

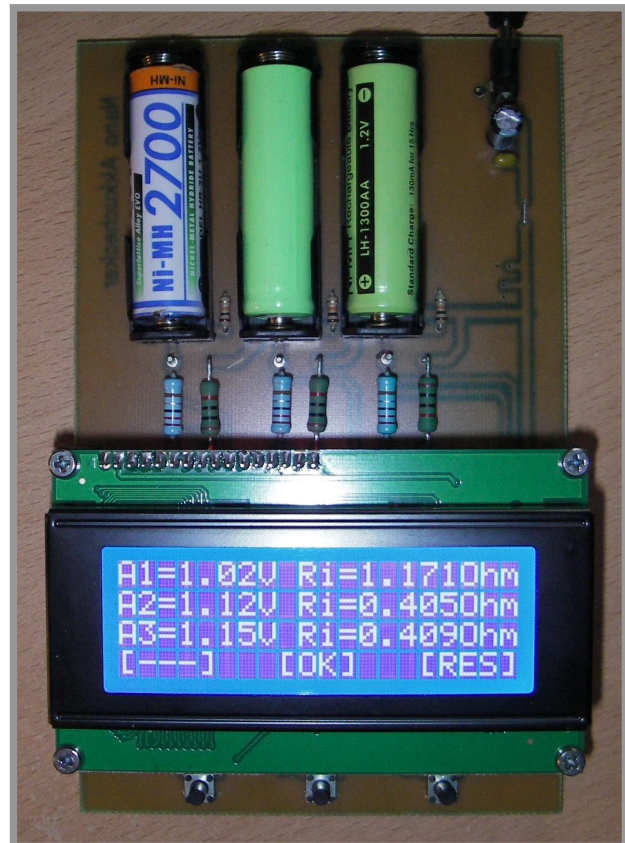
Akku-Checker3 mit dem Arduino-Nano

Intension war zunächst ein Langzeit-Entlade-Kapazitätstest für NiMH- und NiCd-Akkus.

Die ergänzende Ladeprozedur ergab sich wegen der vorhandenen Hardware und freier Arduino-Ressourcen.

Damit wurde die ursprüngliche Idee eines reinen Entladers zu einem vollwertigen Checker erweitert.

