Arduino und Sensoren mit 4mA - 20mA Stromschleifenschnittstelle

In der Industrie sind Sensoren mit Stromschleifenschnittstellen üblich. Dabei wird durch den Sensor nicht eine Spannungsänderung sondern eine Stromänderung - meist von 4mA bis 20mA ausgegeben. Wie man derartige Sensoren mit einem Arduino ausliest, möchte ich kurz erläutern.

Wikipedia fast die analoge Stromschleife ausgezeichnet zusammen:

Werden analoge Spannungswerte von einem Messwertgeber über ein langes Kabel zu einem Empfänger übertragen, so werden diese verfälscht. Ursache ist der nötige Messstrom, der im Kabel aufgrund des unerwünschten Kabelinnenwiderstandes einen Spannungsabfall erzeugt. Durch möglichst hochohmige Eingänge lassen sich der Messstrom und damit auch der Messfehler zwar minimieren, jedoch auf Kosten einer großen Empfindlichkeit gegenüber kapazitiven Störungen. Induktiv eingekoppelte Störspannungen lassen sich so nicht beseitigen.

Daher werden Messwerte über größere Distanzen bevorzugt als Strom (4 - 20 mA) übertragen. Bei der Stromübertragung spielen der Innenwiderstand und damit der Spannungsabfall des Kabels keine Rolle, solange die Spannung ausreicht. Der

Isolationswiderstand hat zwar nun einen Einfluss, er lässt sich jedoch mit heutigen Isolierstoffen sehr hoch halten. Der Strom des Messwertes ist oft hinreichend groß gegenüber den Störströmen durch kapazitive Kopplung (z. B. 50-Hz-Brummen). Induktive Einkoppelungen haben prinzipiell keinen Störeinfluss mehr.

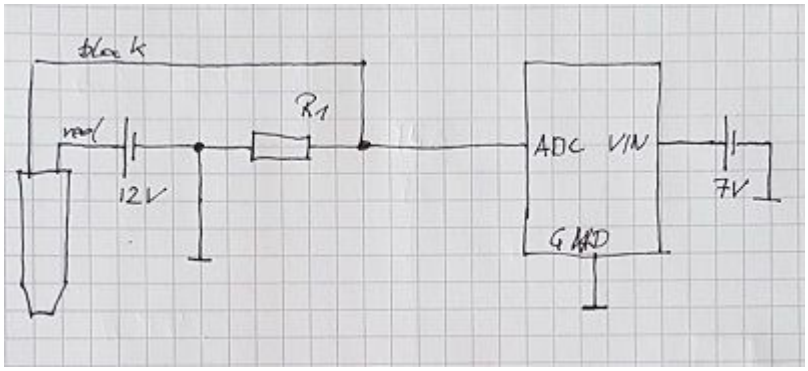
Wird eine Stromschnittstelle verwendet, die dem Standard 4…20 mA entspricht, lassen sich noch weitere Vorteile erreichen:

- *Mit der Mindest-Stromstärke von 4 mA lässt sich der Messwertgeber (Sensor) versorgen und es ist nur eine Hin- und Rückleitung erforderlich.*
- *Kabelbruch (Strom wird < 4 mA) lässt sich detektieren.*

Beispiel 1: ein 4mA-20mA/12V Sensor für einen 5V Arduino Uno

Wie bekommt man die Messwerte vom 4mA-20mA/12V Sensor in einen Arduino Uno?

Der Sensor wird mit +12V versorgt, der zweite Anschluss geht zunächst an den Analogeingang und mit einem Messwiderstand an GND. Die Basisschaltung sieht wie folgt aus:



Den Messwiderstand (R1) bemessen wir so, dass bei dem maximalen Strom von 20mA die zu messende Spannung unter Vref bzw VCC liegt.

