



Entwicklung einer elektronischen Nase

Mikhail Krigman

235598

mikhail.krigman@tu-dortmund.de

März 2024

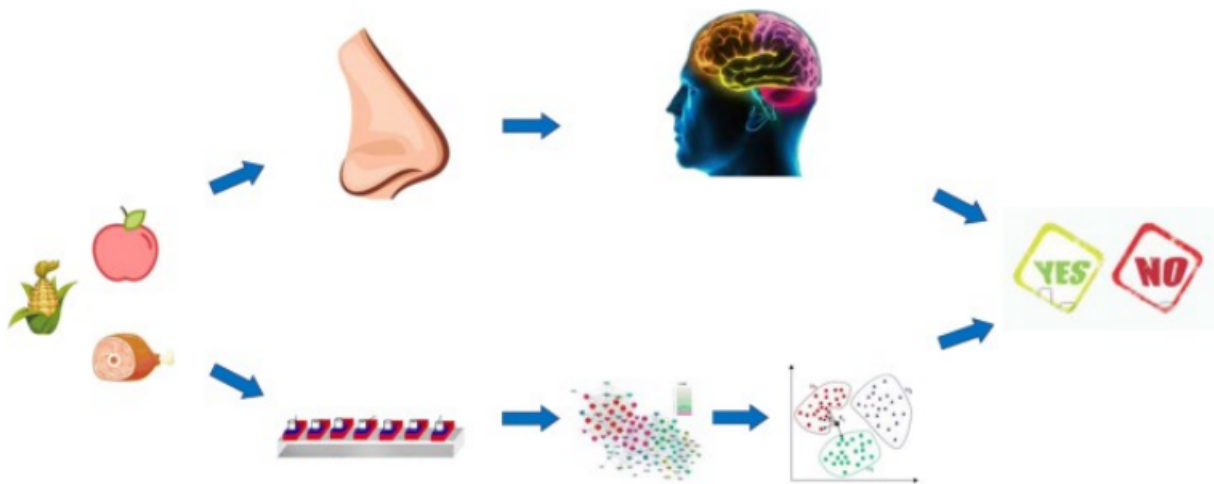
Inhaltsverzeichnis

1	Einleitung	3
2	Theoretische Grundlagen	4
2.1	Sensoraufbau	4
2.2	Spannungsgesteuerte Stromquelle	4
2.3	Messung der Heizerspannung	5
2.4	Sensortemperaturregelung	6
2.5	PID-Regler	7
2.6	Time-To-Digital Verfahren	7
2.7	Versorgung von Arduino mit dem Sensor-Netzteil	9
3	Praktikumsdurchführung	10
3.1	Dimensionierung der Time-To-Digital Schaltung	10
3.2	Genauigkeit von Time-To-Digital Verfahren	10
3.3	Virtuelles Sensorarray	11
3.4	Einstellung des PID-Reglers	12
3.4.1	Interpretation des PID-Regler Outputs	12
3.4.2	Dimensionierung der Reglerparameter	12
3.5	Reale Schaltung	13
3.6	Messung	14
3.6.1	Allgemeines Vorgehen	14
3.6.2	Genauigkeit der Sensordaten bei verschiedenen Temperaturen	16
4	Zusammenfassung	18
A	Time-To-Digital Messwerte	19
	Literatur	21

1 Einleitung

Die Entwicklung von Gassensoren spielt eine entscheidende Rolle für viele Bereiche unserer modernen Gesellschaft, von industriellen Prozessen bis hin zur Überwachung der Umweltqualität. Die Gassensoren finden ihre Anwendung überall - In industriellen Anlagen, um die Konzentration verschiedener Gase zu überwachen; sie detektieren Schadstoffe wie Kohlenmonoxid (CO), Stickstoffdioxid (NO₂), Schwefeldioxid (SO₂) und Ozon (O₃) und tragen dazu bei, Umweltverschmutzung zu messen und zu kontrollieren; sie dienen der frühzeitigen Warnung vor Brandgefahrensituationen und dem Schutz von Leben und Eigentum; in der Medizin werden Gassensoren in Atemgasanalysegeräten eingesetzt, um verschiedene Stoffe in Atemluft zu messen u.s.w.

Als kritische Komponenten für die Detektion und Messung von Gasen tragen sie wesentlich zur Sicherheit, Effizienz und Gesundheit bei.



Im Rahmen des Praktikums wird so eine "Elektronische Nase" entwickelt. Dabei werden sowohl theoretische als auch praktische Aspekte dieser Technologien behandelt. Das Praktikum bietet die Gelegenheit, tiefergehende Einblicke in die Funktionsweise, Einstellung und Anwendungsmöglichkeiten von Gassensoren zu gewinnen.

In diesem Bericht werden zunächst grundlegende Konzepte und Funktionsprinzipien von Gassensoren erläutert und auf einem praktischen Beispiel demonstriert.

2 Theoretische Grundlagen

2.1 Sensoraufbau

Um eine elektronische Nase zu realisieren, benutzt man einen MOX-Sensorchip. Der besteht aus einem Heizer und einer Metalloxidschicht.

Metalloxidschicht ist ein n-dotierter gessensiver Halbleiter. Durch die Wechselwirkung zwischen dem Gas und der Metalloxidschicht ändert sich die Leitfähigkeit des Halbleiters, die sich von außen zu messen lässt. Dadurch kann man das Entstehen anderer Gasen in der Luft merken. Nachteil von solchen Metalloxidschichten ist, dass sie stark temperaturabhängig sind. Um den Temperatureinfluss zu nivellieren ist im sensor ein Heizer eingebaut, der immer gleiche Temperatur für den Sensor einstellen soll.

Der Heizer stellt von sich selbst einfacher Widerstand dar, der elektrische Leistung laut dem Joul'schen Gesetz (1) in die Wärme umwandelt.

$$Q = R \cdot I^2 \cdot t \quad (1)$$

Der Widerstand des Heizers ist aber auch Temperaturabhängig:

$$R(T) = R_0 \cdot (1 + \alpha \cdot T) \quad (2)$$

Diese Eigenschaft ist für den Gassensor von Vorteil. Durch kontinuierliche Messung und entsprechende Anpassung des Heizerwiderstands ist es möglich die Temperatur des Sensors zu bestimmen und regeln. D.h. Heizer lässt sich als Temperatursensor und -regler zu benutzen.

2.2 Spannungsgesteuerte Stromquelle

In diesem Praktikum wird der Gassensor mit Arduino MKR Zero gesteuert. Dieser Mikrocontroller kann aber nur die gewünschte Spannung erzeugen und lässt sich direkt nur als Spannungsquelle zu nutzen. Um den Heizer zu steuern braucht man aber eine Stromquelle.

Dafür wird eine Hilfsschaltung mit einem Operationsverstärker gebaut, die auf der Abbildung (1) dargestellt ist.

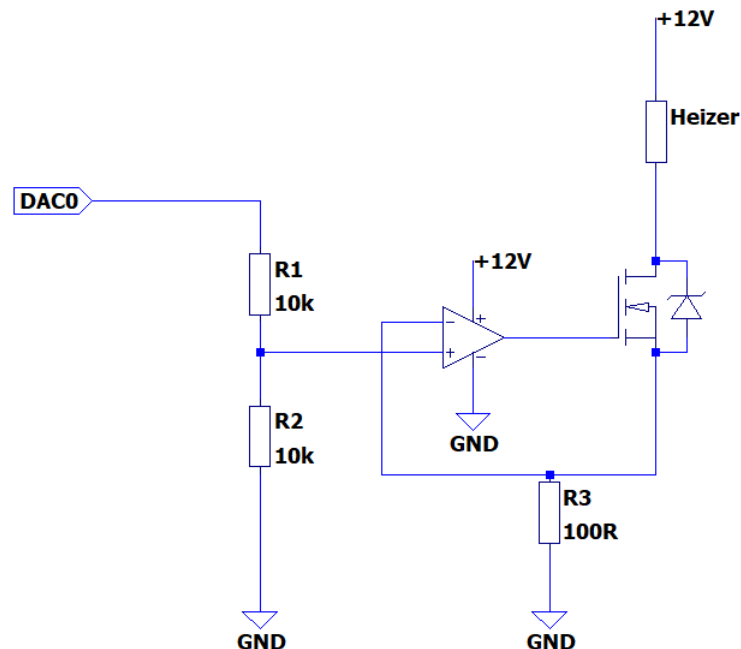


Abbildung 1: Spannungsgesteuerte Stromquelle für Heizerstromregelung

Das ist eine Spannungsgesteuerte Stromquelle. An DAC0 wird vom Arduino erzeugte Spannung von bis zu 3,3V angelegt, die schon einen relativ hohen Heizerstrom aufrufen kann. Um den Sensor zu sichern und dadurch fließenden Strom zu begrenzen, waren in die Schaltung ein Schuntwiderstand R_3 und ein Spannungsteiler zusätzlich hinzugefügt, der mit $R_1 = R_2$ die Eingangsspannung halbiert.