vy 73 de
DL6NBS / Bernd - Sand
dl6nbs@dar.de

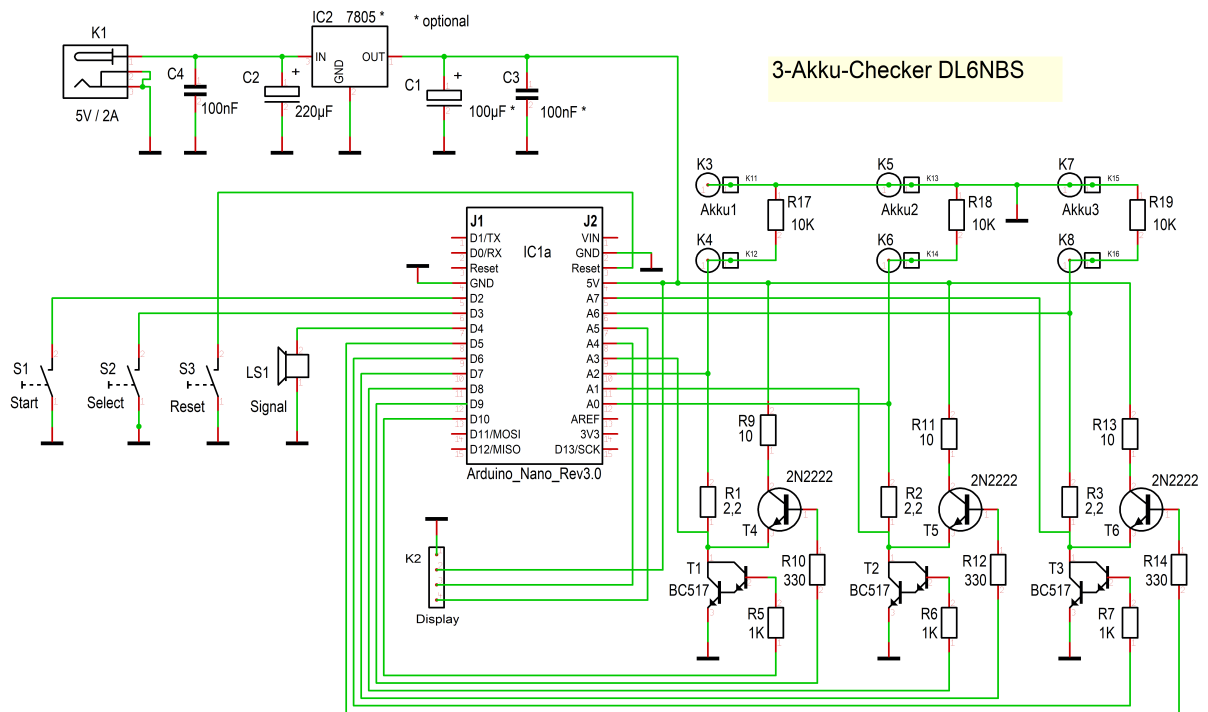


Merkmale

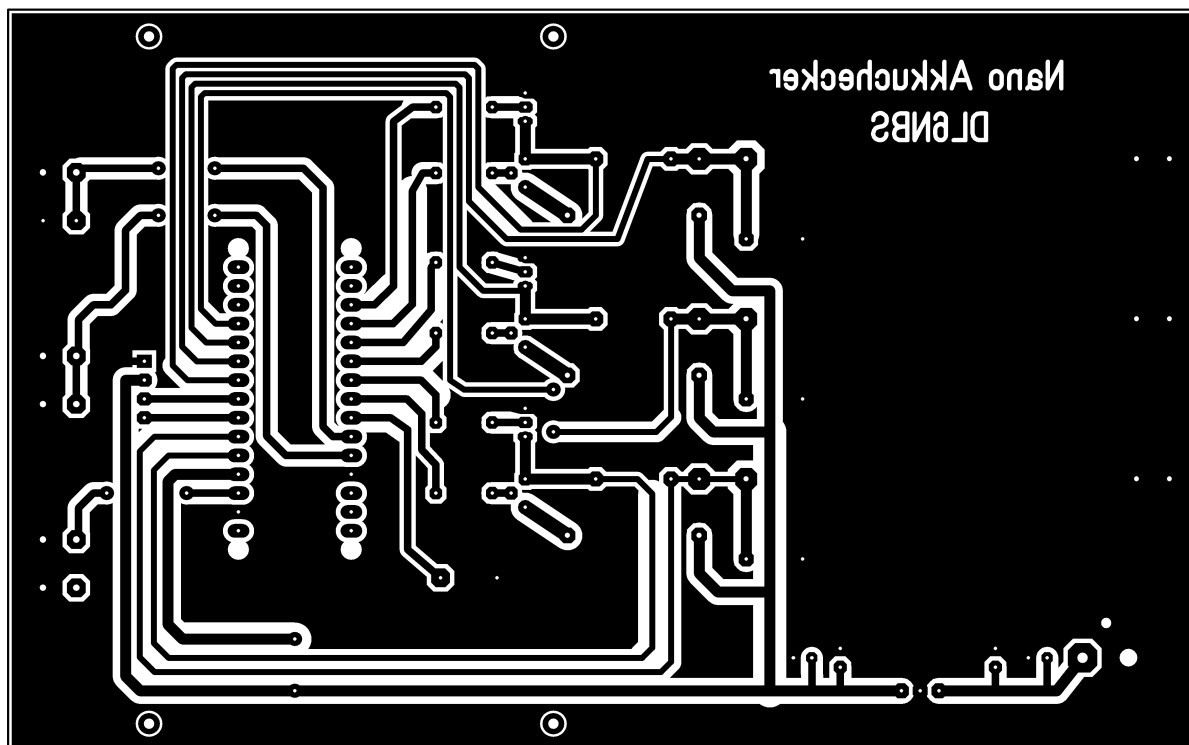
- Gleichzeitiger Test von drei NiMH-Akkus oder NiCd-Akkus.
- Über Analogeingänge wird jeder Akku einzeln analysiert.
- Anzeige über ein 4-Zeilen / 20-Zeichen - LCD-Display.
- Es sind Fünf Modi wählbar:
 1. nur Entladen
 2. nur Aufladen
 3. erst Entladen dann Aufladen
 4. erst Aufladen dann Entladen
 5. erst Entladen dann Aufladen dann wieder Entladen
- Ladeabschaltung über Endspannung, über Delta-U oder maximale Ladezeit.
- Serielle Ausgabe des Protokolls über den Arduino Serial Monitor ist möglich.
- Serieller Plot der Spannungsverläufe mit dem Arduino Serial Plotter ist möglich.

