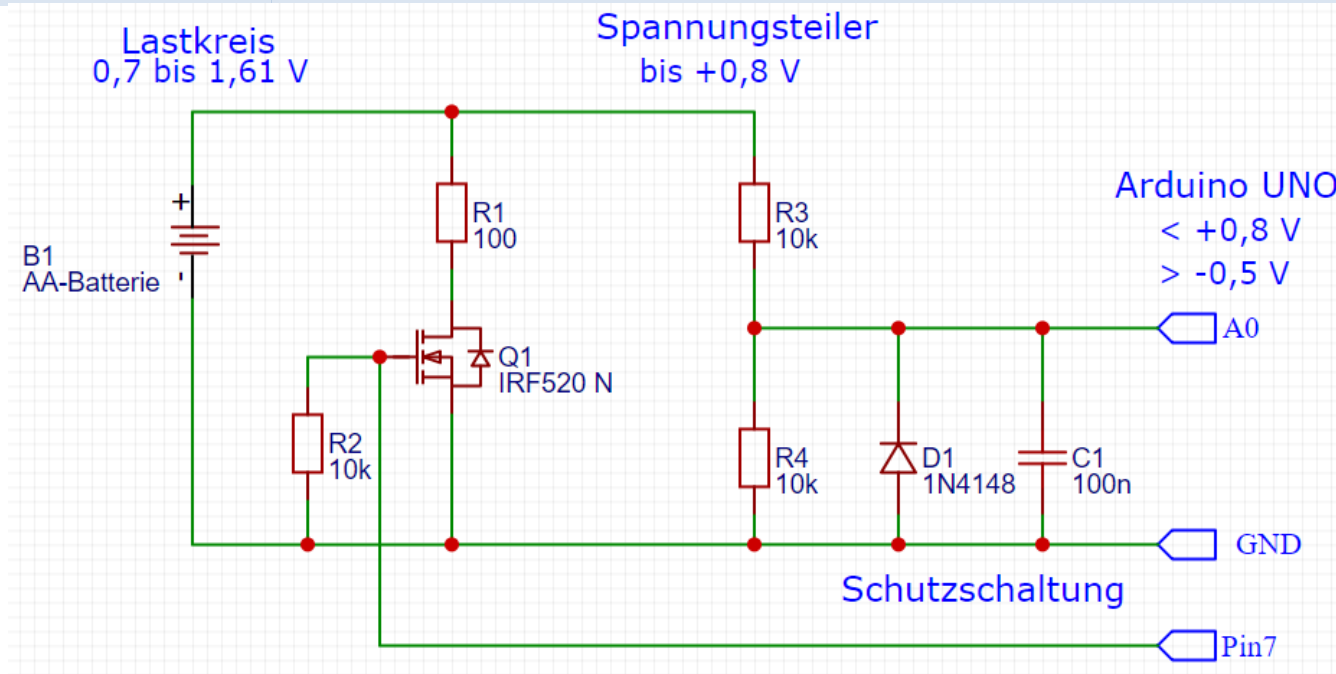
Problem

Batterie an der Messstelle verpolt!

Arduino-Datenblatt

An den analogen Eingängen dürfen Spannungen nicht $< -0,5\text{ V}$ sein.