Mit Eigenschaften des Operationsverstärkers ($U_D \rightarrow 0$ und $Z_{in} \rightarrow \infty$) ergibt sich die Formel für den Heizerstrom in Abhängigkeit von der Eingangsspannung U_{DAC0} :

$$I_{Heizer} = \frac{U_{DAC0}}{2 \cdot R_3} \quad (3)$$

2.3 Messung der Heizerspannung

Um die Sensortemperatur zu regeln, ist es notwendig immer den Heizerwiderstand zu wissen und entsprechend anpassen. Der im Praktikum verwendete Mikrocontroller Arduino ermöglicht die Messung der Spannung an einem der analogen Eingängen. Hier sind aber 2 Punkte zu beachten:

1. Eingang des Arduinos darf nicht mehr als 3,3V annehmen, sonst wird die Platine kaputt.
2. Mit der spannungsgesteuerten Stromquelle, die im Abschnitt (2.2) beschrieben war, ist es die Heizerspannung direkt zu messen nicht möglich, weil der Widerstand R_3 vor dem Heizer und die Betriebsspannung +12V nach dem Heizer stehen. D.h. um die Heizerspannung zu messen, muss die Differenz dieser beiden Potentialen berechnet werden.

Beide Probleme löst die in der Abbildung (2) dargestellte Schaltung.

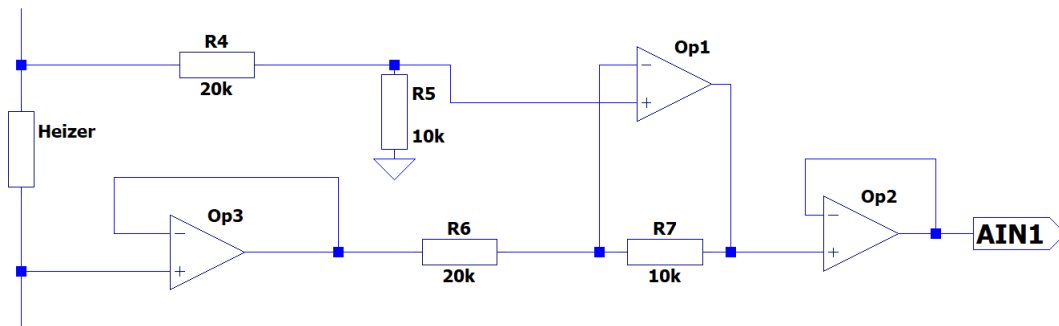


Abbildung 2: Schaltung für Messung der Heizerspannung

Der Operationsverstärker $Op1$ und die Widerstände R_4, R_5, R_6 und R_7 bilden zusammen die Subtrahierschaltung und ergeben entsprechend die Potentialdifferenz beider Enden des Heizers multipliziert mit dem Vorfaktor $\frac{R_7}{R_6}$, falls $R_4 = R_6$ und $R_5 = R_7$. Die Ausgangsspannung des Operationsverstärkers wird dann mit folgender Formel ermittelt:

$$U_{Aus} = \frac{R_7}{R_6} \cdot U_{Heizer} \quad (4)$$

Mit $R_7 = 10k\Omega$ und $R_6 = 20k\Omega$ wird die Heizerspannung halbiert und demzufolge Arduino-Eingangsspannung von der hohen Spannungen gesichert.

Die Weiterleitung des Ausgangssignals vom Operationsverstärker $Op1$ direkt zum Arduino kann zu verfälschten Messungen führen, weil der Messsignal durch die Eingangsimpedanz des Arduinos beeinträchtigt werden kann. Um mögliche Störungen zu vermeiden, wird vor dem Arduino-Eingangsspannung ein Impedanzwandler hinzugefügt, der mit dem Operationsverstärker $Op2$ realisiert ist. Da Operationsverstärker sehr geringe Ausgangsimpedanz und Mikrocontroller umgekehrt sehr hohe Eingangsimpedanz hat, kann der Signal schon nicht mehr vom Arduino beeinflusst werden.

Es ist noch zu beachten, dass diese Schaltung zusammen mit der Schaltung (1) benutzt wird. Die früher erwähnte spannungsgesteuerte Stromquelle beinhaltet einen MOSFET-Transistor, der durch sein Öffnen und Schließen die Spannungsschwankungen erzeugen kann. Damit diese Schwankungen die Spannungsmessung nicht bewirken, wird vor der Subtrahierschaltung noch ein Spannungsfolger $Op3$ verwendet.

2.4 Sensortemperaturregelung

Die aus der spannungsgesteuerten Stromquelle (1) und dem Heizerspannungsmesser (2) zusammengesetzte Schaltung (3) ermöglicht die Messung der halbierten Heizerspannung auf dem Eingang [AIN1]¹ und Steuerung des durch den Heizer fließenden Stroms über den Ausgang [DAC0]¹

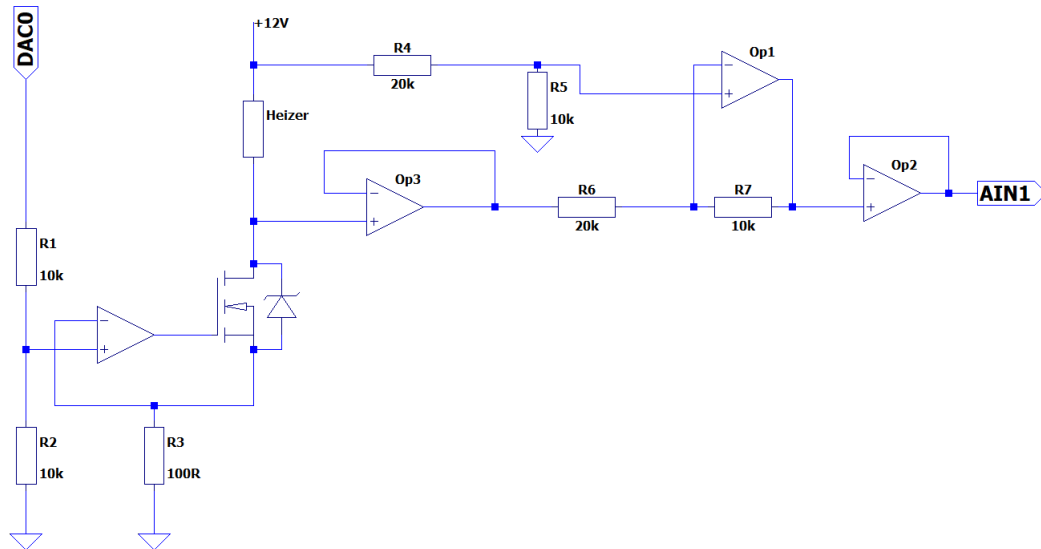


Abbildung 3: Schaltung für Sensortemperaturregelung

Es ist nicht ausreichend, einmal den Heizerstrom einzustellen. Da der Heizerwiderstand mit dem konstanten Strom sich kontinuierlich erwärmt, wird er laut der Formel (2) mit der Zeit steigen. Sensor kann aber nur dann korrekte Daten liefern, wenn seine Temperatur stetig unverändert bleibt und demzufolge der Heizerwiderstand seinen Wert konstant hält. Um dieses Verhalten zu erreichen, ist es wie folgt vorzugehen:

1. Mit dem Mikrocontroller die Heizerspannung messen
2. Mit im Voraus bekannten Steuerspannung² den Heizerstrom mit der Formel (3) berechnen
3. Aus Heizerspannung und -strom den Heizerwiderstand ermitteln
4. Die Steuerspannung entsprechend anpassen

Das gesamte Vorgehen ist auf dem Bild (4) auch veranschaulicht.

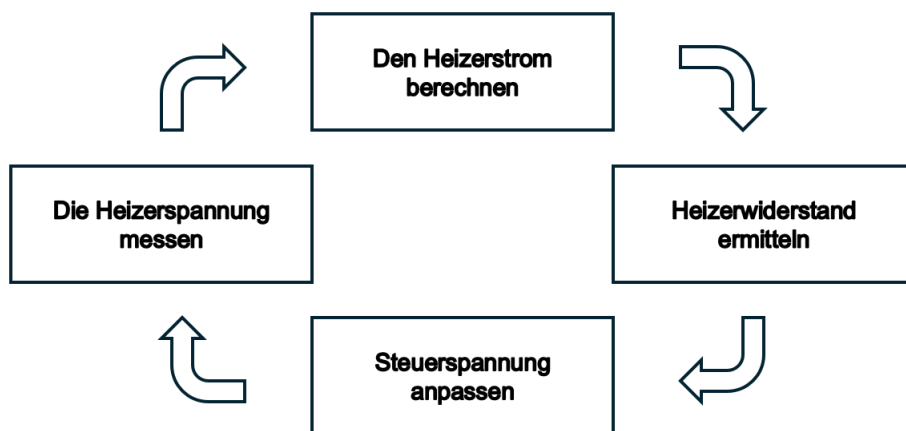


Abbildung 4: Algorithmus für Sensortemperaturregelung

¹Tatsächlich können beliebige analoge Pins gewählt werden

²Die Steuerspannung ist immer bekannt, weil die vom Entwickler einzustellen ist.

2.5 PID-Regler

Um die Steuerspannung richtig anzupassen, wird ein PID-Regler verwendet.

Der PID-Regler, kurz für Proportional-Integral-Derivative-Regler, ist eine Rückkopplungsregelschleife, die in verschiedenen Bereichen verwendet wird, um die Ausgangsgröße eines Systems so zu steuern, dass sie sich dem gewünschten Sollwert annähert.