Aufbau

Der Schaltplan



Falls Du Deine Platine fertigen möchtest, hier das Layout der 100 x 160mm Platine.



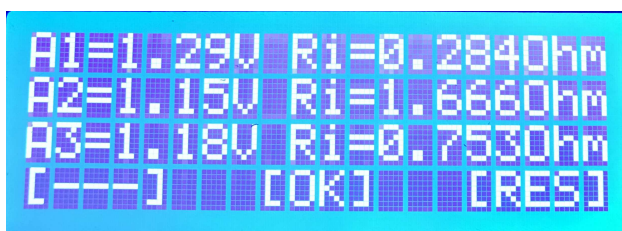
Beschreibung

Wir beginnen mit einem praktischen Anwendungsfall:

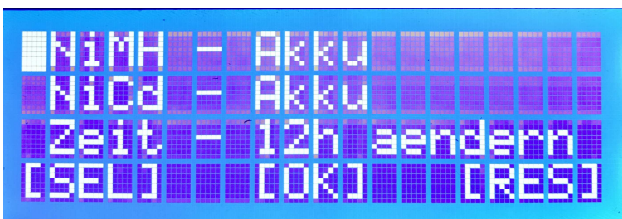
Du möchtest wissen, wie der Wirkungsgrad Deiner Akkus aussieht. Dazu werden sie erst mal aufgeladen und danach wieder entladen um zu ermitteln wieviel % von dem was reinging wieder herauskommt.

Gedanklich gehen wir nun durch die Sequenz "Aufladen -> Entladen". Danach hast Du die Funktionsweise des Checkers verstanden. Alles Weitere ist selbsterklärend.

Nach dem Einschalten meldet sich das Programm mit seiner Kennung und führt während fünf Sekunden eine Prüfung durch. Es wird getestet ob Akkus eingelegt sind und der Innenwiderstand (Ri) berechnet. Akkuspannung und Ri werden angezeigt.



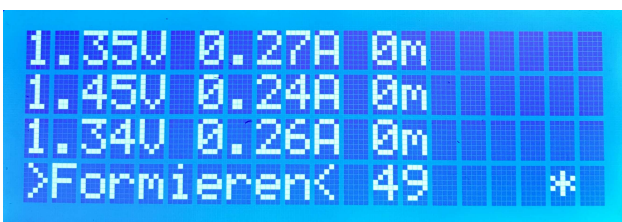
In der Statuszeile, unten, weist die Symbolik [---] [OK] [RES] auf aktive Taster hin. Die Taster schalten beim Loslassen. [RES] führt immer zu einem Neustart. Mit [OK] gelangst Du in das nächste Menü.



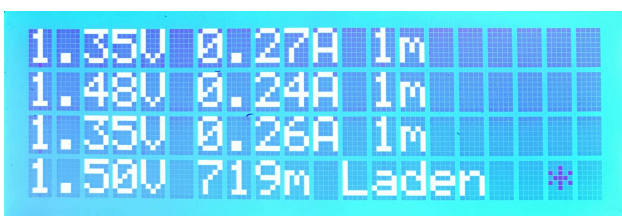
Mit [SEL] wird im Menü geblättert. Mit [OK] wird die Auswahl bestätigt.



Es wird einer der fünf Modi mit [SEL] gewählt. Hinter dem + im Cursor verbergen sich noch 2 weitere Modi die mit [SEL] erreichbar sind. Wir wollen am Beispiel *Aufladen -> Entladen* verstehen was weiter passiert.