Schutzschaltung



Schutzschaltung

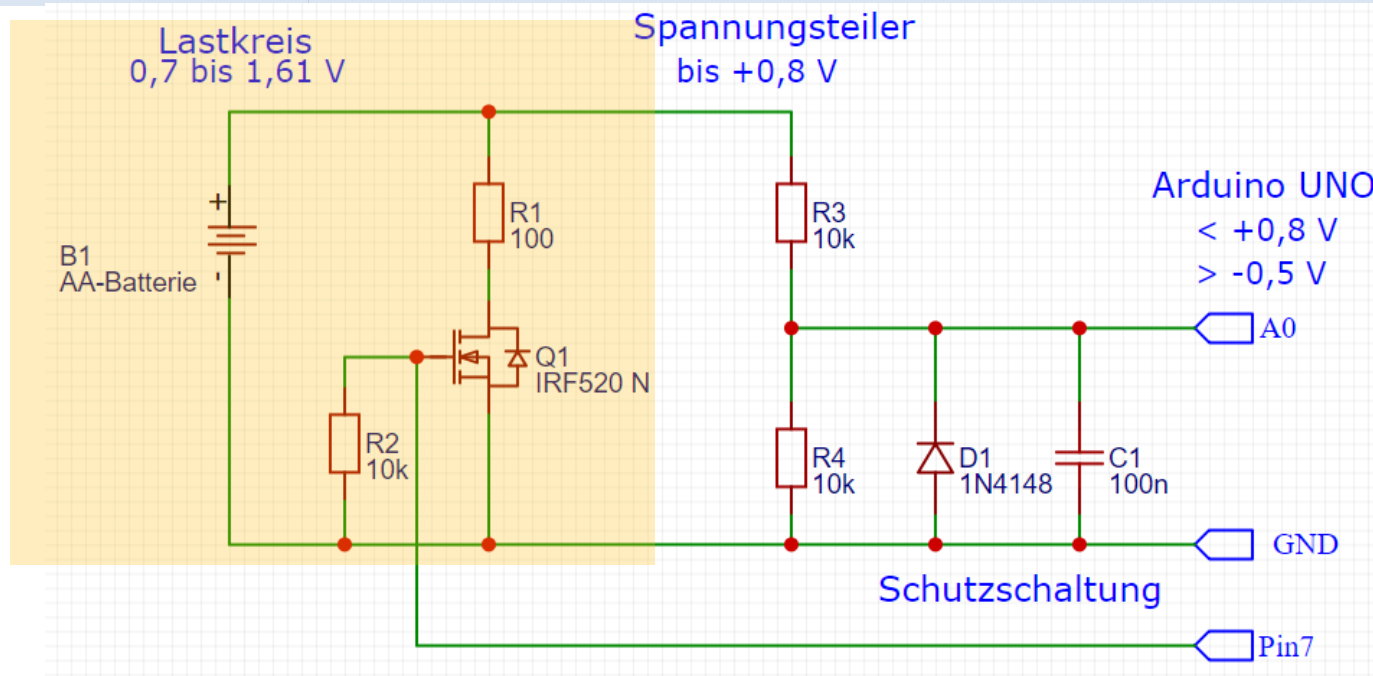
Problem

Batterie an der Messstelle verpolt!

Arduino-Datenblatt

An den analogen Eingängen dürfen Spannungen nicht $< -0,5\text{ V}$ sein.