Wegen U = I * R erhalten wir
5V / 0,020A = 250 Ohm

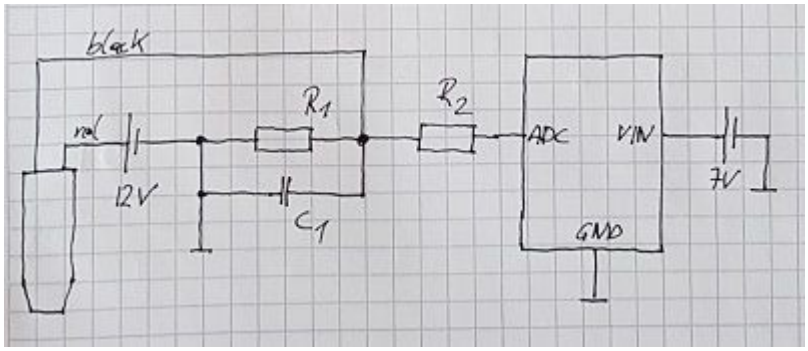
Wir nehmen den nächst kleineren Widerstand von 220 Ohm und ermitteln den Spannungsbereich der am ADC des Microcontrollers ankommen wird:

bei 0,004A * 220Ohm = 0,88V
bei 0,020A * 220Ohm = 4,4V

Der Arduino wird je nach Sensorwert am 10bit Analog-Eingang einen Wert von etwa 180 bis 901 messen.

Diese Messwerte müssen wir dann noch in die jeweilige Messeinheit umrechnen.

Wenn man in einer Schaltung Spannungen verwendet die höher als die 5 V des Arduino sind, empfiehlt es sich, den Eingang zusätzlich abzusichern. Das hilft nicht nur in der Eile des Gefechtes, wenn man irrtümlich falsche Kabel verbindet, sondern auch bei einem ungewolltem Kabelbuch - oder schlimmer - bei einem Kurzschluss in der Sensorleitung.



Wir wollen den Arduino gegen Spannungen die größer als VCC + 0,6V sind schützen und den Strom auf unter 0,1mA bringen. In unserem Fall also
12V - (5V + 0,6V) = 6,4V
6.4V / 0,0009A = 7111 Ohm.
Wir nehmen also für R2 einen 8,2K Widerstand.

Parallel zum Messwiderstand R1 können wir noch einen 100nF Kondensator (C1) zum Glätten setzen.

Beispiel 2: ein 4mA-20mA/24V Sensor für einen 3.3V Arduino Pro Mini

Die im ersten Beispiel ermittelten 4,4V wären für einen 3,3V Arduino Pro Mini zu hoch. Also rechnen wir wieder:

3.3V / 0,020A = 165 Ohm.

Für einen 3,3V MicroController wählen wir daher einen Widerstand (R1) von 150 Ohm.

V = 0,004 * 150 = 0,6V (ADC misst 186)
V = 0,020 * 150 = 3.0V (ADC misst 931)

Der Schutzwiderstand (R2) soll in diesem Falle
24V - (3,3V + 0,6V) = 20,1V
20,1V / 0,0009A = 22333 Ohm
groß sein, ein 27K Widerstand darf es dann schon sein.

Eine Arduino Library für Sensoren mit Stromschleifenschnittstelle

Eigentlich braucht man nicht wirklich eine Library, aber es mach die Arbeit einfacher.

Am Ende der Seite steht ein Download zur Verfügung. Zum Anfangen probierst du am besten das Hello World Beispiel aus.

Erster Schritt ist die Library einzubinden:

```
#include <NoiascaCurrentLoop.h> // download library from ht
```

Die Parameter festlegen:

```
const byte sensorPin = A2;      // ADC pin for the sensor
const uint16_t resistor = 150;  // used pull down resistor
const byte vref = 50;           // VREF in Volt*10 (Uno 16M
```

Hinweis: ein Arduino Uno der nur an USB hängt, hat selten eine VREF von exakt 5 Volt. Wenn also keine stabilen Werte zustande kommen, dann mal die Spannung an VREF messen und evtl. nur 48 oder 47 eintragen.

Anschließend ein Objekt erstellen:

```
CurrentLoopSensor currentLoopSensor(sensorPin, resistor, vr
```

In das Setup kommt noch eine begin Methode:

```
currentLoopSensor.begin(); // start the sensor
```

Die Library hat auch einen eigenen Check, der die erfassten Parameter auf Plausibilität überprüft und beispielsweise vor falschen Widerstandswerten warnt. Üblicherweise braucht man diesen Check nur beim ersten Sketch und kann im laufenden Betrieb auskommentiert werden:

```
currentLoopSensor.check(); // remove this line if check sho
```

Der Sensor wird mit folgender Methode ausgelesen:

```
 int myValue = currentLoopSensor.getValue(); // read sensor
```

Intern führt die Library mehrere Messungen durch und mittelt die Werte. Das geschieht aber dennoch "schnell" - und dauert etwa 150ms.