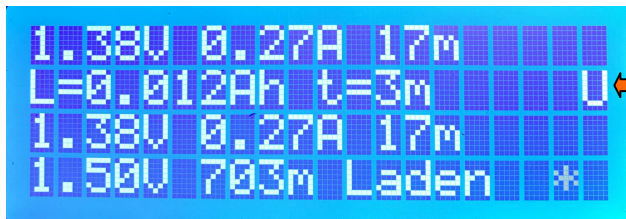


Während der ersten Minute werden die Akkus formiert, sie bekommen erst mal Saft. Die Sekunden in der Statuszeile zählen rückwärts. Ein * signalisiert, dass noch ein Sequenzteil folgt. Zwei ** würden 2 weitere Sequenzteile bedeuten.



Nach dem Formieren wird unter Kontrollbedingungen weiter geladen. In jeder Akkuzelle wird die Spannung, der Ladestrom und die Lademinuten angezeigt.

In der Statuszeile unten wird links die Ladeendspannung angezeigt, daneben die verbleibenden Lademinuten. Eine weitere Kontrolle, der Delta-U Ladebuckel, wird vom Programm durchgeführt nachdem 95% der Ladeendspannung erreicht sind. Das Display schaltet nach 10 Minuten ab, mit der [OK]-Taste wird es wieder hell.



Akku2 ist fertig, es ertönt ein Signal. Er hat während $t=3$ Minuten $L=0.012\text{Ah}$ aufgenommen und erreichte dann die Ladeendespannung 1.50V , angezeigt durch U rechts. Akku1 und 3 laden noch weiter.



Nach dem Laden aller Akkus wurde auf "Entladen" geschaltet. Die Anzeigen von Akku1 und Akku2 wechseln im Sekunden-takt. Akku1 bekam $L=0.588\text{Ah}$ Ladung und schaltete über Delta-U ab (D rechts).

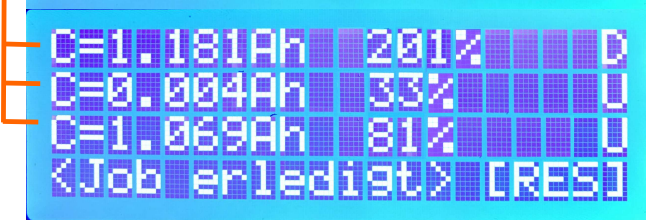


Die Bilanz von Akku1 201% ist nicht plausibel, er war zu Beginn der Sequenz nicht entladen! Modus-5 bewahrt vor diesem Fehler da vorher immer erst entladen wird.

Formel: $\text{Performanz} = C/L * 100 (\%)$



Die Sequenz "Aufladen -> Entladen" ist nun komplett abgeschlossen. Der Job ist erledigt.

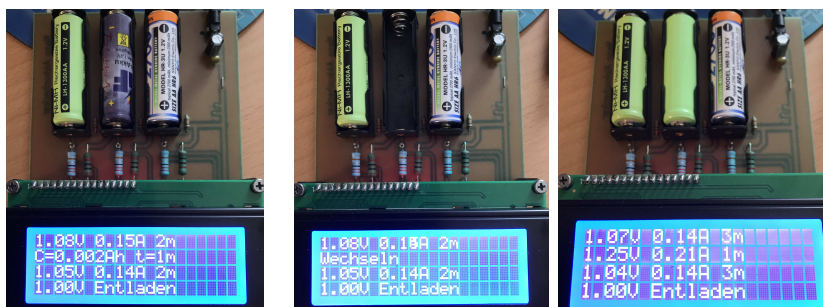


Keiner der Akkus überschreitet die maximale Ladezeit von 12h. In diesem Fall würde rechts Z für die maximale Ladezeit stehen.

Mit der [RES]-Taste wird alles beendet.

Fliegender Wechsel

Nach vorausgegangener Lektion ist das Funktionsprinzip klar, nun zum *fliegenden Wechsel*.



Falls mehrere Akkus im Test laufen und einer fertig wird, kann er ausgetauscht werden, ohne den Test der anderen Akkus zu unterbrechen.