Schutzschaltung



Schutzschaltung

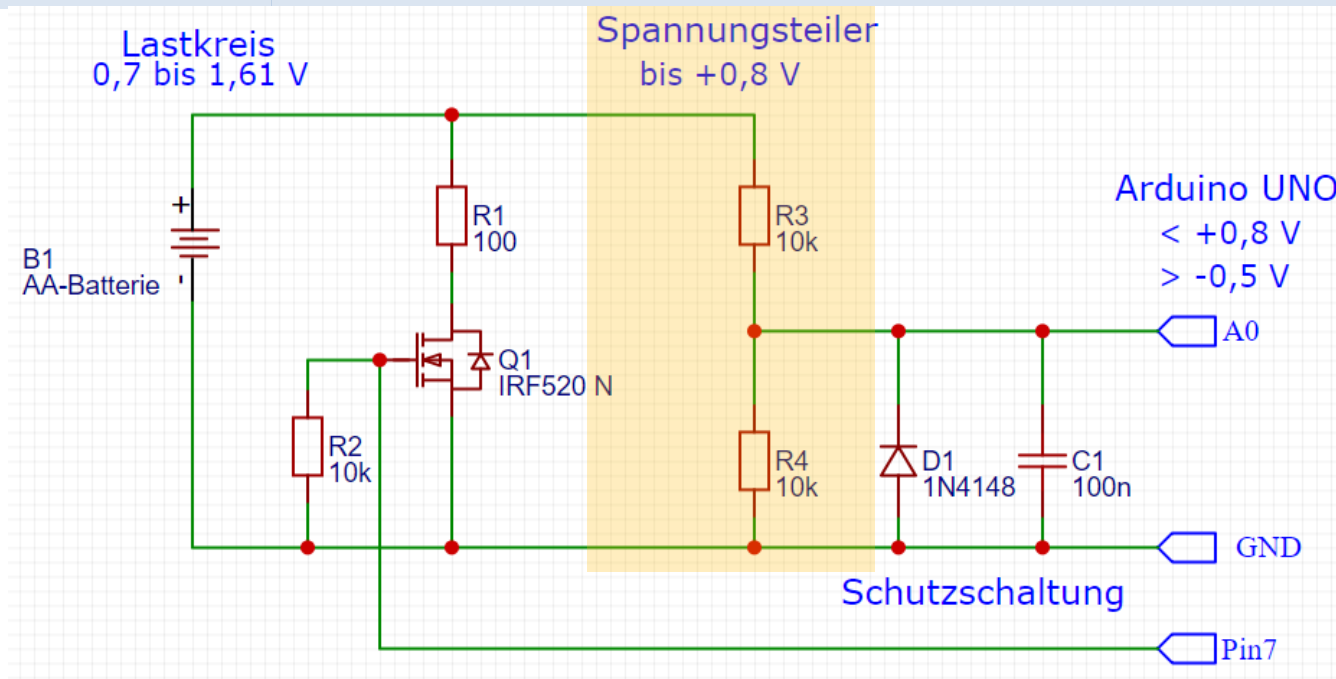
Problem

Batterie an der Messstelle verpolt!

Arduino-Datenblatt

An den analogen Eingängen dürfen Spannungen nicht $< -0,5\text{ V}$ sein.

Schutzschaltung



Schutzschaltung

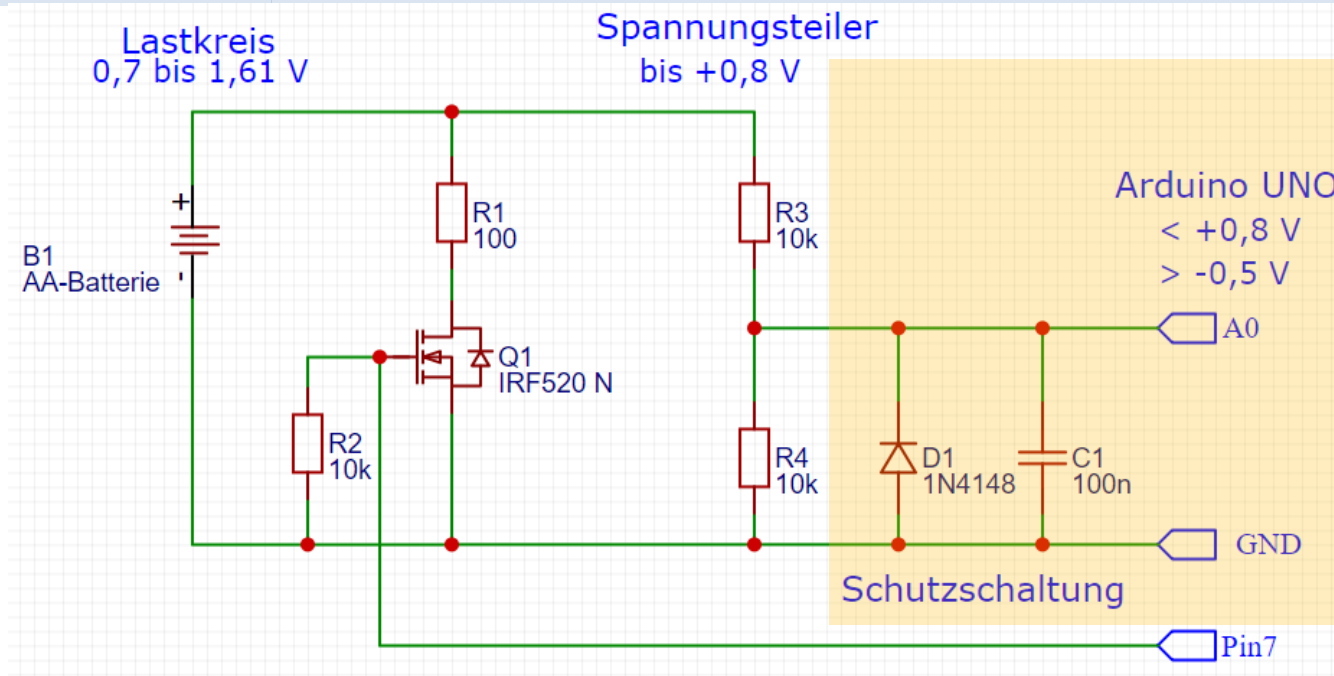
Problem

Batterie an der Messstelle verpolt!

Arduino-Datenblatt

An den analogen Eingängen dürfen Spannungen nicht $< -0,5\text{ V}$ sein.

