

# Bluetooth Messwagen Aufbau und Inbetriebnahme

Eigentlich wollte nur mal schnell den tollen Messwagen von Matthias (<https://www.stummiform.de/viewtopic.php?t=163889>) erweitern damit man damit die Spannung am SuperCap der vorausfahrenden Lok messen kann...

Neun Wochen Später ist der im folgenden beschriebene Messwagen fertig geworden...

Der Messwagen kann neben der Geschwindigkeit und der Wegstrecke noch einige weitere Daten messen und per Bluetooth den PC oder ein Handy übertragen.

- 3 Spannungen (Schiene, Externer und interner SuperCap)
- Steigung und Neigung der Schiene
- Verschieden Zähler der Zeiten und Strecken

Außerdem können die Werte auf einem OLED Display dargestellt werden.

Die Schaltung ist für den Einbau in einem Lux Schien Staubsauger gedacht. Dieser verfügt bereits über eine Spannungsversorgung und eine Hall Sensor zur Geschwindigkeits- und Wegstrecken Messung. Darum übernimmt die Schaltung auch die Ansteuerung des Stabsaugers welcher per DCC in 4 verschiedenen Stärken angesteuert werden kann. Die Maximale Stufe hat eine deutlich stärkere Saugleistung als der Original Wagen.

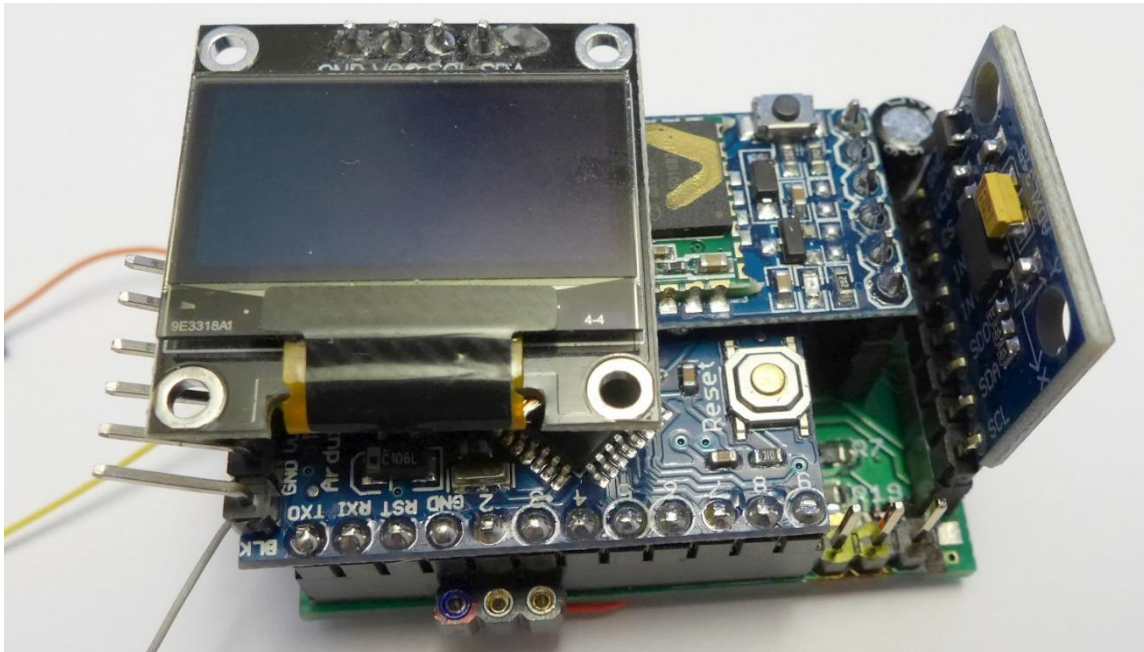
Neben dieser Funktion können noch einige weitere Befehle per DCC gesteuert werden.

Die Platine für dieses Projekt wurde von Matthias erstellt. Die Software basiert auf dem Projekt von Matthias und bubikopf. Sie wurde aber noch ein „bisschen“ erweitert.

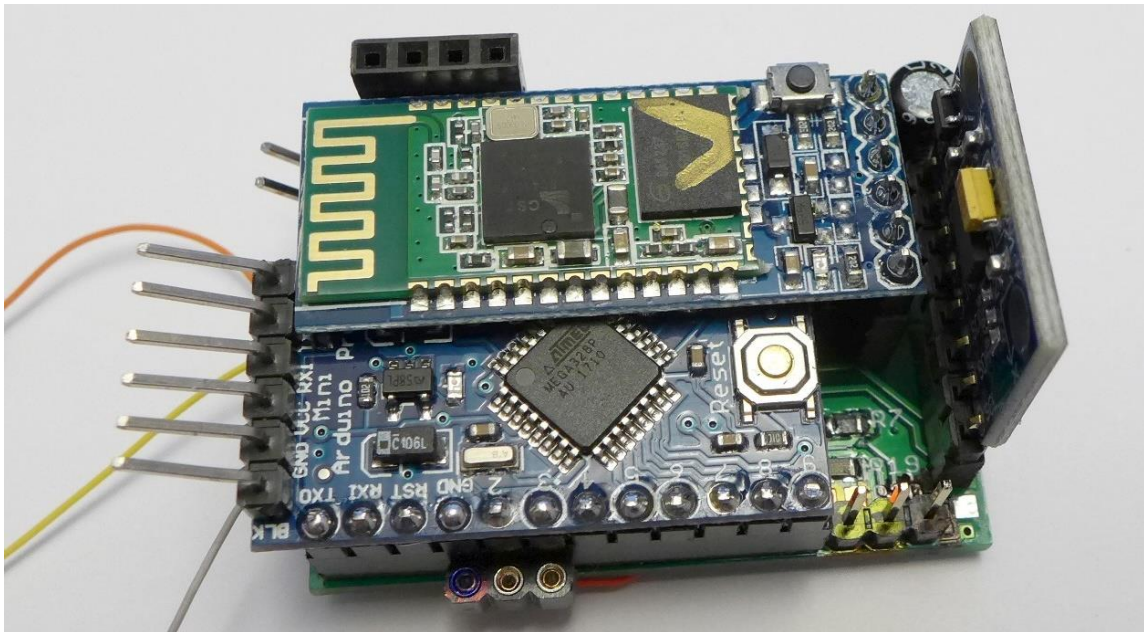
Dieses Dokument beschreibt kurz den Aufbau der Platine und erklärt wie das Bluetooth Modul in Betrieb genommen wird.

