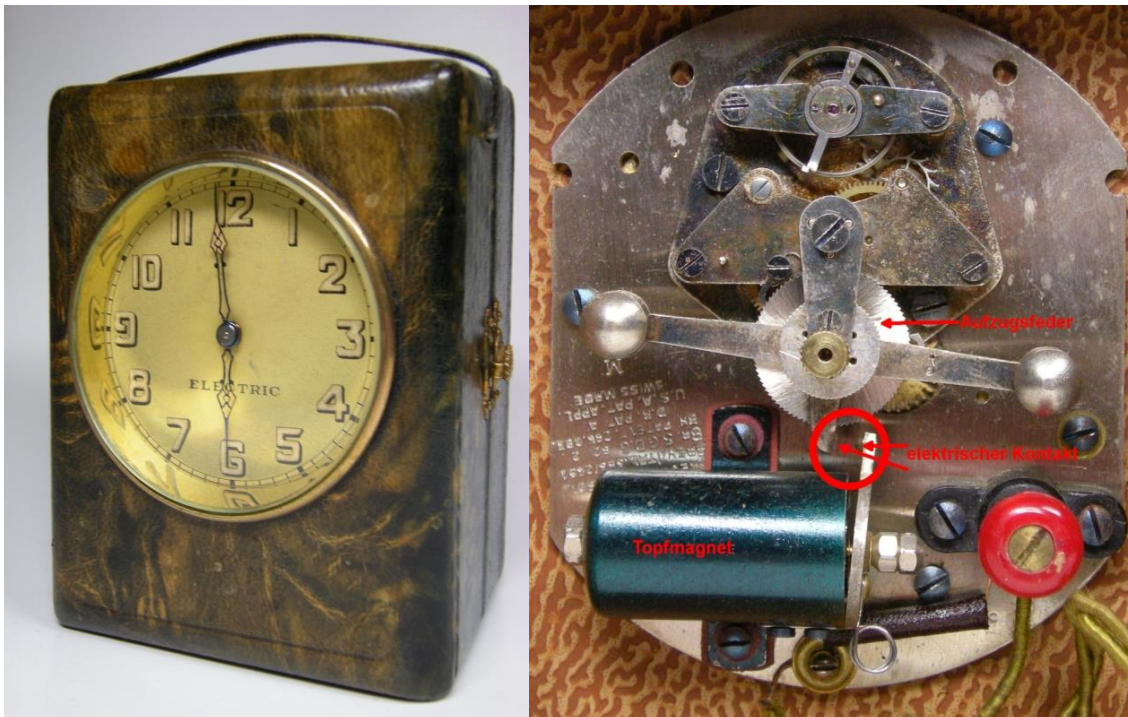


Strommessung mit dem Arduino

Das Buch von Burkard Kainka über sein "Arduino Messlabor" war für mich wirklich inspirierend. Insbesondere die ersten drei Kapitel zeigen auf, was mit einem Arduino möglich und auch nicht möglich ist. Für meine spezielle Messaufgabe erschien mir aber sein Messlabor aus zwei Gründen nicht ausreichend. Erstens sollten Spannungen von weniger als 100mV gemessen werden und zweitens sollte ein Trigger Delay einstellbar sein, um auch Werte vor dem Triggerzeitpunkt darstellen zu können.

Die Messaufgabe

Als Sammler elektrischer Uhren, habe ich eine Menge von Uhrwerken mit Magnetaufzug in der Sammlung. Bei dieser Art des Aufzugs wird durch Ablaufen der Uhr ein elektrischer Kontakt geschlossen und dadurch ein Elektromagnet betätigt, der ein Gewicht oder eine Feder aufzieht. Die mechanische Trägheit des Systems sorgt dafür, dass der Kontakt am Ende des Aufzugs wieder geöffnet wird. Es geht also um die Messung von Kontaktdauer und Stromverlauf.



Im diesem Beispiel verwende ich ein Uhrwerk der Schweizer Firma Schild & Co., welches fast unverändert ab 1926 gebaut wurde. Betrieben wird es mit einer 4,5V Flachbatterie, die erwartete Impulsdauer ist in der Größenordnung von einigen 10ms und der erwartete Strom ca. 700mA. Die Messung des Stroms erfolgt über eine Spannungsmessung an einem in die Versorgungsleitung eingeschleiften Shunt-Widerstand. Der Spannungsabfall sollte klein gegenüber der Betriebsspannung sein.