Schutzschaltung



Programmierung

| Sketch | Tab | Funktion |
|--------------------------|----------------------|---|
| AAA_Tester_Automatik.ino | AAA_Tester_Automatik | Deklaration |
| 1_Setup.ino | 1_Setup | void setup() |
| 2_Loop.ino | 2_Loop | void loop() |
| 3_Help.ino | 3_Help | void help() |
| 4_Terminal.ino | 4_Terminal | void terminalInput() |
| 5_Automatic.ino | 5_Automatic | void automatic() |
| 6_Servo_Stepper.ino | 6_Servo_Stepper | void stepper_move() void servo_move() void servo_attach() void stepper_off() |
| 7_Stress.ino | 7_Stress | boolean battery_present() double check_voltage() void stress_circuit() |
| 8_Time.ino | 8_Time | String echo_time() |
| 9_SD_Card.ino | 9_SD_Card | void read_file() void write_file() |
| Echo.ino | Echo | void echo_config() void echo_trap() void echo_voltage() |
| Serial.ino | Serial | void serialEvent() void bufferToTokens() |

Funktionen I

| Tab | Aufgabe |
|----------------------|--|
| AAA_Tester_Automatik | Deklarieren der Variablen... Deklarieren der Pin-Belegung. Deklarieren der Parameter, die über „config.ini“ vom Anwender verändert werden können. Erzeugen der Objekte: rtc, myServo, stepper, lcd |
| 1_Setup | Initialisierung der Pins. Initialisieren der Objekte: lcd, stepper, rtc |
| 2_Loop | Fragt in einer Endlosschleife die Taster „Start“ und „Stepper“ ab. Das erstmalige Drücken der „Start-Taste“ führt allgemeinem Start und zum manuellem Terminal-Betrieb. Das nochmalige Drücken der „Start-Taste“ führt zum Automatik-Betrieb (ohne PC). Das Drücken der „Stepper-Taste“ lässt den Stepper jeweils 10 Schritte drehen, um die Positionierung des Revolvers zu ermöglichen. |
| 3_Help | Ausgabe der „Terminal-Befehle“ im Terminal/Monitor. |
| 4_Terminal | Auswerten der „Terminal-Befehle“ und veranlassen von Aktionen. |
| 5_Automatic | Steuerung des Automatik-Betriebes. Anwender-Hinweise auf dem LCD und dem Terminal/Monitor). |
| 6_Servo_Stepper | Steuerung von Servo- und Schrittmotor. Die Motoren werden stets (nach der Funktion) elektrisch entkoppelt. |
| 7_Stress | Prüfen ob Batterie vorhanden. Prüfen der Batterie-Spannung. Ein- und Ausschalten des Lastkreises. |

Funktionen II

| Tab | Aufgabe |
|-----------|--|
| 8_Time | Ermitteln von RTC-Datum und RTC-Uhrzeit. Zusammensetzen von Datum und Zeit für das SD-Karten Protokoll. |
| 9_SD_Card | Lesen der „config.ini“. Schreiben der „Excel.csv“. |
| Echo | Anzeigen der Konfigurations-Parameter (nur Terminal/Monitor). Anzeigen der Fallen-Parameter (nur Terminal/Monitor). Anzeigen der gemessenen Spannung. |
| Serial | Definieren eines Ereignisses (event) zur Entgegennahme von seriellen Eingaben vom Terminal/Monitor. Verarbeitung der Eingaben zu Steuervariablen: controlLetter, stringCommand, controlValue |

Libraries

| | |
|----------------------|--|
| Installationsordner | „C:\Users\Public\Arduino“ |
| Libraries-Ordner | “C:\Users\Public\Programme\arduino-1.8.19\libraries” |
| Download & Entpacken | U.g. Libraries herunterladen und entpacken. |

Arduino-LiquidCrystal-I2C-library:

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library/archive/refs/heads/master.zip>

RTCLib

<https://github.com/adafruit/RTCLib/archive/refs/heads/master.zip>

Innerhalb der Arduino-IDE, Werkzeuge, Bibliotheken verwalten
nach „Adafruit BusIO“ suchen und diese installieren (I2C).