PID-Regler basiert auf drei grundlegenden Steuerungskomponenten:

1. **Proportionalkomponente (P)**: Diese Komponente erzeugt eine Ausgangsänderung, die proportional zum Fehler zwischen dem Ist- und dem Sollwert ist. Je größer der Fehler, desto größer ist der Beitrag der Proportionalkomponente.
2. **Integral-Komponente (I)**: Diese Komponente trägt zur Ausgangsänderung bei, basierend auf der Summe der vergangenen Fehler über die Zeit. Sie hilft, langfristige Drifts oder konstante Fehler zu eliminieren und das System genauer auf den Sollwert einzustellen.
3. **Differential-Komponente (D)**: Diese Komponente reagiert auf die Geschwindigkeit, mit der sich der Fehler ändert. Sie hilft, übermäßige Oszillationen zu reduzieren und das System schneller auf den Sollwert einzustellen.

Die Gesamtausgangsänderung des PID-Reglers ergibt sich aus der Summe dieser drei Komponenten:

$$y(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de}{dt} \quad (5)$$

Dabei sind K_p , K_i und K_d die Proportional-, Integral- und Differentialverstärkungen, die durch den Anwender entsprechend den Systemanforderungen eingestellt werden können.

Die Reglerverstärkungen werden am häufigsten experimentell bestimmt, es existiert aber die Methode von Ziegler und Nichols[4], mit der die optimale Parameterwerte ermittelt werden können. Das Verfahren basiert darauf, dass erst die kritische Proportionalverstärkung $K_{p,krit}$ und Periode der Dauerschwingung T_{krit} gefunden werden und danach davon abhängig die restliche Glieder berechnet. Die Formeln für jeweiligen Glied sind in der Tabelle (1) aufgeführt.

Voraussetzung	Regler	Reglerparameter
Kritische Verstärkung und Periodendauer bekannt	P	$K_p = 0,5 \cdot K_{p,krit}$
	PI	$K_p = 0,45 \cdot K_{p,krit}, T_n = 0,85 \cdot T_{krit}$
	PD	$K_p = 0,55 \cdot K_{p,krit}, T_v = 0,15 \cdot T_{krit}$
	PID	$K_p = 0,6 \cdot K_{p,krit}, T_n = 0,5 \cdot T_{krit}, T_v = 0,125 \cdot T_{krit}$

Tabelle 1: Optimale Reglerparameter laut Ziegler und Nichols Verfahren

Hierbei ist $K_i = \frac{K_p}{T_n}$ und $K_d = K_p \cdot T_v$

2.6 Time-To-Digital Verfahren

Solange die Sensortemperatur konstant bleibt, kann man die Sensordaten messen und erfolgreich interpretieren. Wie im Abschnitt (2.1) erwähnt war, ändert der Sensor seinen Widerstand in Abhängigkeit von der Konzentration der verschiedenen Gasen in der Luft. D.h. die Daten, die vom Sensor abgelesen und ausgewertet werden müssen, sind nur sein Widerstand.

Der Mikrocontroller hat aber kein integrierten Ohmmeter oder anderen Messgerät, der den Widerstand messen kann. Deswegen wird "Time-To-Digital" Verfahren zur Bestimmung des Sensorwiderstands genutzt.

Die Methode besteht darin, die Zeit zu messen, die ein Kondensator benötigt, um sich über einen bestimmten Widerstand zu entladen und Ergebnis mit dem Referenzwert zu vergleichen. Dafür ist aber noch eine Schaltung notwendig, die auf der Abbildung (5) gezeichnet ist.

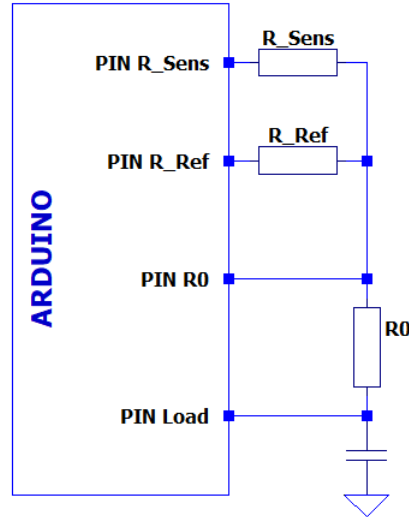


Abbildung 5: Schaltung für Implementierung der Time-To-Digital Methode

Der Mikrocontroller hat sehr hohe Eingangsimpedanz und geringe Ausgangsimpedanz, deshalb ist es sehr einfach den Schaltungsstromfluss zu steuern. Um den Kondensator zu laden müssen alle Pins außer dem PIN Load als Eingang definiert werden. Der PIN Load wird für Ausgang gesetzt, um den Strom dem Kondensator zu liefern. Sobald der Kondensator geladen ist, werden PIN Load, PIN R0 und PIN R_Sens als Eingang definiert, während PIN R_Ref als Ausgang, damit der ganze Strom durch die Widerstände R0 und R_Ref fließt und der Kondensator durch diese Widerstände sich entsprechend entladet. Über R0³ ist es möglich zu merken, wenn der Kondensator leer wird. Mit Hilfe der Arduino-Funktion *micros()* kann die Entladezeit gemessen werden. Danach werden die gleiche Schritte für R_Sens durchgeführt und im Endeffekt 2 gemessene Entladezeiten zur Verfügung gestellt: $T_{ent,R_{Ref}}$ und $T_{ent,R_{Sens}}$.

Die Formel für die Entladezeit des Kondensators lautet:

$$T_{ent} = R \cdot C \cdot \ln \frac{V_i}{V_{TL}} \quad (6)$$

Bildet man das Verhältnis der beiden gemessenen Zeiten, bekommt man die Formel für Bestimmung des Sensorwiderstands in Abhängigkeit nur von der R_{Ref} , $T_{ent,R_{Ref}}$ und $T_{ent,R_{Sens}}$:

$$R_{Sens} = R_{Ref} \cdot \frac{T_{ent,R_{Sens}}}{T_{ent,R_{Ref}}} \quad (7)$$

³Da PIN R0 als Eingang definiert ist, lässt der zu prüfen, ob an dem Pin die Spannung vorhanden ist oder nicht. Mit dem Pin PIN R_Ref bzw. PIN R_Sens ist es unmöglich, weil die zu dem Zeitpunkt als Ausgangspins betrieben sind

2.7 Versorgung von Arduino mit dem Sensor-Netzteil

Der Gassensor und für die Schaltung (3) verwendete Operationsverstärker müssen mit 12V versorgt werden. Der Mikrocontroller Arduino darf aber maximal mit 5V versorgt werden. Dafür wird noch eine Schaltung mit dem Festspannungsregler (6) in die elektronische Nase eingebaut.

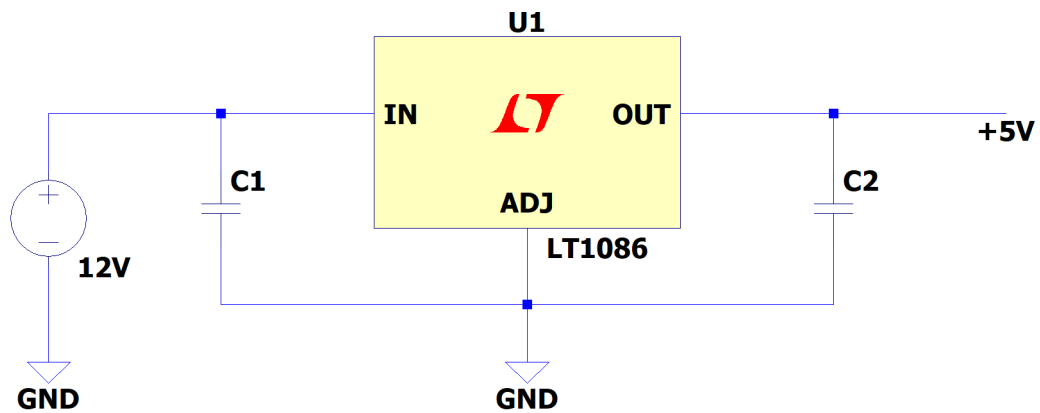


Abbildung 6: Beschaltung des 5V Festspannungsreglers

Die Kondensatoren C_1 und C_2 sind hier eingebaut, um Spannungsschwankungen auszugleichen und Stabilität der Versorgung zu verbessern.

3 Praktikumsdurchführung

3.1 Dimensionierung der Time-To-Digital Schaltung

Für die Schaltung (5) wird $R_{Ref} = 10k\Omega$ eingesetzt. Es bleibt noch die Kapazität zu dimensionieren. Um passenden Kondensator auszuwählen werden verschiedene Kapazitäten in die Formel (6) eingesetzt und die Entladezeit berechnet. Die Eingangs- und Threshold-Spannung werden aus dem Arduino-Mikrocontroller Datenblatt übernommen: $V_i = 0,9 \cdot 3,3V = 2,97V$ und $V_{TL} = 0,3 \cdot 3,3 = 0,99V$

Einige berchnete Werte sind in der Tabelle (2)

Entladezeit [s]	Kapazität [μF]
5,493	500
2,197	200
1,098	100
0,110	10
0,011	1

Tabelle 2: Entladezeit bei verschiedenen Kondensatoren

Tatsächlich in der Schaltung kann beliebige Kapazität stehen, aber die Entladezeit bewirkt direkt die Programmlaufzeit. Deswegen ist es am besten den Kondensator mit der geringsten Kapazität zu nehmen. Aber die Entladezeit $T_{ent} = 0,011s$ ist für den Mikrocontroller schon wirklich klein, was die Abweichungen bei der Zeitmessung aufrufen kann. Um den Felher zu vermeiden, wird eine Kapazität $C = 10\mu F$ gewählt.