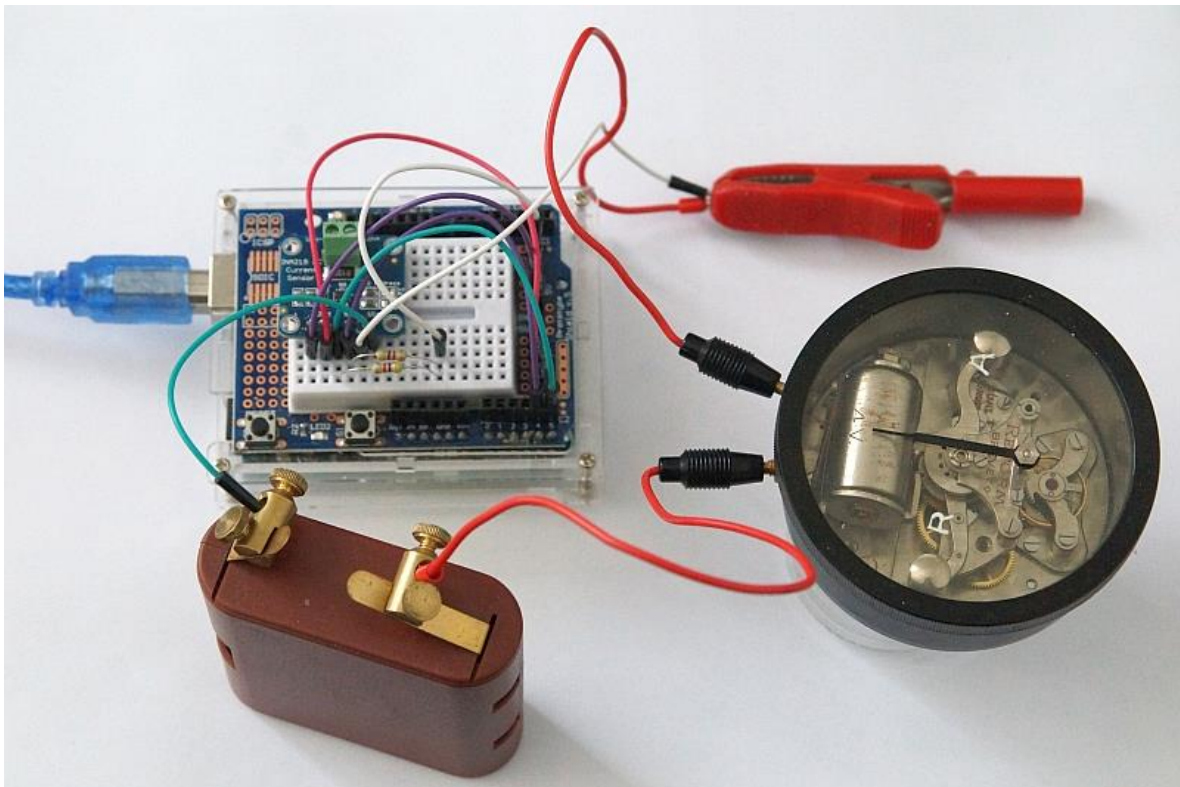
Ein Shunt-Widerstand von 10Ohm erzeugt einen Spannungsabfall von ca. 0,7V, der zwar mit einem Arduino gut messbar, aber bereits beträchtlich gegenüber der Betriebsspannung ist. Zum Glück hatte ich noch ein Strommessmodul mit dem bekannten Chip INA219 herumliegen, welches einen eingebauten Shunt-Widerstand von 0,1 Ohm hat. Das Modul ist aber eigentlich nicht für Messungen

in solch kurzen Zeitabständen gedacht. Mit den folgenden Vereinfachungen ist die Messaufgabe aber doch zu lösen:

- Bei einer Auflösung von 9-Bit erreicht der AD-Wandler des Chips eine Konversionszeit von 83 Mikrosekunden. Diese Genauigkeit reicht für eine Darstellung.
- Normalerweise misst der Chip Bus- und Shunt-Spannung, also zwei Messungen, um neben der Stromaufnahme auch die Leistungsaufnahme berechnen zu können. Hier reicht die Messung der Shunt-Spannung. Damit ist auch die zeitliche Auflösung ausreichend.
- Hauptproblem ist aber die Übertragung der Messwerte per I2C-Interface. Dies ist vergleichsweise langsam und die Arduino Wire Bibliothek ist ohne das ganze Arduino Framework nicht nutzbar. Ohne diese funktionieren auch die erhältlichen INA219 Bibliotheken nicht. Hier wird eine sehr einfache Ansteuerung der I2C Hardware des Arduino verwendet, ohne Interrupt und ohne Fehlerbehandlung. Gleichzeitig konnte die Geschwindigkeit des I2C Busses auf 800kHz erhöht werden. Andere I2C Hardware wird in dieser Konstellation natürlich nicht laufen.

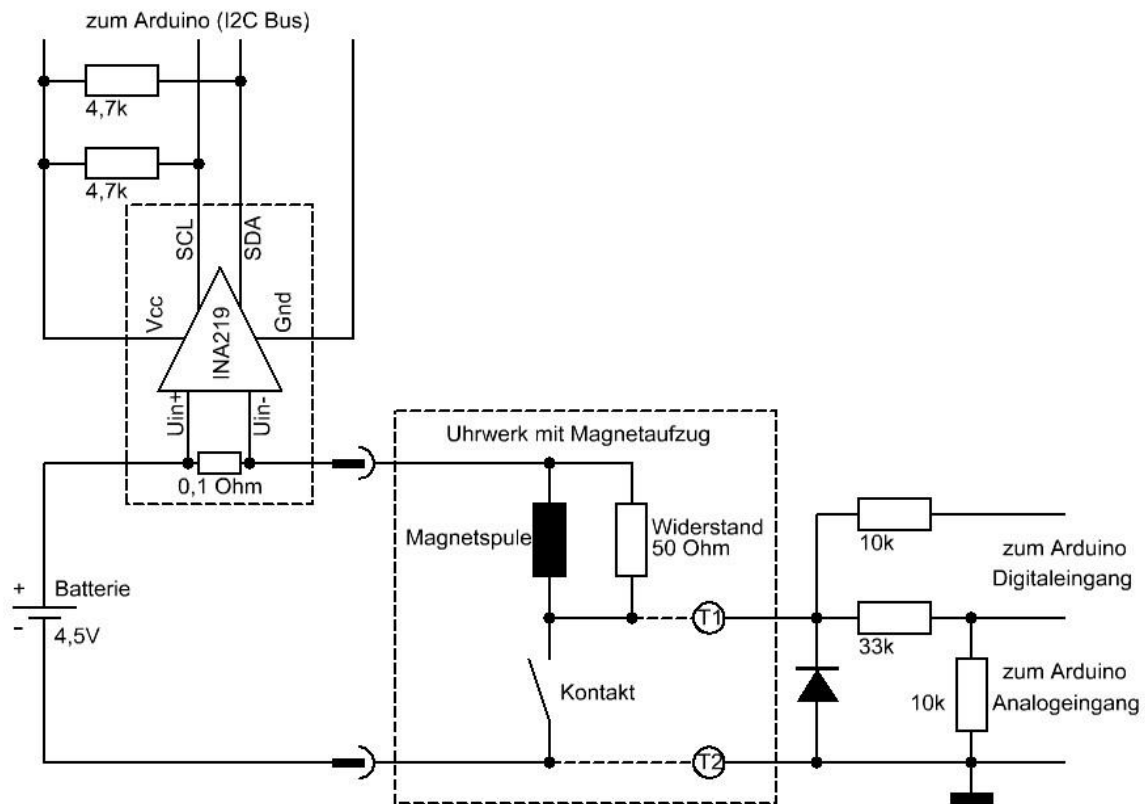
Bei dieser sehr speziellen Messaufgabe ist die eigentliche Messzeit (100ms) sehr klein gegenüber der Periodendauer (Aufzugsintervall ca. 3 Minuten). Daher ist eine Ausgabe über den Seriellen Plotter der Arduino IDE ausreichend. Außerdem lassen sich die Messwerte so einfach in Excel kopieren und als beschriftetes Diagramm darstellen. Auch eine Steuerung der Parameter über eine Benutzeroberfläche ist nicht notwendig. Bei geänderten Parametern wird der Sketch einfach neu kompiliert. Dazu reicht die Zeit zwischen den Messungen allemal aus. Daher wird der Arduino Sketch (die Firmware) nicht an das Frontend des Arduino Messlabors angepasst oder umgekehrt.