Das ist praktisch, wenn mehrere Akkus nacheinander getestet werden sollen. Der fliegende Wechsel ist sowohl im Modus Entladen als auch Aufladen möglich.

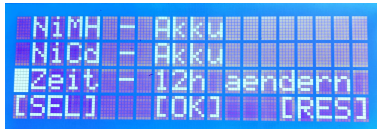
Auffrischung



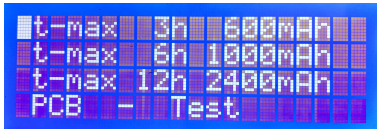
Sind alle Akkus geladen, wird jede Minute ein Refresh zur Ladeerhaltung durchgeführt.

Ladezeit ändern

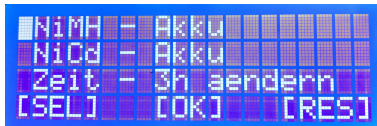
Hier erfährst Du, wie die maximale Ladezeit umgestellt wird.



Mit [SEL] *Zeit - 12h ändern* auswählen und [OK] betätigen.

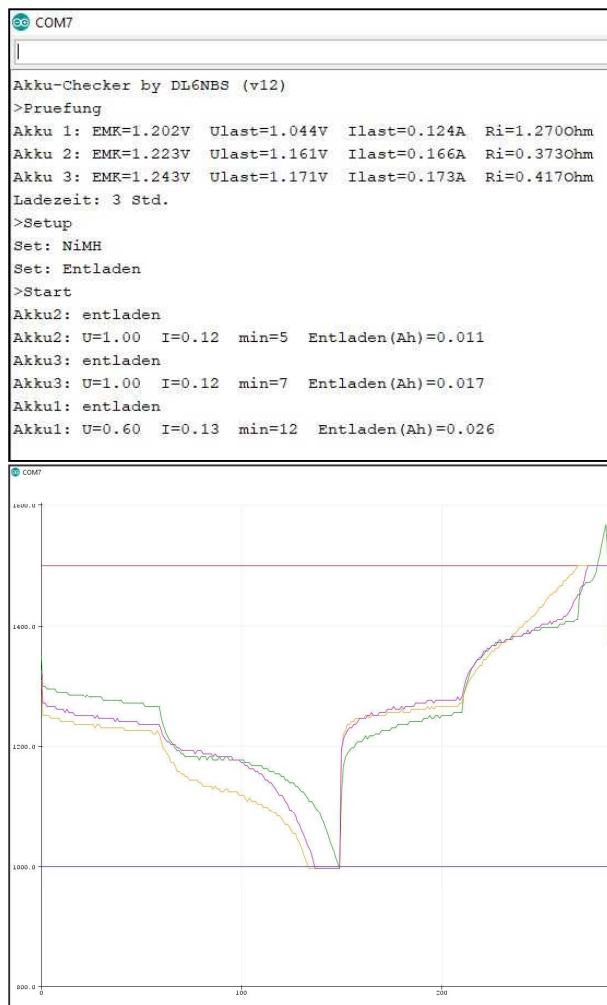


Mit [SEL] neue *t-max 3h* auswählen und [OK] betätigen.



Die geänderte Zeit "*Zeit - 3h*" wurde übernommen und im EEPROM des Arduino als neuer Standard abgelegt.

Serielle Ausgabe am Arduino



Protokoll-Infos werden vom Arduino auch seriell mit 19.200 Baud ausgegeben.

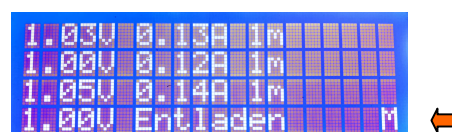
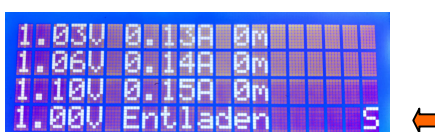
Wenn der Arduino-Serielle-Monitor aktiv ist, kann das Ereignisprotokoll am Bildschirm verfolgt werden.

Alternativ können mit dem Arduino-Serial-Plotter die Spannungsverläufe auch grafisch verfolgt werden.

Der Plotmodus kann im Sketch fest eingestellt werden (*int Plotter = 1*) oder mit der [SEL]-Taste zugeschaltet werden.