Sollte für anfängliche Testausgaben auch der reine Wert aus den ADC Messungen gewünscht sein, so kann man diesen mit einer eigenen Methode abfragen:

```
currentLoopSensor.getAdc(); // get the raw ADC value from t
```

Dabei handelt es sich um den zwischengespeicherten Wert der letzten Ermittlung.

Hintergründe zur "Noiasca Current Loop" Library

Pro .getValue() wird der Analogeingang mehrmals gemessen und das arithmetische Mittel gezogen.

Der Messbereich ist immer von 4mA - 20mA. 4mA (oder weniger) wird als 0 interpretiert, 20mA (oder mehr) ergibt das was du im Konstruktor als maxValue übergeben hast.

Die Berechnung des Ausgabewertes erfolgt im wesentlichen mit 3 Berechnungen:

```
minAdc = (0.004 * resistor * 1024 / (vref / 10.0));
maxAdc = (0.020 * resistor * 1024 / (vref / 10.0));
int32_t value = (adc - minAdc) * int32_t(maxValue) / (maxAd
```

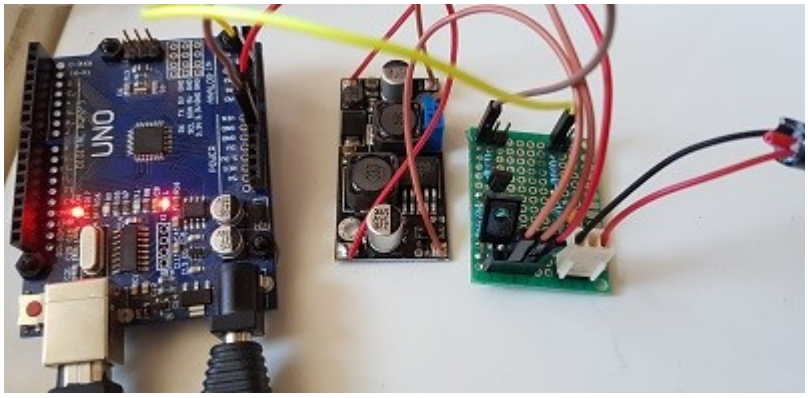
Mein Usecase: Wasserpegel Messung mit dem Arduino

Aktuell messe ich an einem Wasserlauf die Wassertiefe mit einem 4-20mA Transducer.

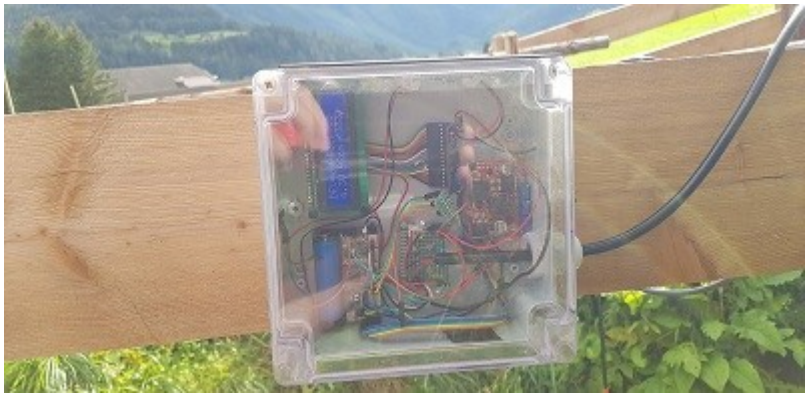
Diese Bauartart von Sensoren eignen sich zur

Wasserstandsmessung in engen Schächten, Brunnen oder Probebohrungen. Die Erfassung des Wasserpegels erfolgt über die Messung des Wasserdrucks. Pro Meter Eintauchtiefe entsteht ein Druck von etwa 9,807 kPa. Durch das Kabel des Sensor wird auch ein Luftschlauch geführt. Damit kann der Differenzdruck zwischen dem Wasserdruck und dem aktuellen Luftdruck ermittelt werden und über die 4-20mA Schnittstelle übertragen werden.