Die Messung

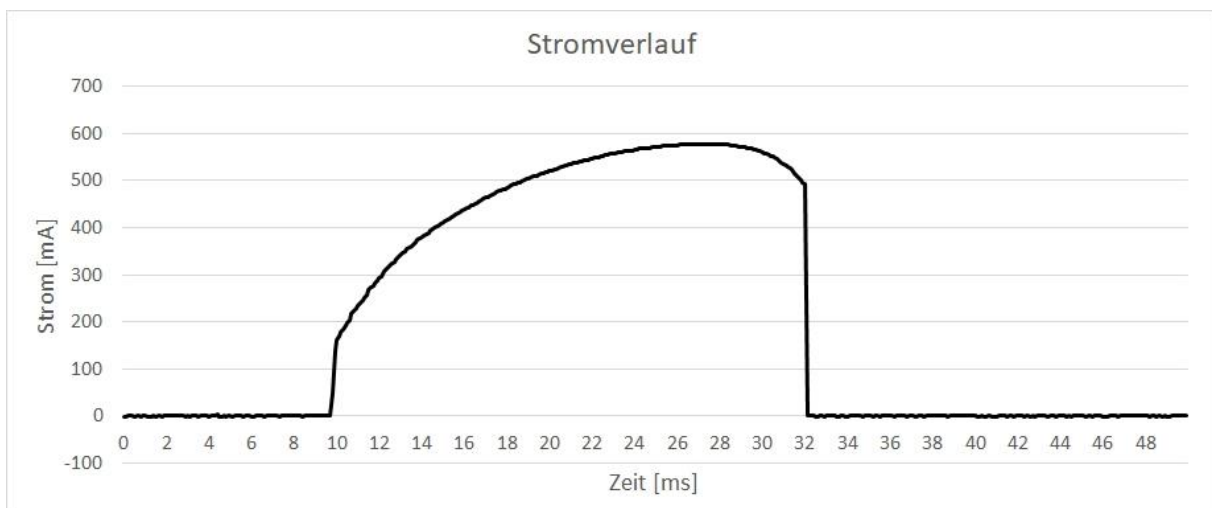


Das Bild zeigt den Messaufbau. Als Messobjekt dient jetzt ein loses Werk aus meiner Sammlung. Auf dem Arduino Uno sitzt ein Prototyping Shield, auf das das INA219 Modul gesteckt ist. Außer zweier Pullup-Widerständen für den I2C-Bus sind nur noch ein paar Drahtverbindungen nötig. Das Modul ist

für sogenannte "high side" Messung vorgesehen, es muss also in der positiven Leitung der Betriebsspannung liegen. Der Arduino Uno wird über den sowieso nötigen USB Anschluss mit Strom versorgt. Der "Schaltplan" für das Uhrwerk selbst ist denkbar simpel.



Als Ergebnis der Messung erwartet man einen bei 0 beginnenden Strom, der bis zum durch den ohmschen Widerstand der Spule festgelegten Maximalwert ansteigt.