Zunächst ein paar Bilder

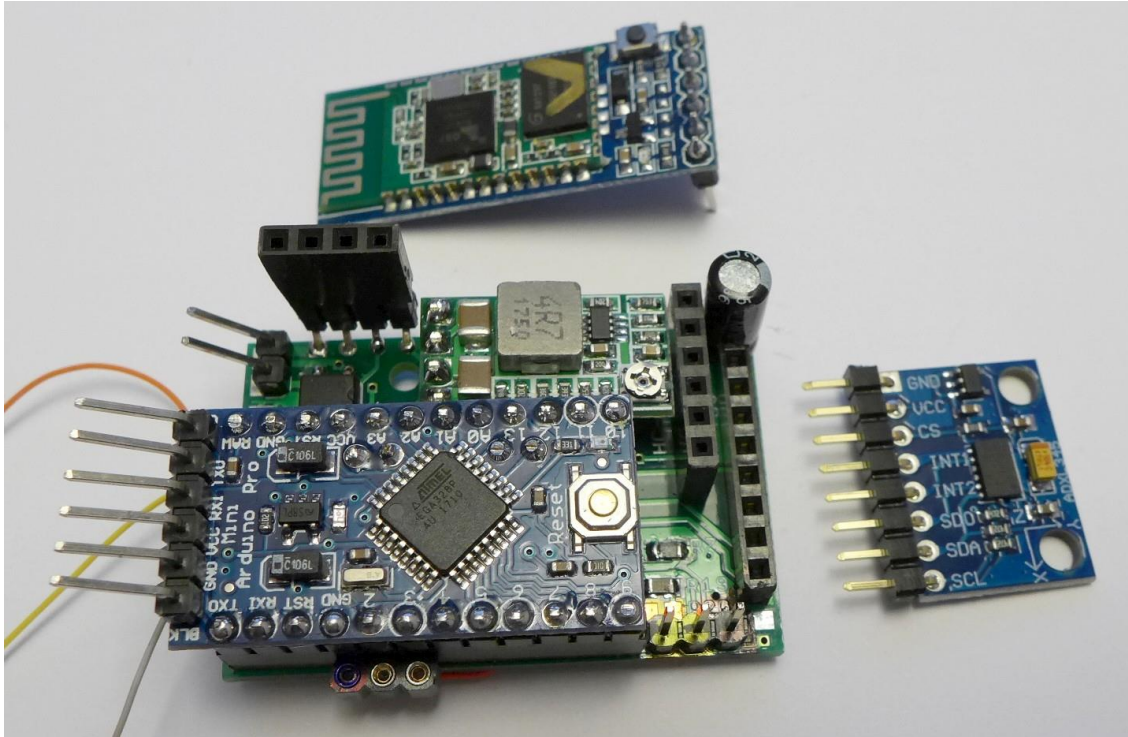
Alle Platinen:



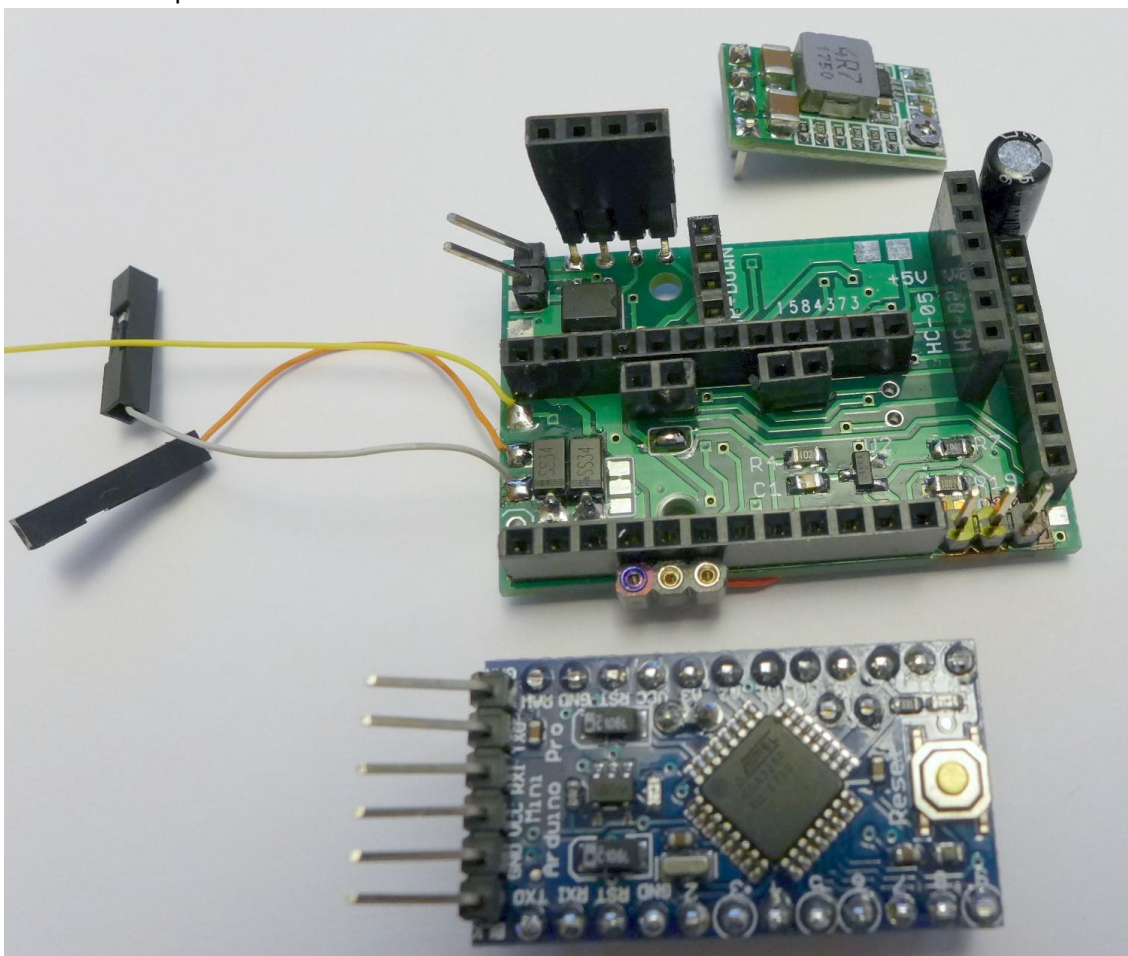
Mit abgestecktem Display:



Bluetooth Modul und Beschleunigungssensor entfernt:



Nur die Grundplatine:



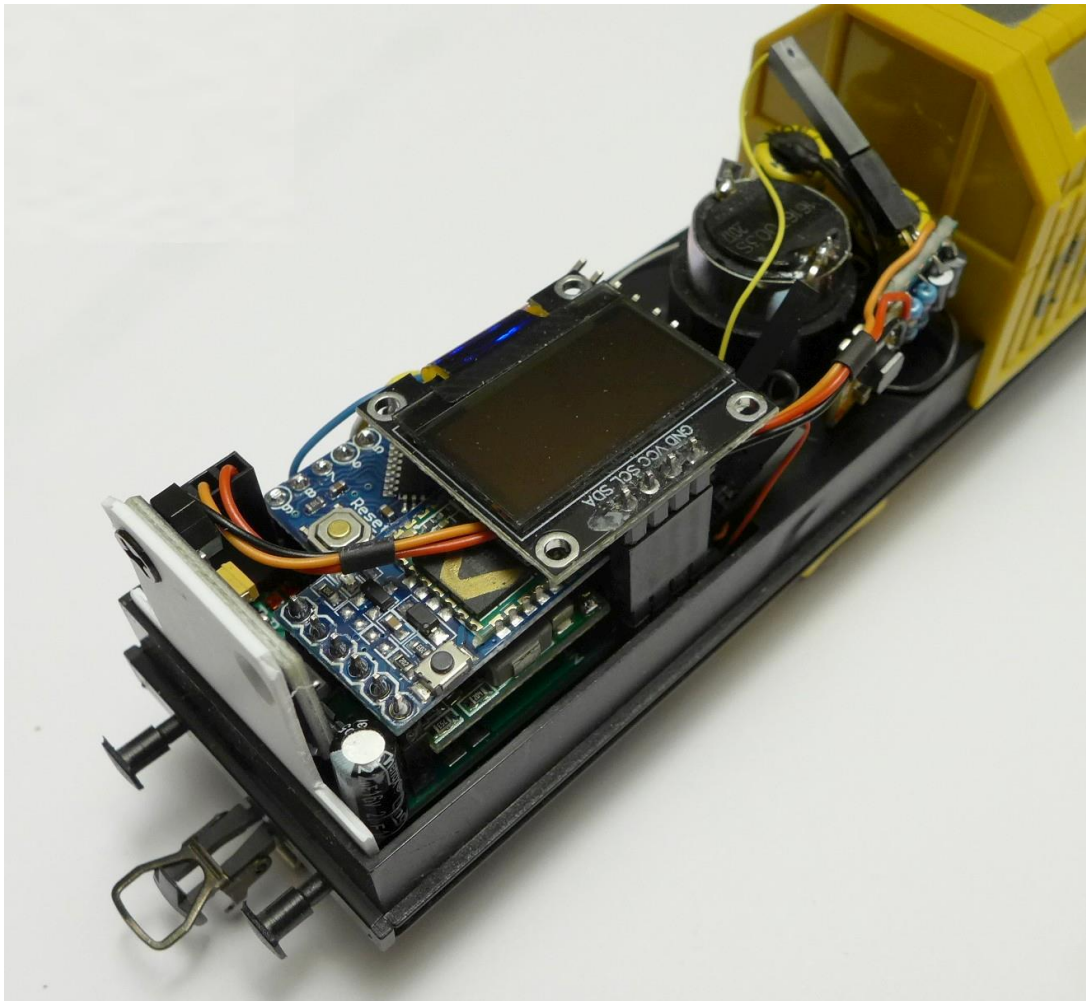


Grundplatine im Lux Staubsaugerwagen:

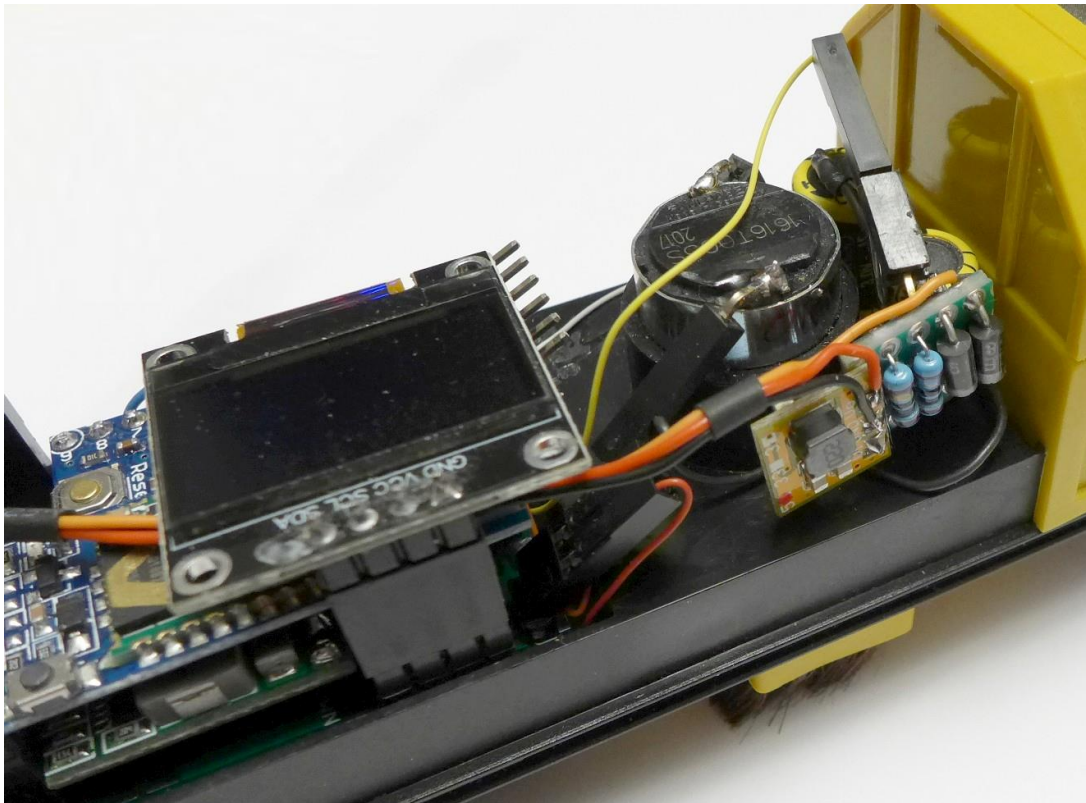


Das graue und orange Kabel gehen zum Motor. Der Stecker mit dem gelben Kabel wird an den Pluspol der SuperCaps gesteckt. Die Versorgung von den Schienen wird über das rote und schwarze Kabel und den Stecker hinten links eingespeist.