SD-Karte

| | | | | | |
|----------------------------|---|-----------|-----------------------|---------------|------------|
| Konfigurationsdatei | Auf der SD-Karte eine Datei mit dem Namen „config.ini“ anlegen. | | | | |
| Automatik-Betrieb | Die Parameter in “config.ini” überschreiben im Automatik-Betrieb die voreingestellten Parameter vom Sketch. | | | | |
| Achtung: | Die Parameter werden nicht auf Plausibilität geprüft. | | | | |
| Parameter | Nur die Werte in „config.ini“ untereinander eingeben! | | | | |
| Testspannung in V | 1.10 | | (compareVoltage) | | |
| Belastungsdauer in s | 10 | | (referenceStressTime) | | |
| AREF (gemessen) in V | 1.1 | | (AREF_Voltage) | | |
| Ladepunkt-Messposition | 390 | | (stepsLoadTest) | | |
| Messposition-Falle | 350 | | (stepsTestTrap) | | |
| Protokolldatei | Die Datei „Excel.csv“ auf der SD-Karte anlegen. | | | | |
| | Der Sketch erzeugt folgendes Datenformat: | | | | |
| Datensatz | Datum | Uhrzeit | Leerlauf-Spannung | Last-Spannung | gut / leer |
| Beispiel | 28.09.2022; | 12:21:35; | 1,5; | 1,44; | 1 oder 0 |
| Hinweis | Im Sketch werden C++-Werte (1.22) in Excel-Werte konvertiert (1,22). | | | | |

Inbetriebnahme

| Checkliste | |
|------------|--|
| | <ul style="list-style-type: none">• Die Prüfanlage ist zusammengebaut.• Der Arduino & Steckbrett sind mit den externen Komponenten verbunden. |
| | <ul style="list-style-type: none">• Die Software ist auf den Mikrocontroller übertragen worden.• Die SD-Karte mit der „config.ini“ und der „Excel.csv“ ist vorbereitet und eingesteckt. |
| | <ul style="list-style-type: none">• Die Stromversorgung von Steckbrett & Arduino ist angeschaltet.• Der Arduino ist per USB-Kabel mit einem PC verbunden. |
| | <ul style="list-style-type: none">• Die Arduino-IDE und der Arduino-Monitor sind geöffnet. |

Anwender-Interface

| | |
|-------------------|---|
| Anleitung | „AAA-Tester Anleitung.pdf“ auf GitHub. |
| Automatik-Betrieb | Betrieb mit Taster & LCD |
| Test-Betrieb | Prüfanlage testen mit Monitor/Terminal. |

Monitor/Terminal-Befehle

| Befehl | | Bedeutung |
|----------------------|---------------------|---|
| Automatik | 'a' or 'auto' | Einschalten des Automatik-Betriebes. |
| Zeit? | 'time' | Anzeigen von Datum und Uhrzeit. |
| Lese config.ini | 'r' | Einlesen der „config.ini“ von der SD-Karte. |
| Zeige config.ini | 'i' | Anzeigen der aktuellen Parameter. |
| Falle initialisieren | 'ti' | Falle auf Position gute Batterie drehen. |
| Batterie vorhanden? | 'b' | Befindet sich eine Batterie an der Messposition. |
| Anliegende Spannung | 'v' | Messen und Ausgeben der Batteriespannung. |
| Last-Kreis an/aus | 'l' or 'load' | Lastkreis ein- und ausschalten. |
| Syntax für Falle | 't:90' or 'trap' | Drehen der Falle um Werte zwischen 0 bis 90 Grad. |
| Syntax für Stepper | 's:90' or 'stepper' | Drehen des Schrittmotors um sinnvolle Werte, ½ Drehung mit Wert 1024. |
| Hilfe-Text | 'h' | Liste der Monitor/Terminal-Befehle |

Prüfanlage Funktionen

| | |
|----------------------|---|
| | 1. Prüfvorrichtung anpassen |
| Ablauf einer Messung | 2. Batterie laden (schiefe Ebene). 3. Ladepunkt -> Messposition |
| Messen | 4. Batterie vorhanden? 5. Falls keine Batterie vorhanden Automatik anhalten. 6. Leerlauf-Spannung messen. 7. Lastkreis schließen. 8. Last-Spannung messen. 9. Lastkreis öffnen. 10. Gute oder leere Batterie? |
| Sortieren | 11. Messposition → Falle 12. Gute Batterie in Behälter „Gute Batterie“. 13. Leere Batterie in Behälter „Leere Batterie“. |
| | 14. Falle → Ladepunkt |

Ergebnisse & Schlusswort

| | |
|-------------|---|
| Ergebnisse | <ul style="list-style-type: none">• Selbstständig brauchbare und leere Batterien unterscheiden.• Einsatz in Schulen und Jugendgruppen. |
| | <ul style="list-style-type: none">• Ressourcenschonung• Statistische Auswertungen. |
| | |
| Schlusswort | <ul style="list-style-type: none">• Wettbewerb Jugend forscht. |
| | <ul style="list-style-type: none">• Maker Faire• Schulprojekte |