3.2 Genauigkeit von Time-To-Digital Verfahren

Um sicherzustellen, dass für Bestimmung des Sensorwiderstands gewählte Methode gut genug ist, werden 10 im Voraus bekannte Widerstände gemessen. Der Testwiderstand wird statt R_Sens in die Schaltung (5) eingesteckt, mit dem Time-To-Digital Verfahren 20 Mal gemessen und daraus der Durchschnitt berechnet. Die Ergebnisse sind auf dem Diagramm (7) veranschaulicht.

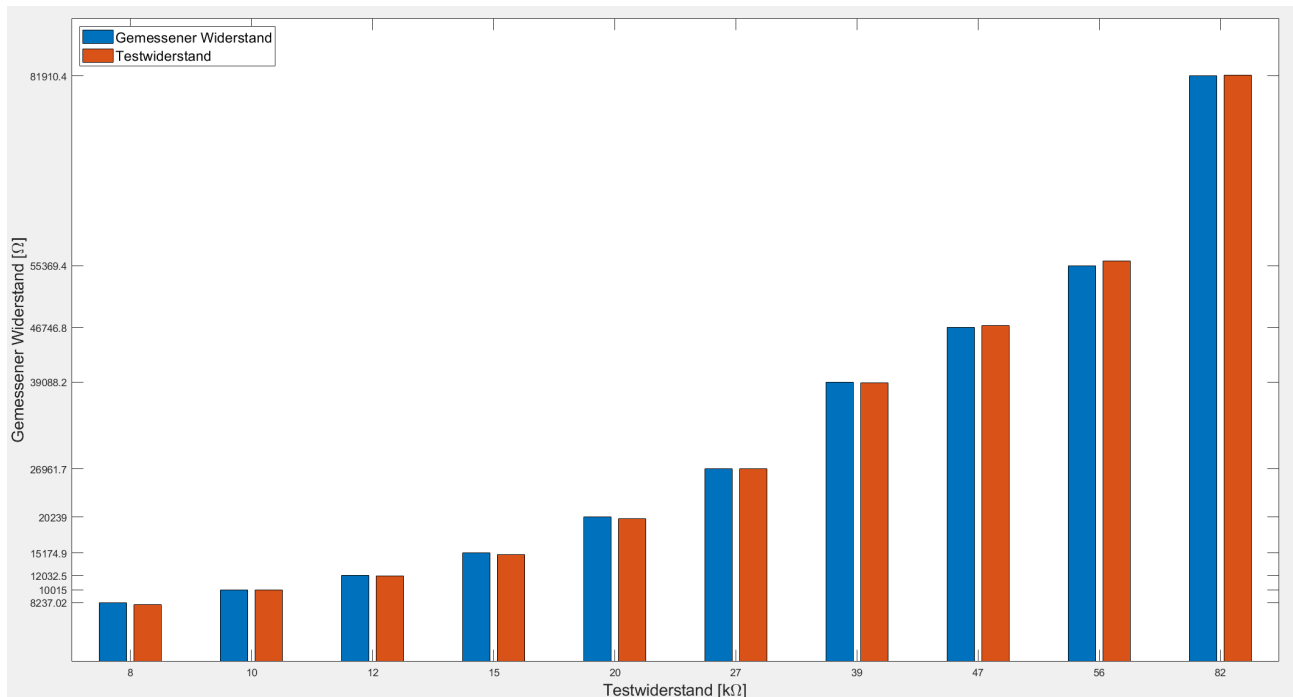


Abbildung 7: Visualisierung der Genauigkeit von Time-To-Digital Verfahren

Die gemessene Werte und berechnete Durchschnitte sind in der Tabellen (5), (6) und (7) zu finden.

Mit dem Formel (8) kann auch die Abweichung eingeschätzt werden. Der Messungsfehler für jeden Widerstand ist auch in der Tabelle (7) angegeben.

$$\delta = \frac{|R_{TTD} - R_{Test}|}{R_{Test}} \cdot 100\% \quad (8)$$

Mit der berechneten Abweichungen ist es zu zusammenfassen, dass das Time-To-Digital Verfahren einen durchschnittlichen Fehler 0.79% hat.

3.3 Virtuelles Sensorarray

Während des Praktikums wird Gassensor bei verschiedenen Temperaturen betrieben. Um dieses Verhalten zu erreichen müssen aber auch entsprechende Heizerwiderstände bekannt sein. Aus dem Datenblatten des Sensors ist es anzunehmen, dass der Sensor beim Raumtemperatur den 83Ω Widerstand besitzt. In der Praktikumsanleitung steht auch, dass bei $U_{Heizer} = 5V$ die $350^\circ C$ erreicht werden. Mit der Annahme, aus dem Datenblatt hergenommenen Heizerstrom($350^\circ C$) und Formeln (1) und (2) wird der Heizerwiderstand bei $350^\circ C$ berechnet: $R_{Heizer}(350^\circ C) = 120\Omega$

Für das Praktikum relevante Temperaturen sind: $150^\circ C$, $250^\circ C$ und $350^\circ C$. Um die entsprechende Widerstände zu ermitteln, wird aus der 2 schon bekannten Temperatur-Widerstand Punkten die Geradengleichung gebildet und mit deren Hilfe der nötige Heizerwiderstand gefunden. Die Abhängigkeit der Temperatur vom Widerstand lautet:

$$T_{Sens}[^\circ C](R_{Heizer}) = 8,92 \cdot R_{Heizer} - 720,27 \quad (9)$$

Die Gerade ist auch auf dem Graph (8) dargestellt.

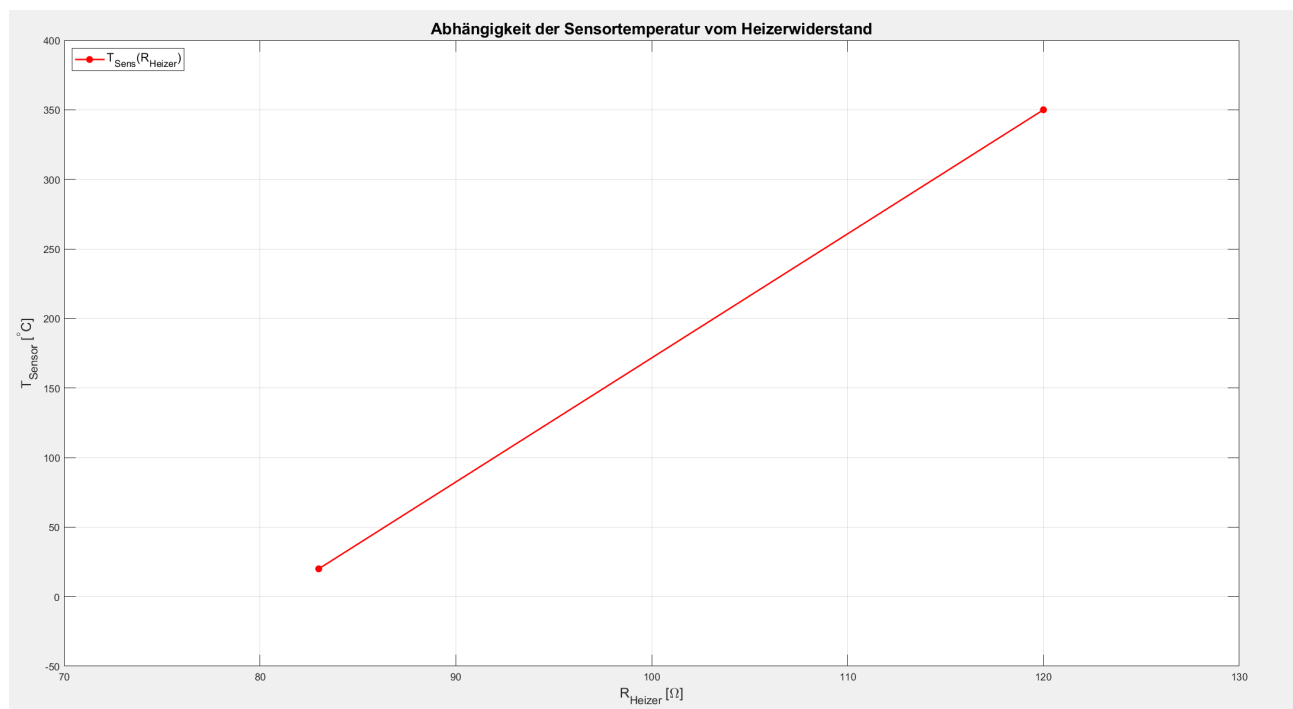


Abbildung 8: Abhängigkeit der Sensortemperatur vom Heizerwiderstand

Aus der Temperaturgleichung (9) bzw. aus dem Graph (8) ergeben sich die Heizerwiderstände für interessante Temperaturen:

Sensortemperatur $[^\circ C]$	Heizerwiderstand $[\Omega]$
150	98
250	109
350	120

Tabelle 3: Heizerwiderstände mit entsprechenden Sensortemperaturen

Die 3 betätigte Widerstände werden in ein virtuelles Sensorarray eingepackt. Das Ziel von dem Array ist den Sensor bei 3 verschiedenen Temperaturen zu betreiben. So ein Array aufzulegen bringt großen Vorteil für die Nutzung des Gassensors, weil der stark Temperaturabhängig ist und bei jeder Temperatur unterschiedliche Empfindlichkeit hat. Die zyklische Messung ermöglicht eine genauere Einschätzung der Sensordaten.