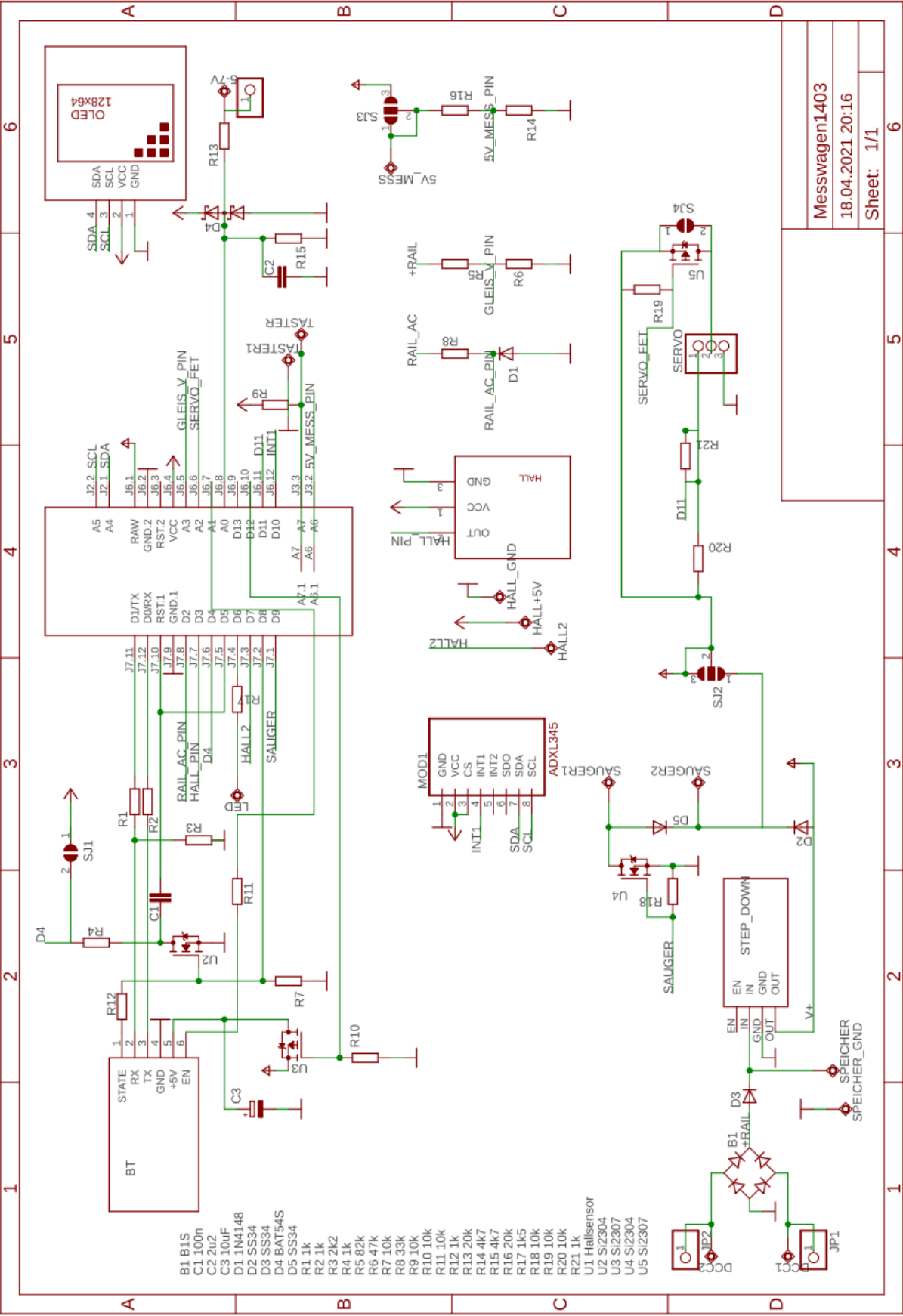
Im nächsten Bild ist alles zusammengesteckt. Das Beschleunigungssensor Modul ist über eine zusätzliche Platte mit dem Wagen verschraubt, weil es sonst in der Steckverbindung hin und her wackelt. Evtl. wäre es besser gewesen, wenn man das Modul direkt in die Platine gelötet hätte. Für die Softwareentwicklung ist ein Stecker aber von Vorteil. Es ist nicht notwendig, dass das Modul exakt senkrecht montiert wird. Das wird bei der Kalibrierung ausgeglichen.



Ausschnitt mit den SuperCaps und dem StepUp:



Schaltbild:





## Geänderte Bauteile gegenüber dem Schaltbild:

R6: 4.7K

R19: Nicht bestückt \*

R20: Nicht bestückt

R21: Nicht Bestückt und Brücke darunter getrennt

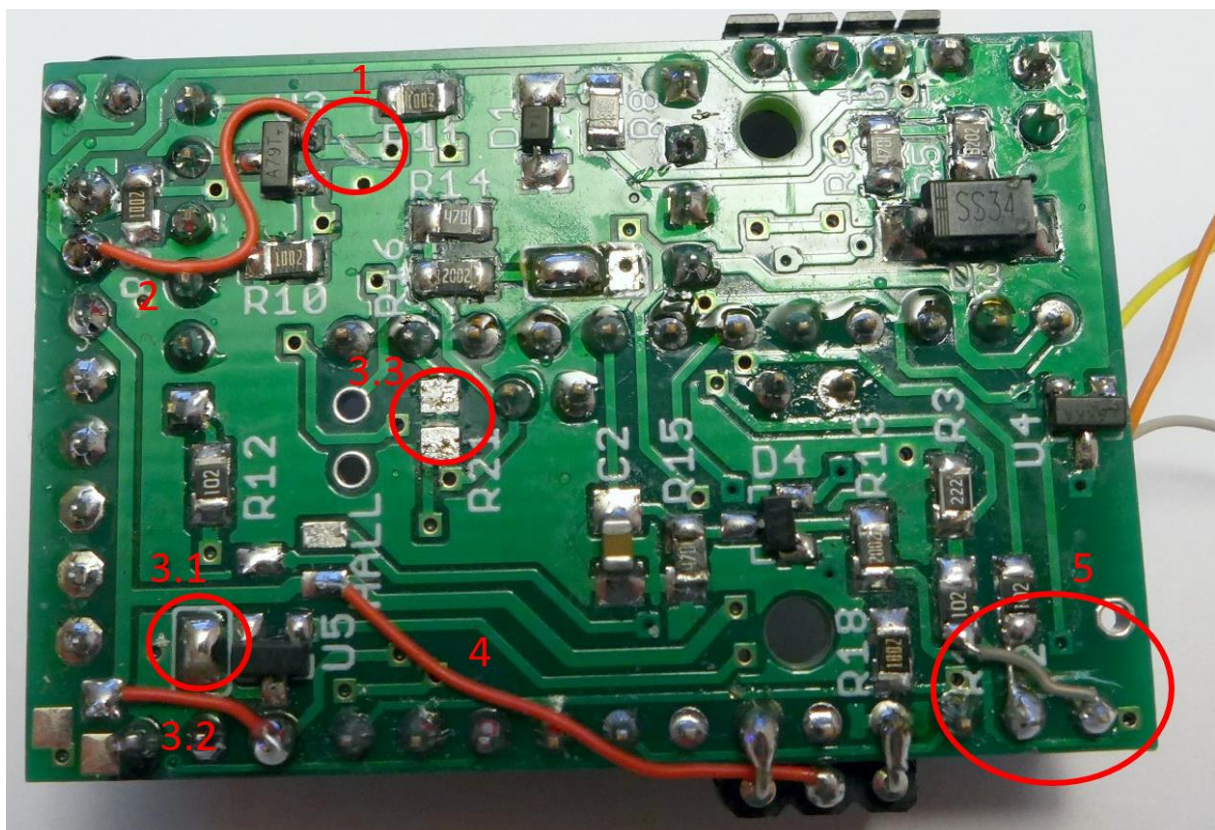
U1: Nicht bestückt

U5: Nicht bestückt \*

\*: In den Bildern noch vorhanden

## Änderungen auf der Platine

1. Versorgung vom U3 (Power BT-Modul) darf nicht direkt an der DCC Versorgung hängen sonst wird bei Spannungsschwankungen ein Reset generiert
2. Versorgung U3 von 5V des Arduinos
3. Der Servo Stecker wird zum Anschluss des StepUp Wandlers und des SuperCap benutzt. Dazu wird SJ4 verbunden (3.1) und 5V des Arduinos mit Pin 1 verbunden (3.2). Die Widerstände R20 und R21 sind nicht bestückt und die Verbindung unter R21 wird getrennt (3.3). Diese Änderungen sind nur notwendig, wenn nicht der vorgesehene SuperCapLader von Matthias verwendet wird.
4. Der Vorgesehene Hall Sensor auf der Platine ist nicht empfindlich genug und hat zumindest für die Magnete in meinem Lux Wagen die falsche Polarität. Darum wurde ein Stecker für den im Drehgestell des Wagens verbauten Hall Sensor angebracht. Die Brücke speist die 5V ein.
5. Beim Arduino Symbol im Schaltplan sind die Pin-Nummern 11 und 12 vertauscht. Dieser Fehler wird durch Kreuzen der Signale behoben. Die Original Leiterbahnen wurden getrennt und durch gekreuzte Kabel ersetzt.

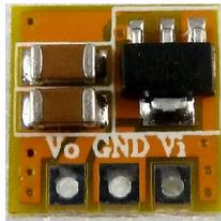


## SuperCap

Die Schaltung soll über eine zusätzliche SuperCapLader Platine von Matthias bei einer Unterbrechung der Strom versorgt werden. Dafür sind die beiden Lötspots SPEICHER+ und SPEICHER\_GND vorgesehen.

Da ich aber die wertvollen Platinen lieber in die Lokomotiven verbaue habe ich eine alternative Notstromversorgung verwendet. Diese Schaltung nutzt einen zusätzlichen StepUp Wandler:

<https://de.aliexpress.com/item/32800430445.html>

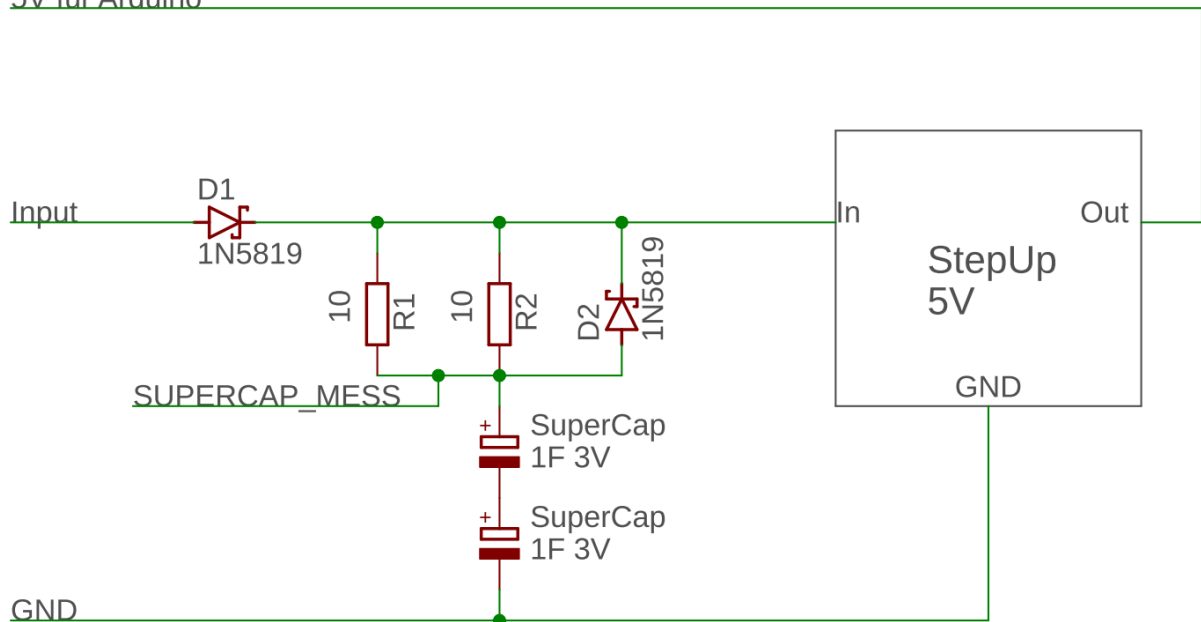


Dieser wird über drei Kabel mit der Grundplatine verbunden. Dazu habe ich den Servo Anschluss umfunktioniert. Der StepUp Wandler, zwei Widerstände und zwei Schottky Dioden werden über eine kleine Platine mit den SuperCaps verbunden. Eine 4. Leitung dient der Messung der Kondensatorspannung. Sie führt zu dem Lötspot „SUPERCAP\_MESS“.

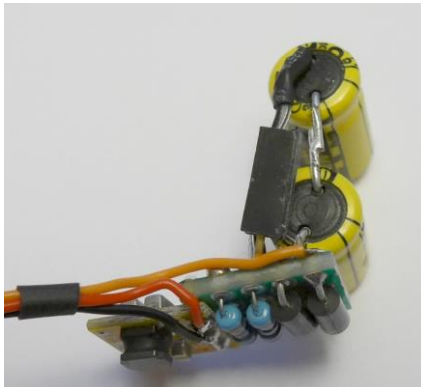
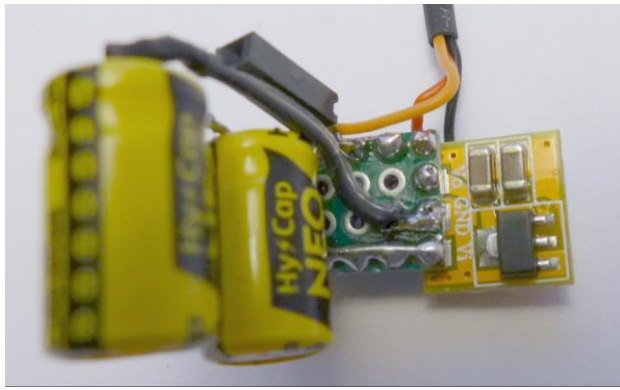
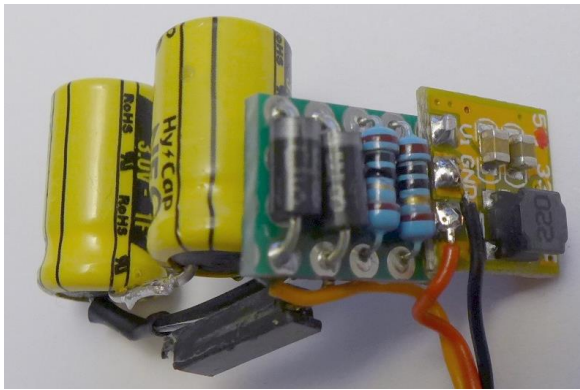
Der StepDown Spannungsregler auf der Grundplatine ist über das Poti auf 5.7V eingestellt. Diese Spannung ist die maximale Spannung des SuperCaps.