Der Plot erfolgt während der ersten Minute sekundlich, danach minütlich, S oder M wird rechts in der Statuszeile angezeigt.

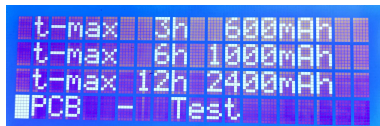
Mit [SEL] kann erneut eine Minute lang auf Sekundenplot geschaltet werden.



PCB-Test und Erst-Kalibrierung

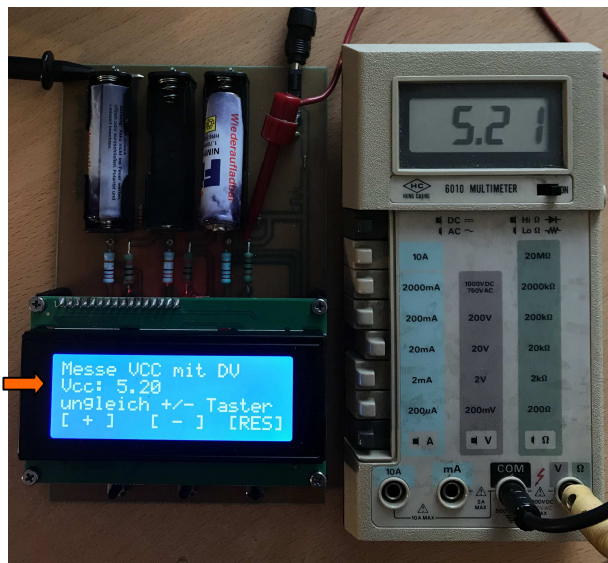
Falls der Arduino noch Jungfrau ist, wird bei erster Inbetriebnahme der Hinweis *! PCB-Test machen !* angezeigt.

Für genaue Messergebnisse muss die Höhe und Qualität der Versorgungsspannung einbezogen werden. Schaltnetzteile liefern häufig nicht exakt die geforderten 5V oder schwanken, das Programm korrigiert dies. Bei Inbetriebnahme ist deshalb einmalig der Parameter "VccFein" zu justieren. Der PCB-Test führt durch diese Kalibrierungsroutine.



Zum PCB-Test gelangst Du über das Ladezeit-Menü mit [SEL] und [OK].

Im ersten Schritt wird die Betriebsspannung +5V / Vcc des Arduino justiert.



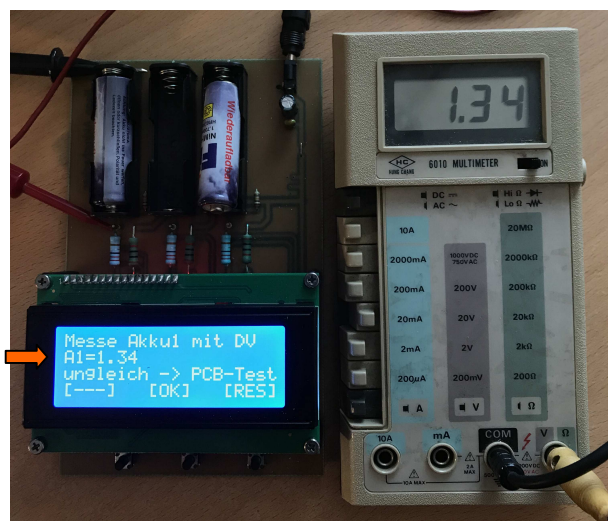
Vom Programm wurde Vcc mit 5.20V ermittelt.

Ein Digitalvoltmeter (DV) hängt an der Spannungsversorgung des Arduino und bestätigt mit 5.21 Volt den Arduino-Wert, ± 20 Millivolt sind ok.

Weichen die Ergebnisse ab, wird mit dem [+] oder [-] Taster korrigiert.

Erfolgt innerhalb von 10 Sekunden keine Eingabe mehr, wird der Korrekturwert ins EEPROM übernommen und der PCB-Test schaltet in Schritt 2.

Im zweiten Schritt wird die Spannungsmessung des Arduino nochmals am Akku verifiziert. Du legst einen Akku in Schacht1 ein.