3.4 Einstellung des PID-Reglers

3.4.1 Interpretation des PID-Regler Outputs

Der PID-Regler liefert allgemein irgendeine Zahl, die als Steuersignal angenommen werden kann. Der Wert des Ausgangs hängt nur vom Systemfehler $e(t) = X(t) - X_{Soll}$ und der Reglerparameter. Dieses Verhalten erlaubt mit den Reglerverstärkungen K_p , K_i und K_d den Regler an jedes System anzupassen.

Um mit Arduino den Steuersignal zu senden, wird die Arduino-Funktion `analogWrite()` verwendet. Diese Funktion nimmt 2 Parameter an: Pin, an den der Signal gesendet werden soll, und den Arbeitszyklus des Signals, der die Werte zwischen 0 und 1023 annehmen kann. Demzufolge soll der PID-Regler die Werte zwischen 1 (0 führt zu mathematischen Erschwerungen in der Berechnung des Heizerwiderstands) und 1023 zurückgeben.

Für unsere Aufgaben reicht ein PI-Regler voll aus, weil Heizerwiderstand die Zeit zum Erwärmen bzw. zum Kühlen braucht und demzufolge wird D-Anteil kaum eine Rolle spielen.

Im Arduino Programm kann so ein PID-Regler wie folgt realisiert werden:

```
double unit2v(int unit){
    return (3.3 * ((1.0 * unit) / 1023));
}

double R_heater, R_heater_goal;
double Kp, Ki, PID_Output, integralSum = 0;
double U_heater, I_heater;
int control_voltage; // U_DAC in Arduino units for analogWrite()

void compute_control_voltage(){
    U_heater = 2.0 * unit2v(analogRead(PIN_AIN)); // 2*U_AIN
    I_heater = (1.0 * unit2v(control_voltage) / (2.0 * 100.0)); // R3 = 100 Ohm

    R_heater = (I_heater - 1e-4 > 0) ? U_heater / I_heater : 83;

    double error = R_heater_goal - R_heater;
    integralSum = constrain(integralSum + (double)(error) * Ki, 1, 1023);
    PID_output = constrain(Kp * error + integralSum, 1, 1023);

    control_voltage = PID_output;
    analogWrite(PIN_DAC, control_voltage);
}
```

3.4.2 Dimensionierung der Reglerparameter

Die Reglerparameter werden mit Hilfe der Methode von Ziegler und Nichols[4] dimensioniert. Der experimentell gefundene kritische Proportionalverstärkung $K_{p,krit}$ ist ungefähr 50 und Nachstellzeit T_n ungefähr 5s. Dann sind entsprechend $K_p = 25$ und $K_i = 10$.

Die Sprungantwort des PID-Reglers ist auf der Abbildung (9) dargestellt.

Aus dem Graph ist es zu merken, dass es eine kleine Schwingung gibt, die bei kleinen Widerständen merkbarer ist, als bei höheren. Es kann daran liegen, dass Heizerwiderstand bei niedrigen Temperaturen sich schneller erwärmt und kühlt und demzufolge ist es für den Regler komplizierter den stetigen Wert zu halten.

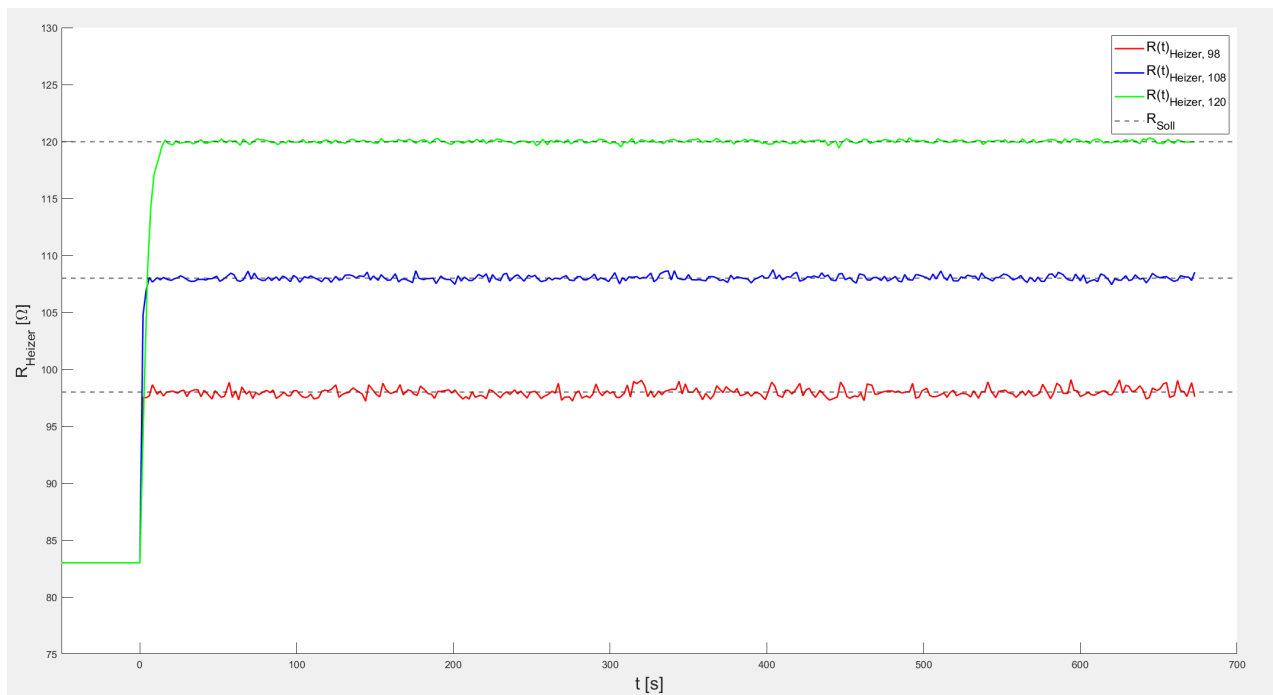


Abbildung 9: Sprungantwort des PID-Reglers für 3 Praktikumsrelevante Widerstände

3.5 Reale Schaltung

Um die elektronische Nase endlich fertigen. Müssen ganze Schaltungen auf der Reale Platine aufgebaut werden. Das gesamte Gerät ist auf der Abbildung (10) gezeigt.

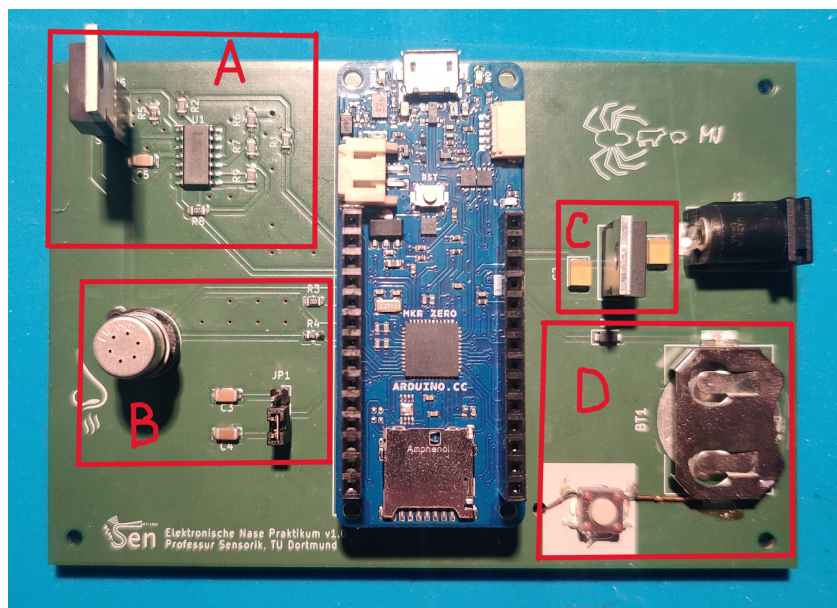


Abbildung 10: Reale Schaltung

- Bereich A: Schaltung für Sensortemperaturregelung(3)
- Bereich B: Schaltung für Time-To-Digital Verfahren(5)
- Bereich C: Festspannungsregler-Schaltung (6)
- Bereich D: Anschaltung der Knopfzelle für StandBy Modus

Die Knopfzelle und externe Batterie stehen hier, um Arduino in StandBy Modus umzuschalten und in dem zu betreiben. StandBy Modus - ist sogenannter Arduino-Sparmodus, in dem Arduino viel weniger Leistung verbraucht und viel längere autonome Arbeitszeit hat, als im einfachen Betrieb.

3.6 Messung

Die Messung hat um 15 Uhr angefangen und hat in einem Schlafzimmer 24 Stunden gedauert. Der Quellcode war so geschrieben, dass die Elektronische Nase jede Temperatur aus dem virtuellen Sensorarray für 15 Minuten haltet und die gemessene Werte auf die SD-Karte speichert. Danach wird die Schaltung 3 Minuten im „Stillstand“ betrieben, wobei nur PID-Regler betätigt ist, um neue Temperatur einzustellen.