Der folgend Plan zeigt den Aufbau der zusätzlichen Schaltung. D1 verhindert den Rückfluss aus den Kondensatoren zum Sauger Motor. Die beiden Parallelgeschalteten Widerstände begrenzen den Ladestrom der Kondensatoren. Sie sind parallelgeschaltet zur Reduzierung der Verlustleistung. Vermutlich wäre das aber nicht nötig. D2 soll die Verlustleistung reduzieren. Sie wird aber erst bei Strömen über 40mA aktiv. Vielleicht sollten die Widerstände noch weiter verringert werden (4.7 Ohm) damit die Kondensatoren schneller geladen werden.

### 5V für Arduino







### Pufferung der Staubsauger Motors per SuperCap

Der SuperCap versorgt bei beiden Schaltungsvarianten auch den Staubsauger Motor. Wenn die SuperCap Ladeplatine verwendet wird ist das im Schaltplan klar ersichtlich. Hier wird die Spannung vor den StepDown Regler eingespeist.

Wenn man die oben vorgestellte Variante mit einem zusätzlichen StepUp Wandler benutzt ist das nicht so leicht ersichtlich. Hier sorgt eine Schutzdiode auf der Nano Platine dafür, dass die 5V zurück zum „RAW“ Eingang fließen und darüber den Staubsauger Motor versorgen.

Es macht durchaus Sinn, dass der Staubsauger Motor eine gewisse Zeit lang vom SuperCap gespeist wird, wenn die Schienen verschmutzt sind. Aber der Arduino sollte trotzdem möglichst lange versorgt werden. Darum wird der Motor abgeschaltet, wenn die Spannung am SuperCap unter `VACU_DISAB_VOLT = 5.3V` fällt. Der Motor wird wieder Freigeschaltet sobald die Spannung am Kondensator über `VACU_ENAB_VOLT = 5.4V` ist. Damit wird verhindert, dass der Motor den Kondensator zu stark entlädt. Die Beiden Spannungen können per „#define“ eingestellt werden. Das ist dann nötig, wenn die Maximalspannung des SuperCaps auf eine kleinere Spannung als 5.7V eingestellt ist.

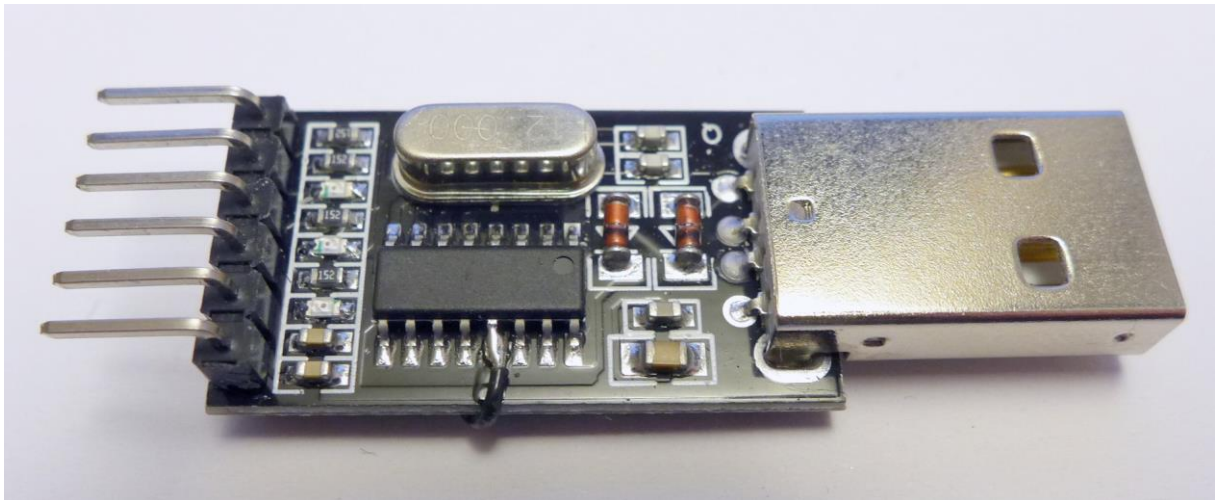
**Achtung:** Wenn die Ladeschaltung so dimensioniert ist, dass die SuperCap Spannung die 5.4V nie erreicht, dann funktioniert der Stabsauger nicht.

Das automatische Ein- und Ausschalten des Motors kann man über die Compiler Konstante „`VACU_SUPERCAP_VOLT_CONTR`“ deaktivieren.

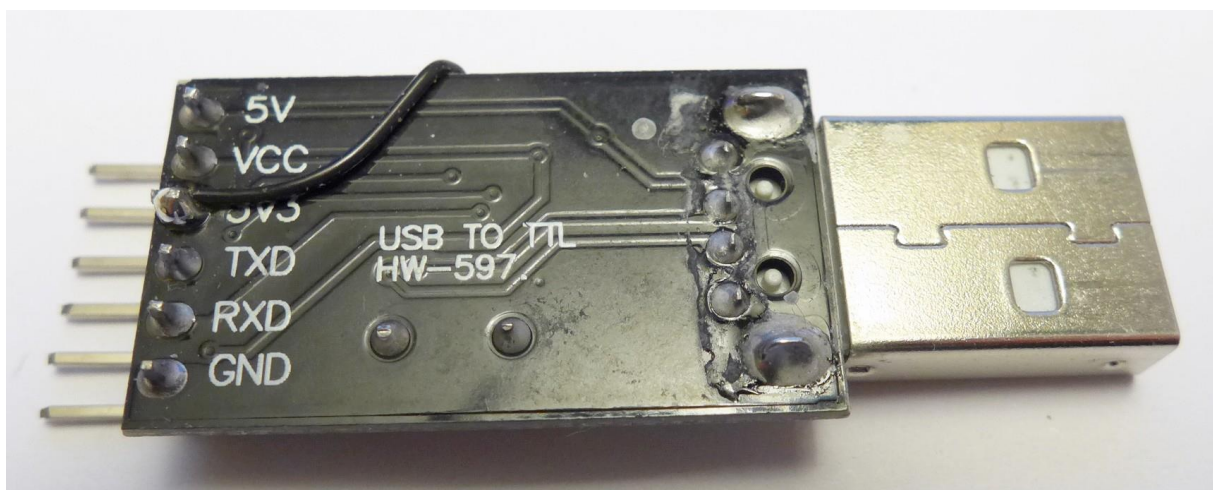
### Programmierung des Nano Mini Pro

Zur Programmierung eines Nano Mini Pro benötigt man einen USB nach TTL Adapter. Leider gibt es unterschiedliche Typen dieser Adapter. Bei manchen fehlt die DTR Leitung welche den Nano zum