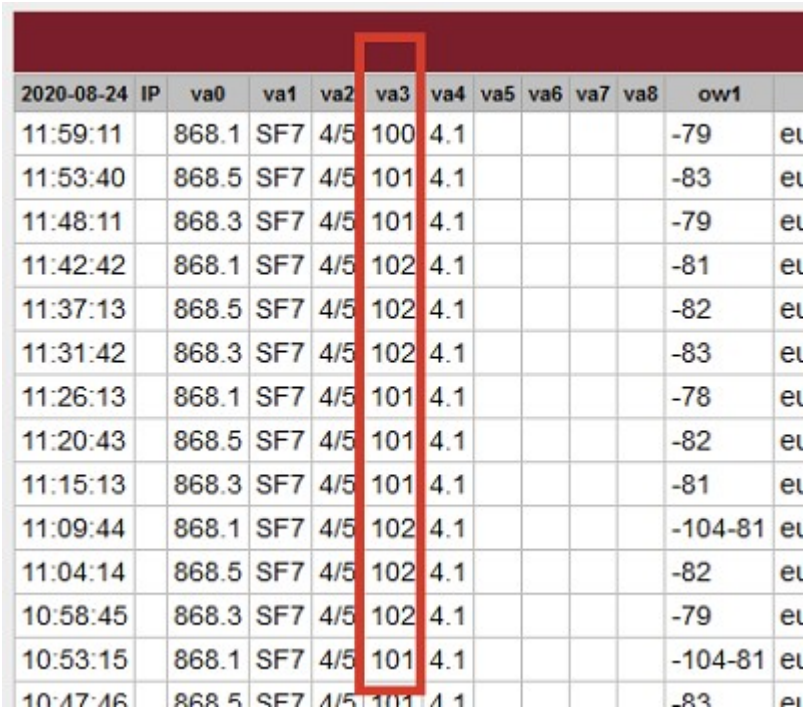
Zur Auswertung reicht ein Atmega328.



Im konkreten Fall verwende ich einen Akku/Solar betriebenen Lora Radio Node und übertrage den Messwert mittels LoraWAN an TheThingsNetwork.



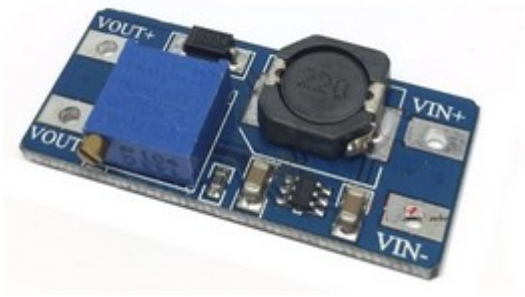
In der Folge erhalte ich die Messwerte via internet:



2020-08-24	IP	va0	va1	va2	va3	va4	va5	va6	va7	va8	ow1	
11:59:11		868.1	SF7	4/5	100	4.1					-79	eu
11:53:40		868.5	SF7	4/5	101	4.1					-83	eu
11:48:11		868.3	SF7	4/5	101	4.1					-79	eu
11:42:42		868.1	SF7	4/5	102	4.1					-81	eu
11:37:13		868.5	SF7	4/5	102	4.1					-82	eu
11:31:42		868.3	SF7	4/5	102	4.1					-83	eu
11:26:13		868.1	SF7	4/5	101	4.1					-78	eu
11:20:43		868.5	SF7	4/5	101	4.1					-82	eu
11:15:13		868.3	SF7	4/5	101	4.1					-81	eu
11:09:44		868.1	SF7	4/5	102	4.1					-104-81	eu
11:04:14		868.5	SF7	4/5	102	4.1					-82	eu
10:58:45		868.3	SF7	4/5	102	4.1					-79	eu
10:53:15		868.1	SF7	4/5	101	4.1					-104-81	eu
10:47:46		868.5	SF7	4/5	101	4.1					-83	eu

Exkurs von 3.3 auf 12V:

Mein 4-20mA Sensor benötigt zum stabilen Betrieb eine Spannung von 12-24V. Das spießt sich leider mit dem Batteriebetrieb durch 3.7V LION Akkus. Eine Lösung ist die Verwendung von Step-Up Wandlern. Du musst jedoch darauf achten, dass der Step-Up Wandler mit deiner Eingangsspannung stabil funktioniert. Konkret, die üblichen Step Up Wandler mit einem stehenden blauen Spindelpoti haben bei mir nicht funktioniert. Ich schalte den Step-Up Wandler vor der Messung über eine Transistorschaltung ein - führe die Messung durch und schalte anschließend den Step-Up Wandler wieder ab.



Arduino IDE 1.8.13

Diese Library ist unter Arduino IDE 1.8.13 entstanden und sollte auch mit späteren Versionen kompatibel sein.

Zusammenfassung

Die Auswertung von 4-20mA Sensoren klappt auch mit dem Arduino UNO.