Damit Praktikumsversuch veranschaulichender wird, werden in dem Zimmer verschiedene Umstände modelliert. Die Daten werden gemessen, wenn der Raum voll leer ist, wenn einige Personen da sind, wenn das Fenster geöffnet ist usw. Allgemein ist es zu sagen, dass die Schaltung wird so betrieben, dass möglichst viele äußere Faktoren die bewirken. Solche Vielfachheit von unterschiedlichen Umständen lässt die qualitative Konsequenzen über die Sensorgenauigkeit unter verschiedenen Bedingungen treffen und verbessert demzufolge die Endergebnisse des Praktikums.

Der gesamte Verlauf ist auf der Abbildung (11) gezeichnet.

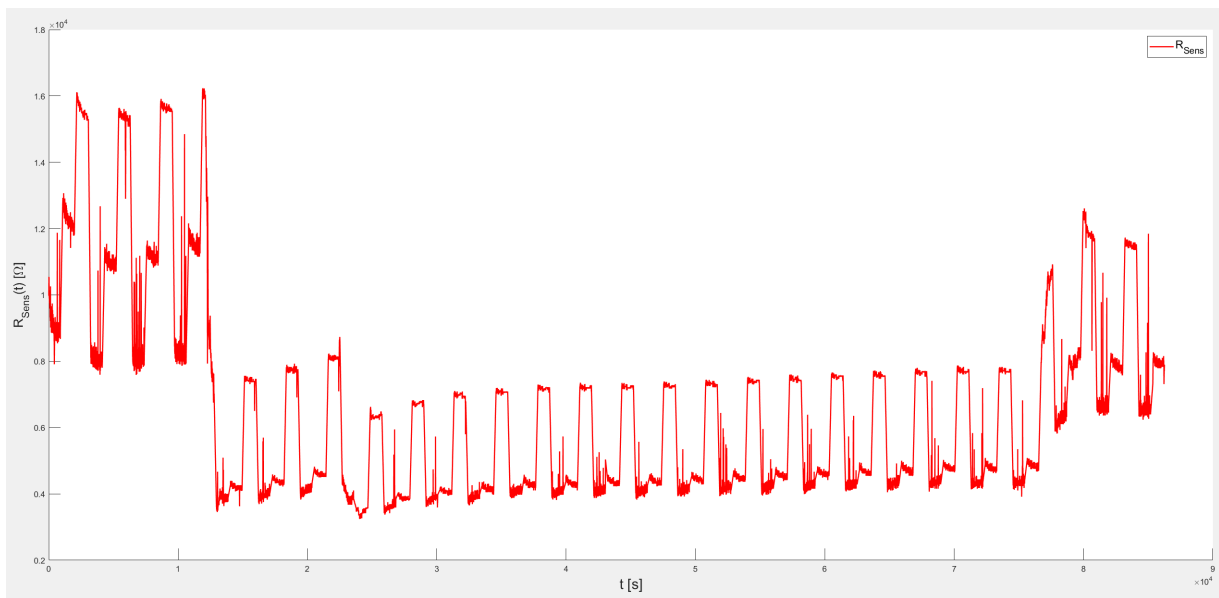


Abbildung 11: Gesamte Messung für 24 Stunden

Es ist hier ein Muster zu erkennen: die Kennlinie hat Form einer 3-stufigen Treppe, die sich zyklisch wiederholt. Jede Stufe gehört zu einer Temperatur und dauert 15 Minuten, wie es früher erwähnt war. Insgesamt nimmt ein Zyklus $3 \cdot (15 + 3) = 46$ Minuten Zeit.

Da der Widerstand mit der Temperatur steigt (2), kann man die Pegeln sofort einordnen:

1. (Niedrigste) Stufe - 150°C
2. (Mittlere) Stufe - 250°C
3. (Höchste) Stufe - 350°C

Diese Stufen sind auch zur Veranschaulichung auf der Abbildung (12) mit Pfeilen gezeigt

3.6.1 Allgemeines Vorgehen

Laut dem Datenblatt des Sensors [2] reagiert der auf Methane, CO_2 (Kohlenstoffdioxid), Iso-butane, Ethanol und Hydrogen. Besonders interessant ist dabei Kohlenstoffdioxid, weil der Gas vom Mensch abgeatmet wird und alle andere sind in einem einfachen Raum nicht zu treffen.

Gleichzeitig mit dem Messungsanfang wurde das Fenster im Zimmer zum Lüften geöffnet. Die Entlüftung speist immer frische Luft in den Raum ein und demzufolge sinkt die Konzentration des Kohlendioxids. Wie auf dem Sensordatenblatt zu sehen ist - je mehr CO_2 in der Luft, desto kleiner ist sein Widerstand.

Das Schließen des Fensters ist auch leicht erkennbar - das ist ein steiler Abfall des Sensorwiderstands nach dem 4. Zyklus (ungefähr 18:22). Gleichzeitig hat man auch auf den Sensor zusätzlich angehaucht, was das Sinken des Widerstands beschleunigt hat. Noch dazu hat eine Person neben dem Sensor telefoniert und entsprechend noch mehr Kohlenstoffdioxid erzeugt, was den Sensor bewirkt hat. Das ganze Vorgehen ist auf der Abbildung (13) veranschaulicht.

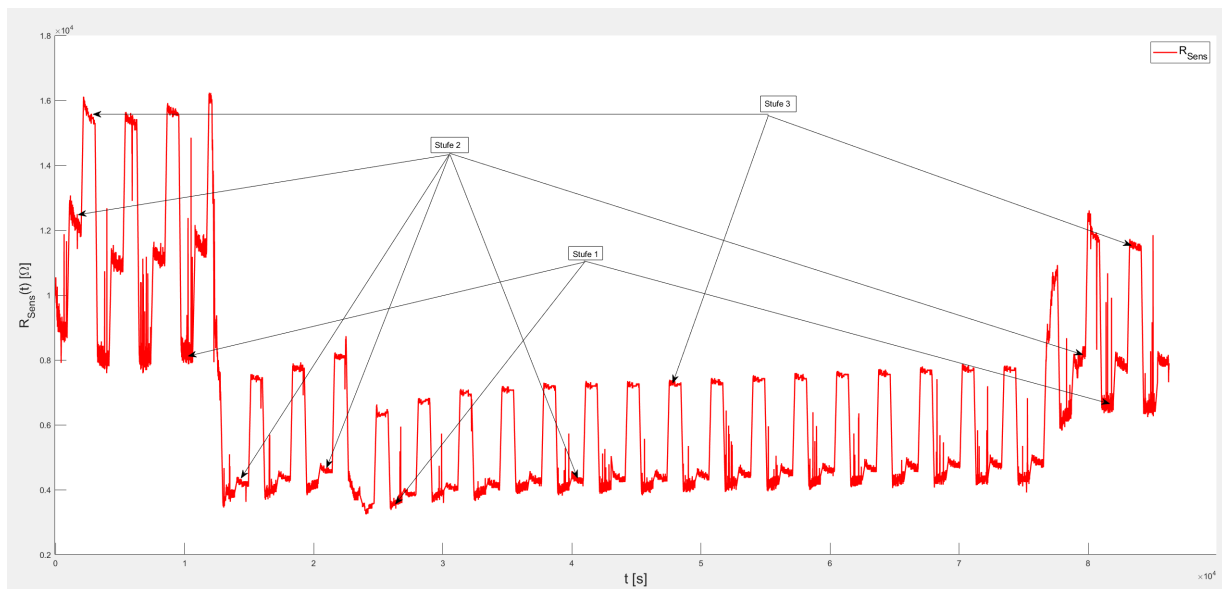


Abbildung 12: 3 Stufen im Messungsmuster

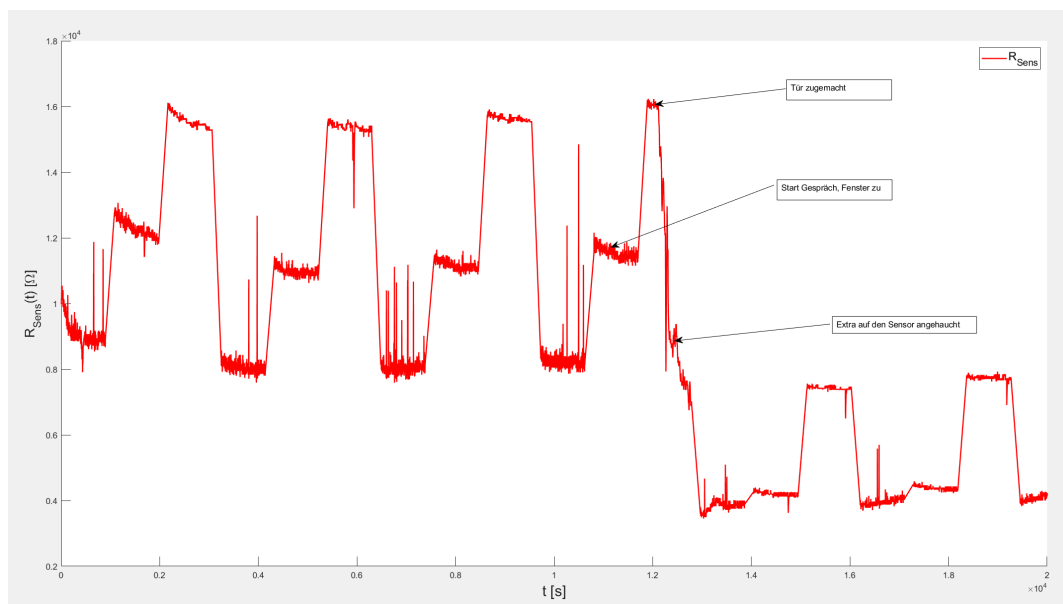


Abbildung 13: Messdaten vor und nach dem Schließen des Fensters

Wie es zu sehen ist, gibt es kein Aufstiege des Widerstands danach. Die Erklärung dazu ist, dass CO_2 mit dem geschlossenen Fenster und zugemachten Tür aus dem Zimmer nicht rausfliegen kann und deswegen behaltet der Sensorwiderstand die kleinere Werte.