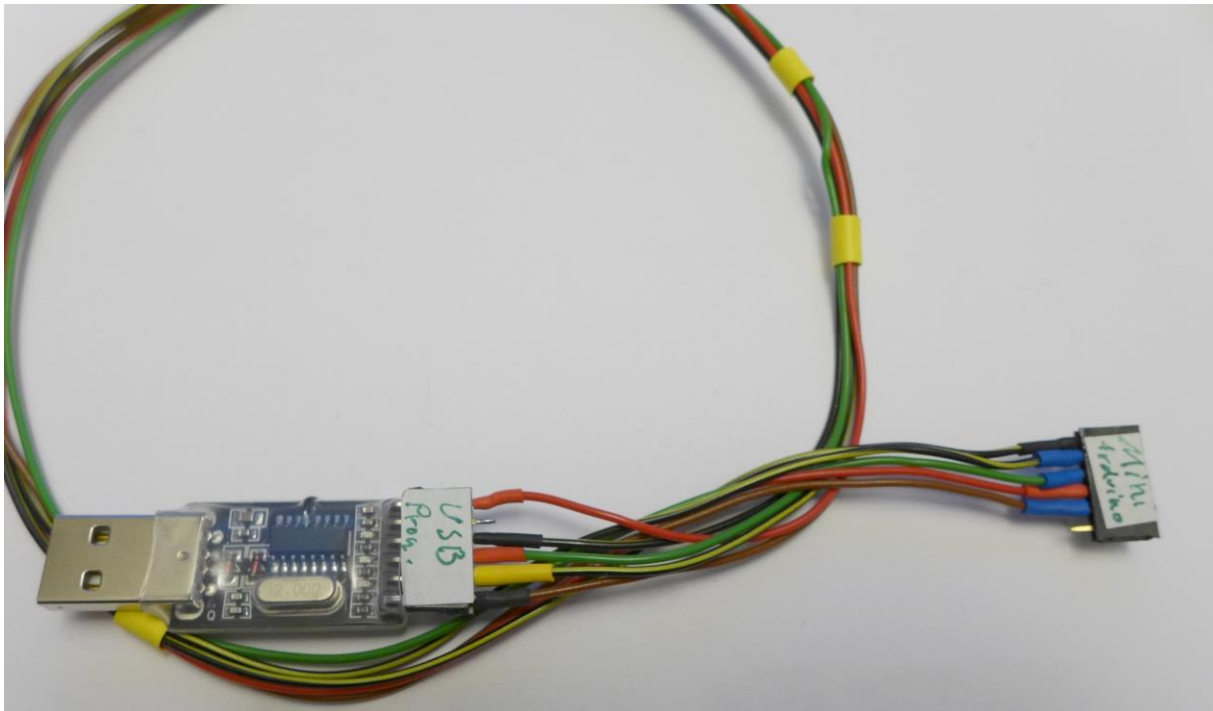
programmieren neu startet. In dem Fall muss man, wenn die Anzeige „Hochladen...“ in der Arduino IDE kommt auf den Reset Knopf des Nanos drücken. Dazu hat man beliebig viel zeit, es ist aber trotzdem lästig. Darum habe ich meinen Adapter umgebaut:



Auf der Unterseite habe ich die 3.3V Verbindung getrennt und das DTR Signal angelötet:



Das Kabel zum Programmieren muss diese Belegung haben:



Achtung: Das ist nur dann nötig, wenn der Adapter keinen automatischen Reset generiert und geht nur, wenn man genau diesen Adapter hat.

Zur Programmierung muss in der Arduino IDE der richtige Prozessor ausgewählt sein:

Eine Programmierung während der Messwagen per DCC Versorgt wird kann problematisch sein.

Zum einen, weil dann die DCC Spannung mit der Masse des PCs verbunden wird. Ich schalte darum einen Optisch getrennten USB Adapter dazwischen.

Zum anderen kann das Reset Signal nicht erkannt werden, weil die Spannung am USB-Port sich zu sehr von der Spannung des Moduls unterscheiden. Bei mir funktioniert es, wenn man die +5V vom Programmieradapter trennt. Normalerweise muss der Arduino nur einmal per USB Programmiert werden. Danach kann das Programm über Bluetooth Hochgeladen werden. Allerdings ist das zehnmals. Es dauert 155 Sek anstelle von 14 Sek.

Werkzeuge Hilfe	
Automatische Formatierung	Strg+T
Sketch archivieren	
Kodierung korrigieren & neu laden	
Bibliotheken verwalten...	Strg+Umschalt+I
Serieller Monitor	Strg+Umschalt+M
Serieller Plotter	Strg+Umschalt+L
WiFi101 / WiFinINA Firmware Updater	
Board: "Arduino Pro or Pro Mini"	>
Prozessor: "ATmega328P (5V, 16 MHz)"	>
Port: "COM3"	>
Boardinformationen holen	
Programmer: "ArduinoISP"	>
Bootloader brennen	

## Inbetriebnahme des Bluetooth Moduls

Nachdem die Software auf dem Arduino installiert wurde muss das BT-Modul konfiguriert werden. Dazu hält man die Taste am BT-Modul fest während man die Versorgungsspannung des Messwagens anschaltet. Die Taste muss 5 Sekunden lang gehalten werden. Es reicht nicht, wenn man anstelle des Trennens der Versorgungsspannung nur den Reset Knopf des Arduinos betätigt.

Achtung: Evtl. ist der Schrumpfschlauch über das BT-Modul zu steif zum zuverlässigen betätigen des Tasters.

Die Konfiguration erfolgt automatisch. Das BT-Modul bekommt in diesem Schritt den Namen "Messwagen?" zugewiesen (Siehe #define BT\_NAME).

Während der Konfiguration Blinkt die LED auf den Arduino ein paar Mal unregelmäßig kurz.



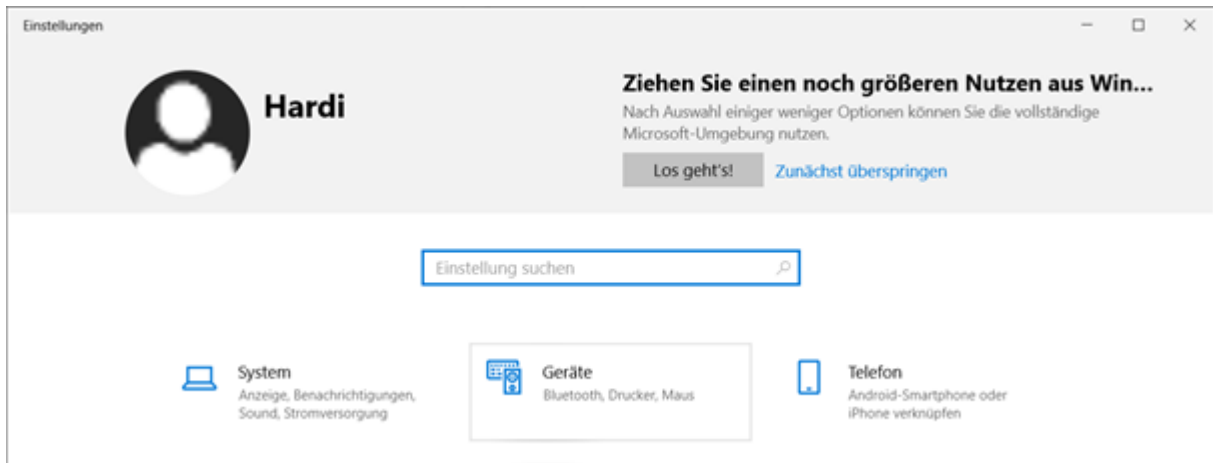
Wenn alles gut gegangen ist, dann blinkt sie am Ende 5-mal langsam (0.5Hz).  
Im Fehlerfall blinkt sie 20-mal ganz schnell.