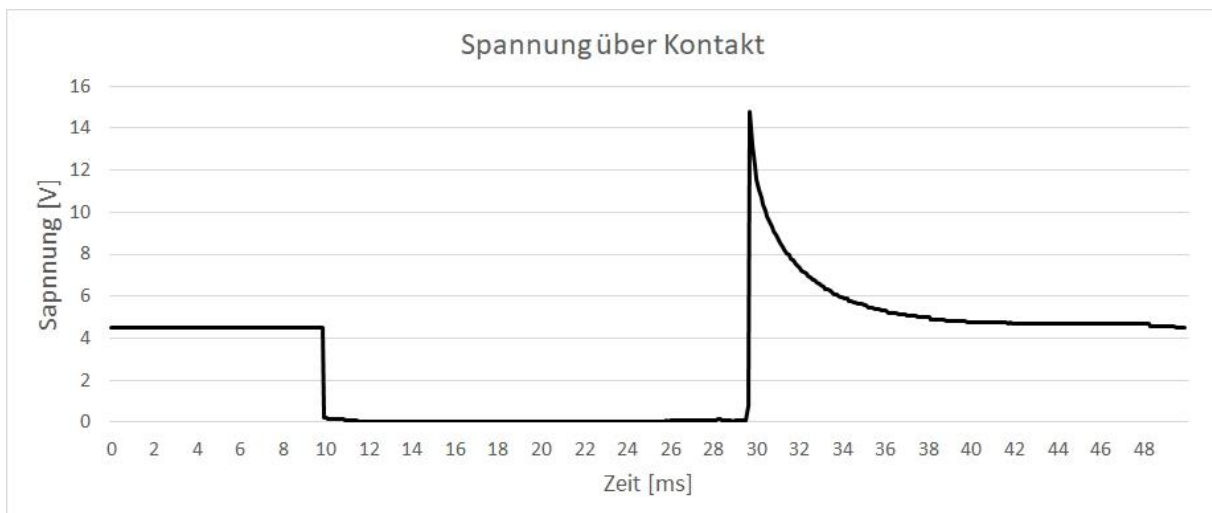


Der gemessene Stromverlauf sieht aber etwas anders aus. Dies hat zwei Ursachen.

Zum einen sieht man, dass ein Strom direkt nach dem Einschalten fließt. Daran erkennt man, dass der Spule ein Widerstand zur Funkenlöschung parallelgeschaltet ist. Die Höhe dieser Einschaltstroms deutet auf einen Widerstand von 30 Ohm hin, was nicht ganz den Angaben des Herstellers (50 Ohm) entspricht.

Als zweites fällt ein Abfall des Stroms kurz vor dem Abschalten auf. Die Ursache liegt darin, dass die Induktivität nicht konstant ist. Wenn während des Aufzugvorgangs der Hebel vom Topfmagneten angezogen wird, so nimmt die Induktivität signifikant zu und erreicht ihr Maximum, wenn der Anker den Topfmagnet schließt und damit der Strom abgeschaltet wird.

Wie im Schaltplan oben angedeutet, kann man auch die Spannung über dem Kontakt messen. Dazu wird der interne ADC des ATmega verwendet. Wegen des zu erwartenden Abschaltfunks, wird der Analogeingang über einen Spannungsteiler an den Kontakt angeschlossen. Die Diode soll eine versehentlich Verpolung und damit eine mögliche Zerstörung des Prozessors verhindern.



Bei offenem Kontakt wird die volle Batteriespannung gemessen, bei geschlossenem Kontakt natürlich 0 Volt. Zum Abschaltzeitpunkt erreicht die Spannung über dem Kontakt ein Mehrfaches der Batteriespannung. Durch den der Spule parallel geschalteten Funkenlöschwiderstand, hält sich die Spannungsspitze aber noch in Grenzen. Ohne wirksame Funkenlöschung sollte eine solche Messung nicht durchgeführt werden. Der entstehende Funke würde den Analogeingang sofort zerstören.

Verbindet man den Kontakt wie im Schaltplan gezeigt mit einem Digitaleingang des Arduino, so lassen sich auch Langzeitmessungen der Kontaktdauer, z.B. auch in Abhängigkeit der Versorgungsspannung des Uhrwerks durchführen.

Der Sketch

Der Sketch ist wie im "Arduino Messlabor" zwar mit der Arduino IDE, aber in purem C programmiert und damit nicht portabel. Er wurde nur auf einem Arduino Uno getestet, sollte aber zumindest auf allen ATmega328 Prozessoren laufen. Er ist für die Messung des Stroms über ein INA219 Modul oder der Spannung über den internen ADC geeignet. Das Abtastintervall wird über einen Timer (Timer0 oder Timer2) gesteuert, der auch den ADC startet. In den Interrupt Routinen ("Timer/Counter Compare match" bei Strommessung bzw. "ADC Conversion complete" bei Spannungsmessung) wird nur eine Flagge gesetzt, die in der Endlosschleife des Hauptprogramms ausgewertet wird. Die Messwerte werden in einem Ringpuffer gespeichert. Dieser ist korrespondierend zu dem Seriellen Plotter der Arduino IDE 500 Messwerte groß.