Weiter sieht man 3 Zyklen, die ungefähr gleiche Werte haben - in der Zeit, also zwischen 18:22 und 21:40, war der Raum geschlossen war und keiner den besucht hat. Nachdem sieht man wieder eine Senkung der Pegeln. Der Grund dafür ist, dass eine Person ab dem Zeitpunkt im Zimmer war und eine neue Menge vom CO_2 produziert hat.

Die nächste 16 Zyklen (ungefähr 21:40 - 11:42) bleiben ca. auf einem Niveau, es ist aber trotzdem eine sehr kleine Erhöhung der Sensordaten zu merken. Das passiert, weil der abgeatmete Gas nicht sofort gleichmäßig sich im Raum verteilt und seine Konzentration neben dem Sensor anfangs größer war, als nach einer Weile.

Der nächste Verlauf beginnt schon nach der Aufstehzeit (11:15). Etwa um 12:23 war das Fenster wieder geöffnet, was auf der Abbildung (14) deutlich merkbar ist. Bis das Fenster zugemacht wurde, hat der Sensorwiderstand allmählich erhöht, was die Raumauffrischung und Befreiung von CO_2 bedeutet.

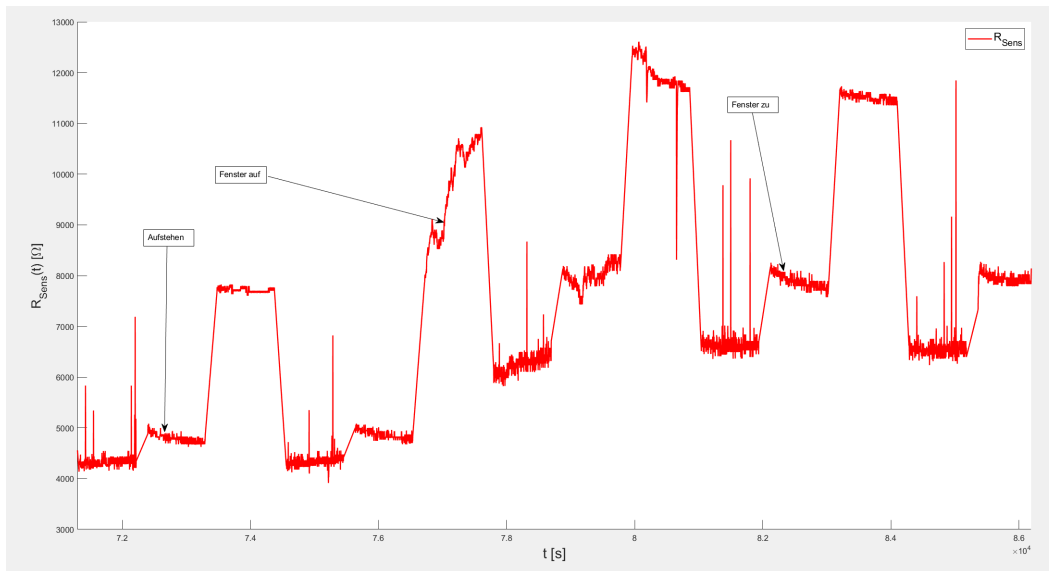


Abbildung 14: Messdaten nach dem Aufstehen

3.6.2 Genauigkeit der Sensordaten bei verschiedenen Temperaturen

Um die Genauigkeit der Sensordaten einzuschätzen, wird ein Zyklus genauer betrachtet. Der Ausschnitt aus dem gesamten Vorgehen ist auf der Abbildung (15) zu sehen.

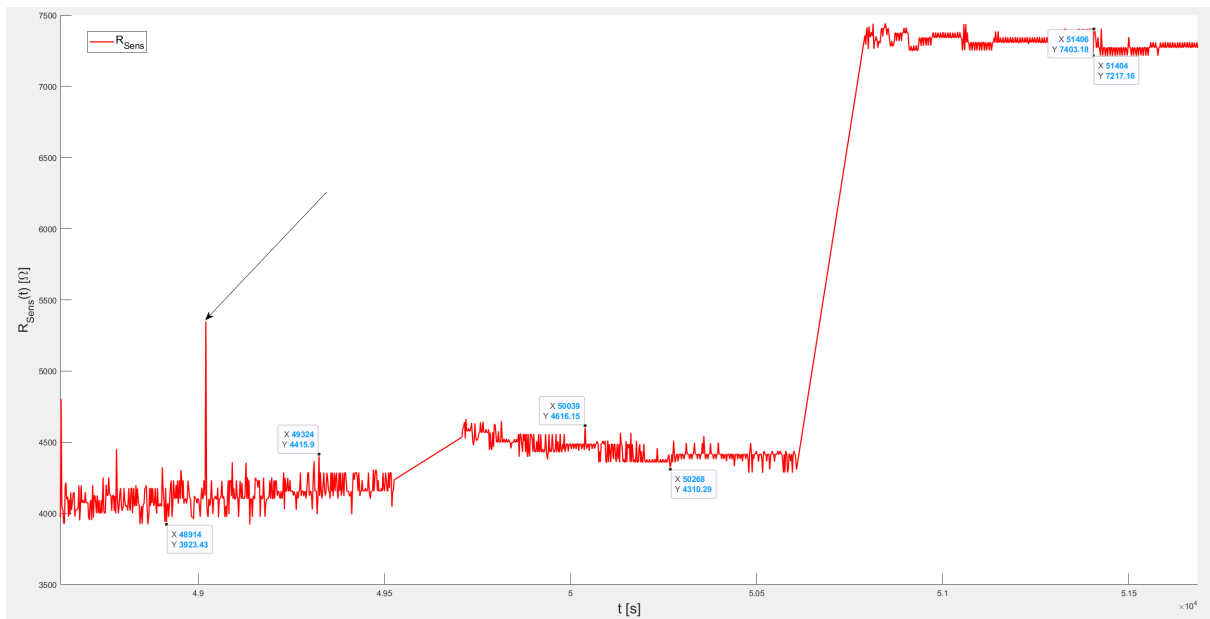


Abbildung 15: Ausschnitt der Messdaten während der Schlafzeit

Aus dieser Abbildung ist es deutlich erkennbar, dass die oberste Stufe die beste Genauigkeit besitzt, während die untere am meisten Rauschen hat. Die Widerstandsschwingungen bei 350°C haben die kleinste Amplitude und zeigen den stabilsten Signalverlauf. Die Unterschiede zwischen dem kleinsten und dem größten Widerstandswert für jede Stufe sind noch in der Tabelle (4) geschrieben.

Man kann noch die kurze Sprünge finden, die auf der Abbildung (15) auch mit einem Pfeil hingewiesen sind. Die sind nicht mehr als Messungsfehler, der wegen der Sensorwiderstandsabweichung sich erscheinen kann. Noch ein Grund dafür kann das Time-To-Digital Verfahren (2.6) sein, weil der Sensorwiderstand genau mit dieser Methode gemessen wurde. Es könnte so passieren, dass der Kondensator nicht immer voll geladen war, weil digitaler Arduino-Ausgangspin nicht stetig $3,3\text{V}$ ins Netz einspeist, was auch im Datenblatt [3] geschrieben ist. Diese Fehler sind aber kurzfristig und leicht erkennbar, deswegen spielen die kaum eine Rolle bei der Messungsanalyse.

Stufe 1 (150°C)	Stufe 2 (250°C)	Stufe 3 (350°C)
492Ω	306Ω	186Ω

Tabelle 4: Unterschied zwischen dem größten und dem kleinsten Sensorwiderstand bei der jeweiligen Temperatur während der Schalfzeit

Gleiche Aussagen lassen sich über die Zyklusstufen in der Tageszeit treffen. Das Vorgehen ist auf der Abbildung (16) dargestellt.