### Koppeln mit dem Rechner

Anschließend muss das BT-Modul mit dem Windows Rechner (oder dem Smartphone) gekoppelt werden.

Hier die Beschreibung für Windows 10:

In den Windows Einstellungen (Tastenkombination WIN+I) geht man zu "Geräte Bluetooth, Drucker, Maus".



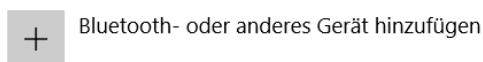
Zunächst prüft man ob der "Messwagen?" evtl. schon von früheren Versuchen gekoppelt ist. Dazu scrollt man auf der Seite nach unten.



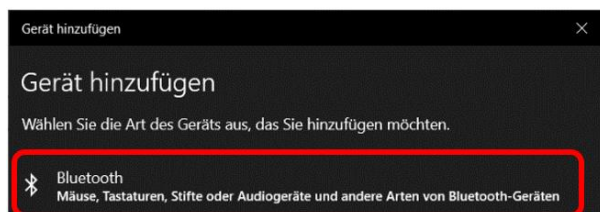
Hier sind bereits zwei Messwagen vorhanden. Diese muss man mit einem Klick auf den Eintrag entfernen.

Wenn der Messwagen noch nicht verbunden ist, dann klickt man ganz oben auf "Bluetooth- oder anders Gerät hinzufügen"

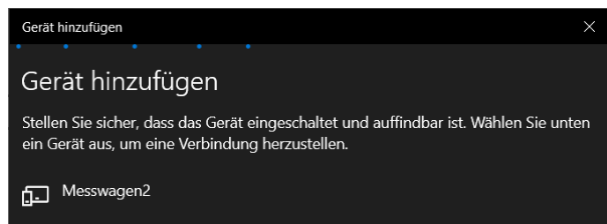
### Bluetooth- und andere Geräte



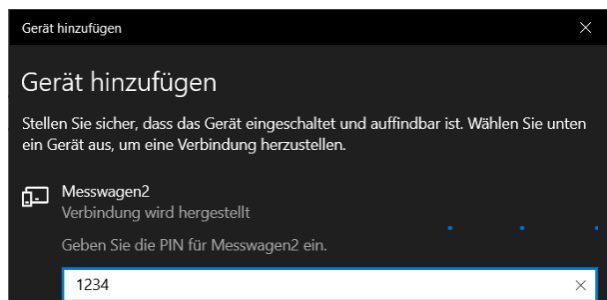
und Anschließend "Bluetooth Mäuse, ...".



Dann sollte nach einiger Zeit der „Messwagen?“ auftauchen, wenn er nicht bereits verbunden war.

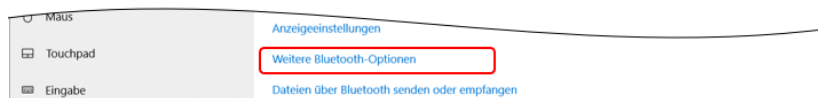
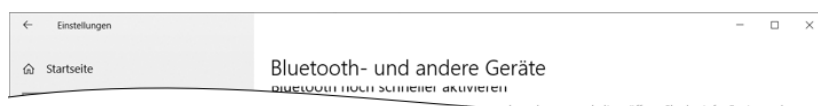


Mit einem Klick auf den Namen kommt man zur Password Eingabe. Standardmäßig wird "1234" verwendet.



Mit „Verbinden“ sollte der Messwagen in der Liste der gekoppelten Geräte erscheinen.

Jetzt muss man noch herausfinden welchen COM Port Windows für den Messwagen reserviert hat. Dazu klickt man ganz unten in dem "Bluetooth- und andere Geräte" Fenster auf "Weitere Bluetooth-Optionen".



In dem erscheinenden Fenster aktiviert man den Tab "COM-Anschlüsse". Dort sollte der "Messwagen?" zwei Mal erscheinen. Der erste COM Port mit der Richtung "Ausgehend" ist der Port über den der Wagen angesprochen werden kann.



Er wird in der Arduino IDE Gewählt, wenn man die Daten sehen will oder wenn man das Programm per Bluetooth zum Arduino schicken will.

Diese Schritte müssen nur einmal gemacht werden.

Dummerweise gibt es unterschiedliche BT Module und auch bei optisch genau dem gleichen Modul kann eine andere Software installiert sein. Bei dem von mir verwendeten Modul ist die Version 4.0 installiert (Kann über AT Kommandos ermittelt werden). Mit dieser Software funktioniert das invertieren des Status Signals nicht mehr. Bei der älteren Version 2.0 geht es laut Angaben im Netz noch. Darum habe ich bei der Schaltung einen Transistor vorgesehen mit dem das Signal invertiert wird. Damit sollte es mit beiden Softwareständen zu benutzen sein.

Es kann aber sein, das andere Inkompatibilitäten dazu führen das bestimmte BT-Module nicht benutzt werden können.

Das BT-Modul „BEI-09 Android IOS BLE 4,0“ funktioniert **NICHT** mit der aktuellen Version des Programms.

## Displays

## Steuerung per Klopfezeichen

## Status LEDs

Arduino

Bluetooth Modul

## DCC Befehle

## Anzeige der Werte am PC

## Befehle der seriellen Schnittstelle

## Handy Programm

## Konfiguration des Programms

## Links zu den verwendeten Zusatzmodulen:

Nano Mini Pro: <https://de.aliexpress.com/item/32605434250.html>

StepDown Wandler mit einstellbarer Ausgangsspannung: <https://de.aliexpress.com/item/32826540392.html>

StepUp Wandler 5V Ausgang: <https://de.aliexpress.com/item/32800430445.html>

Bluetooth Modul HC05 new version 6-Pins: <https://de.aliexpress.com/item/1005001636656116.html>

BeschleunigungssensorADXL345: <https://de.aliexpress.com/item/32281429971.html>

0.96" I2C 128X64 OLED SSD1306: <https://de.aliexpress.com/item/32643950109.html>

0.91" I2C 128x32 OLED SSD1306: <https://de.aliexpress.com/item/4001028654247.html>

0.87" I2C 128x32 OLED SSD1316: <https://de.aliexpress.com/item/4000182887362.html>

1.3" I2C 128X64 OLED SH1106: <https://de.aliexpress.com/item/32683739839.html>