Das Abtastintervall (die Messzeit für 500 Punkte) wird am Anfang des Sketches festgelegt und nach diesem #define werden zu Beginn des Hauptprogramms die Komponenten (Timer, ADC und INA219) konfiguriert. Für die Strommessung beträgt die minimale Messzeit 50 ms, für die Spannungsmessung 10 ms. Messzeiten über 5 Sekunden sind mit dem 8-Bit Timer nicht möglich.

Zeitbasis (bei 500 Punkten)			Timer		ADC			INA219	
Zeitbasis [ms]	Abtastintervall [ms]	Samples/s	Prescaler	OCRA	Prescaler	Clock [kHz]	Conv. Time [ms]	Mode	Conv. Time [ms]
10.000	20	50	1024	311,5	64	250	0,052	8 Samples	4,26
5.000	10	100	1024	155,3	64	250	0,052	4 Samples	2,13
2.000	4	250	256	249	64	250	0,052	2 Samples	1,06
1.000	2	500	256	124	64	250	0,052	12 Bit	0,532
500	1	1000	64	249	64	250	0,052	11 Bit	0,276
200	0,4	2500	64	99	32	500	0,026	10 Bit	0,148
100	0,2	5000	64	49	32	500	0,026	9 Bit	0,084
50	0,1	10000	8	199	32	500	0,026	9 Bit	0,084
20	0,04	25000	8	79	16	1000	0,013	n/a	n/a
10	0,02	50000	8	39	16	1000	0,013	n/a	n/a

In der Endlosschleife des Hauptprogramms wird zunächst die Flagge aus den Interrupt Routinen geprüft und wenn diese gesetzt ist, wird der Messwert vom INA219 Modul bzw. vom ADC geholt und der Ringpuffer gefüllt. Dabei wird auch geprüft, ob der Trigger Level erreicht wurde. Außerdem gibt es eine kleine Statusmaschine. Zunächst werden nur Daten in den Ringpuffer geschrieben. Wenn der Trigger Level zweimal überschritten wurde, so wechselt der Status und es werden nur noch maximal 500 Messwerte gesammelt, abzüglich des Trigger Delays, also der Anzahl der Messwerte die noch vor dem Trigger Zeitpunkt dargestellt werden sollen. Danach wechselt der Status zur Ausgabe der Messwerte über die serielle Schnittstelle. Nach dieser Ausgabe wechselt der Status wieder in den Ursprungszustand.

Die Kommunikation mit dem Strommessmodul ist in einer separaten Datei untergebracht. Die Programmierung der I2C Hardware entspricht dem Atmega328 Datenblatt, ohne detaillierte Prüfung der Statusregister, also eigentlich ohne Fehlerprüfung und -behandlung. Darauf aufbauend gibt es einige wenige Routinen zur Steuerung des INA219. Diese erlauben das Schreiben des 16-Bit Konfigurationsregisters und das Lesen der 16-Bit Shunt Spannung. Bei letzterer wird beim ersten Lesen die interne Adresse des Shunt voltage register gesendet, die bei den folgenden Lesevorgängen nicht mehr benötigt wird. Die Routinen sind dafür vorgesehen, auch mehrere solcher Module am I2C-Bus betreiben zu können.

Insgesamt ist der Sketch recht einfach, bis auf das Setzen der Register zur Konfiguration der Prozessor Hardware. Damit sollte er auch leicht für andere, ähnliche Messaufgaben änderbar sein. Verbesserungswürdig ist auf jeden Fall die Triggererkennung. Wenn die zu messenden Einzelereignisse zu schnell aufeinander folgen, so könnte nach dem Senden der Daten an den PC eine neue Messung auch erst durch einen Tastendruck eingeleitet werden.

Fazit

Ein Arduino wird ein DSO wohl nicht ersetzen können. Für diese einfachen Messungen ist er aber mehr als ausreichend, zumindest wenn sie nicht zu häufig vorkommen. Das INA219 Modul hat mir für diese Messaufgabe den Kauf eines Shuntwiderstandes und den Bau eines Verstärkers erspart. Ich werde mir diesen Aufbau auf einem Prototyping Shield fest verlöten, um ihn bei passender Gelegenheit zur Hand zu haben. Einen Arduino habe ich wohl immer hier rumliegen.

Hartmut Wynen

kontakt@wynen.info