Vom Programm wurde eine Spannung von 1.34V am Akku1 ermittelt.

Ein Digitalvoltmeter (DV) hängt am Akkuschaft1 und zeigt 1.34 Volt an. ± 10 Millivolt sind tolerierbar.

In diesem Zusammenhang: damit alle drei Messzweige identische Ergebnisse liefern, wird eine Vorselektion der 10 Ohm und 2,2 Ohm Widerstände sowie der Transistoren bezüglich HFE empfohlen.

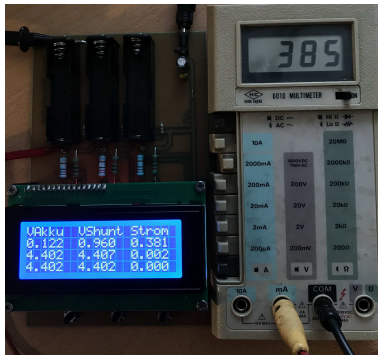
Im dritten Schritt werden die Ströme überprüft.

Du nimmst Akku1 wieder heraus und schaltest mit [OK] in den dritten Schritt.

```
Messe Stroeme mit DU
Pruefe Schacht 1,2,3
Soll: 360mA - 390mA
[---] [---] [RES]
```

Zunächst erscheint für einige Sekunden die Anleitung zum weiteren Vorgehen: *nacheinander die drei Ströme messen.*

In den Akkuschacht wird ein Amperemeter eingeschleift und der angezeigte mit dem gemessenen Strom verglichen



```
U Akku  Ushunt  Strom
0.122  0.960  0.378
4.426  4.426  0.000
4.402  4.402  0.000
```

Alle drei Kanäle sollten im Strom eng beieinander liegen. ± 10 mA sind ok.

```
U Akku  Ushunt  Strom
4.367  4.367  0.000
0.125  0.946  0.373
4.328  4.328  0.000
```

Noch einmal: eine Vorselektion der 10 Ohm und 2,2 Ohm Widerstände sowie der Transistoren wird empfohlen.

```
U Akku  Ushunt  Strom
4.383  4.383  0.000
4.388  4.388  0.000
0.126  0.950  0.375
```

Die Prüfung und Kalibrierung der Schaltung mit dem zugehörigen Arduino ist nun beendet. Eine erneute Prüfung und ggf. VccFein - Korrektur wird nur beim Austausch des Arduino nötig, die interne Referenzspannung kann differieren. Ein Netzteilwechsel hat keinen Einfluss auf die Qualität der Spannungsmessung, wohl aber auf den effektiven Ladestrom, umso härter und höher die Spannungsquelle, desto höher der Ladestrom.

Aufbau Tipps

Ich empfehle die Schaltung direkt über ein Schaltnetzteil mit 5V / 2A oder besser 3A zu versorgen. Der Spannungsregler 7805 auf der Platine kann dann entfallen, stattdessen wird eine Drahtbrücke eingesetzt. Bei Nutzung des Reglers auf der Platine ist für ausreichende Kühlung zu sorgen. Das Netzteil sollte in diesem Fall 7,5V bis 8,5V / 2A liefern. Höhere Spannungen heizen den 7805 im Ladebetrieb stark auf!

Programmierung

Auf Eleganz wurde verzichtet. Nachvollziehbarkeit was/wo/wie/wann passiert war mir wichtiger. Die Kommentierungen im Programm sollten reichen, um den Ablauf zu verstehen.

Noch ein Wort zum Thema "Akkus richtig laden"

Schau mal im Netz, Du wirst Dich über die unterschiedlichsten Sichten wundern!

Bei den "*Verstellbaren Parametern*" findest Du vier mit "*" gekennzeichnete Stellschrauben zur optimalen Abstimmung des Akku-Checkers für Deine Bedürfnisse. Die DeltaMV Empfindlichkeit ist naturgemäß etwas tricky. Bei zu frühem Auslösen sollte der Wert sehr behutsam nach oben verändert werden. Ich habe mit den aktuellen Werten bei Akkus unterschiedlicher Kapazität und Hersteller gute Ergebnisse erzielt.