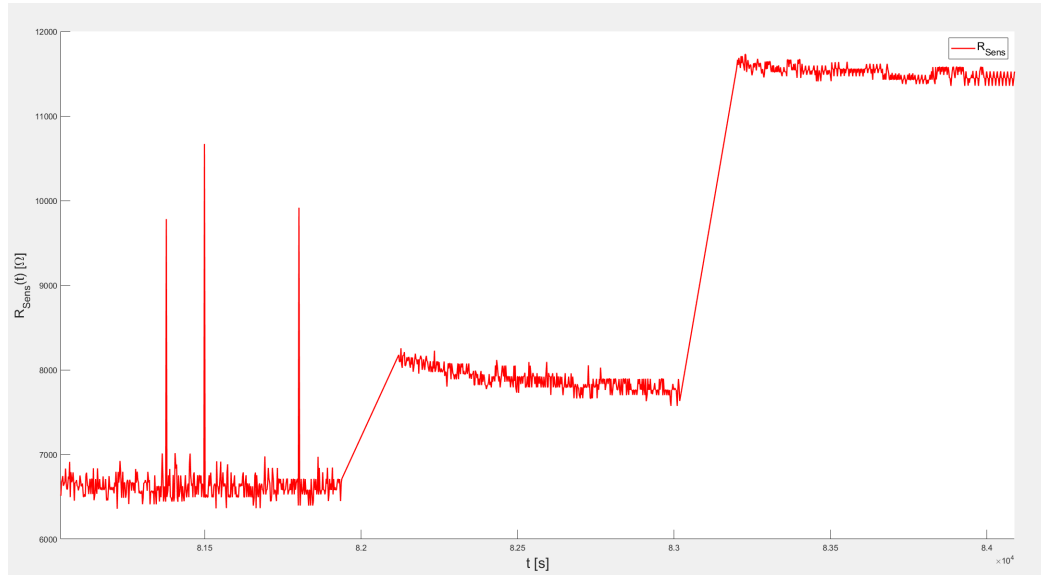


Abbildung 16: Ausschnitt der Messdaten während der Tageszeit

Auffällig ist hier größerer Abstand zwischen der Stufen. Grund dafür ist die Feuchtigkeit. Laut dem Datenblatt des Sensors [2], ist sein Widerstand desto größer, je geringer die Feuchtigkeit ist. Weil diese Abhängigkeit nicht linear ist, nimmt der Sensorwiderstand auf jeder Stufe auch unterschiedlich auf.

4 Zusammenfassung

Das Praktikum bot eine wertvolle Gelegenheit, sich intensiv mit der Arbeit mit Gassensoren auseinanderzusetzen und ein tieferes Verständnis für ihre Funktionsweise, Anwendung und Bedeutung zu entwickeln. Während des Praktikums wurden verschiedene Aspekte der Gassensorik behandelt, angefangen von theoretischen Grundlagen bis hin zur praktischen Anwendung.

Während des Praktikums wurden die grundlegenden Konzepte und Funktionsprinzipien von Gassensoren kennengelernt, darunter verschiedene Messmethoden und Kalibrierungstechniken. Durch praktische Übungen konnten diese theoretischen Kenntnisse in die Praxis umgesetzt und vertieft werden. Dabei wurden Herausforderungen wie die Optimierung von Messmethoden und die Interpretation von Messergebnissen adressiert.

Das wichtigste Ergebnis des Praktikums ist die erfolgreiche Entwicklung und Programmierung einer elektronischen Nase, die auf Gassensoren basiert. Diese Technologie zeigt das Potenzial von Gassensoren in der Detektion und Überwachung von Gasen und ihre Anwendbarkeit in verschiedenen Szenarien. Die erfolgreiche Umsetzung dieses Projekts unterstreicht die Bedeutung von Praxiserfahrung und experimentellem Lernen für die Weiterentwicklung.

Die Erkenntnisse aus dem Praktikum haben auch wichtige Implikationen für den Alltag. Es wurde festgestellt, dass die Luftqualität in Innenräumen erheblich von verschiedenen Faktoren beeinflusst wird, darunter die Anwesenheit von Schadstoffen wie Kohlenmonoxid und flüchtigen organischen Verbindungen. Eine einfache Maßnahme zur Verbesserung der Luftqualität besteht darin, Räume regelmäßig zu lüften, um Schadstoffe zu reduzieren und für eine ausreichende Belüftung zu sorgen.

A Time-To-Digital Messwerte

$R_{TTD} [\Omega]$ № \ $R_{Test} [\Omega]$	8000	10000	12000	15000	20000
1	8232.28	10014.22	12019.12	15169.11	20241.57
2	8236.35	10020.86	12038.26	15169.93	20242.56
3	8228.72	10010.43	12033.53	15161.93	20263.63
4	8238.14	10011.37	12045.09	15191.58	20243.46
5	8241.14	10014.22	12028.59	15183.39	20240.53
6	8233.40	10004.74	12020.81	15168.01	20256.02
7	8236.13	10018.95	12043.75	15178.62	20245.45
8	8239.58	10008.53	12040.74	15191.43	20222.47
9	8246.78	10012.31	12024.42	15172.35	20252.01
10	8237.97	10018.96	12032.76	15188.55	20242.38
11	8233.18	10016.10	12032.39	15182.38	20229.17
12	8236.58	10021.80	12033.13	15163.27	20232.91
13	8239.25	10025.59	12031.44	15169.44	20237.62
14	8239.81	10016.11	12037.86	15157.99	20230.00
15	8235.02	10017.06	12026.12	15162.79	20226.20
16	8237.64	10014.21	12029.16	15190.94	20218.58
17	8238.30	10005.68	12043.75	15163.67	20232.91
18	8237.30	10016.11	12026.89	15170.94	20268.17
19	8234.57	10011.37	12032.95	15174.67	20231.94
20	8238.19	10021.79	12029.33	15186.69	20223.36

Tabelle 5: Messwerte für Einschätzung des Time-To-Digital Verfahrens Teil(1)

$R_{TTD} [\Omega]$ № \ $R_{Test} [\Omega]$	27000	39000	47000	56000	82000
1	26957.67	39030.55	46807.85	55392.34	81897.86
2	26968.26	39119.68	46740.26	55358.09	81870.80
3	26970.73	39131.84	46742.42	55387.46	81853.75
4	26952.61	39059.93	46714.96	55363.33	81921.36
5	26961.75	39113.05	46736.74	55349.12	81821.89
6	26959.40	39103.13	46701.06	55307.47	81878.02
7	26978.11	39048.93	46734.19	55402.77	81916.59
8	26959.49	39127.43	46748.46	55426.11	81923.08
9	26969.21	39060.02	46744.63	55353.29	81983.89
10	26936.48	39112.52	46738.31	55376.46	81955.47
11	26964.39	39083.51	46725.67	55371.72	81907.92
12	26971.39	39103.05	46752.87	55359.11	81925.59
13	26946.97	39086.61	46718.74	55352.78	81867.49
14	26953.52	39085.57	46778.47	55338.20	81923.48
15	26950.76	39151.58	46740.18	55341.79	81963.24
16	26978.40	39049.38	46752.53	55386.66	81938.60
17	26981.33	39073.18	46740.55	55403.56	81879.73
18	26950.88	39074.48	46771.20	55373.76	81894.84
19	26965.99	39062.04	46770.86	55408.97	81957.04
20	26956.36	39087.78	46776.85	55335.10	81926.61

Tabelle 6: Messwerte für Einschätzung des Time-To-Digital Verfahrens Teil(2)

Testwiderstand $[\Omega]$	Durchschnittswert $[\Omega]$	Messungsfehler [%]
8000	8237.017	2.96
10000	10015.021	0.15
12000	12032.505	0.27
15000	15174.884	1.17
20000	20239.047	1.20
27000	26961.685	0.14
39000	39088.213	0.23
47000	46746.840	0.54
56000	55369.405	1.13
82000	81910.363	0.11

Tabelle 7: Testwiderstand im Vergleich zum Durchschnitt der gemessenen Werte

Literatur

- [1] Arduino. Arduino Reference. URL <https://www.arduino.cc/reference/en/>.
- [2] Figaro. *TGS 2600 - for the detection of Air Contaminants*. URL <https://www.figarosensor.com/product/docs/TGS2600B00%20%280913%29.pdf>.
- [3] Microchip Technology. *SAM D21/DA1 Family*. URL https://ww1.microchip.com/downloads/en/DeviceDoc/SAM_D21_DA1_Family_DataSheet_DS40001882F.pdf.
- [4] Nathaniel B. Nichols, John G. Nichols. Faustformelverfahren. URL [https://de.wikipedia.org/wiki/Faustformelverfahren_\(Automatisierungstechnik\)](https://de.wikipedia.org/wiki/Faustformelverfahren_(Automatisierungstechnik)).
- [5] Texas Instruments. *LM1086 1.5-A Low Dropout Positive Voltage Regulators*, . URL https://www.ti.com/lit/ds/symlink/lm1086.pdf?ts=1710563545614&ref_url=https%253A%252F%252Fwww.google.com%252F.
- [6] Texas Instruments. *LM124, LM124A, LM224, LM224A, LM324, LM324A, LM2902, LM2902V, LM224K, LM224KA, LM324K, LM324KA, LM2902K, LM2902KV, LM2902KAV QUADRUPLE OPERATIONAL AMPLIFIERS*, . URL <https://www.shoptronica.com/files/lm324k.pdf>.
- [7] TU Dortmund, Professur Sensorik. Wahlpflichtpraktikum: Entwicklung einer elektronischen Nase. URL https://moodle.tu-dortmund.de/pluginfile.php/2937731/mod_resource/content/1/Entwicklung%20einer%20elektronischen%20Nase_Praktikumsbeschreibung.pdf.
- [8] Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram. Vorlesung Steuerungs- und Regelungstechnik 3.2: Einfache Entwurfsverfahren. URL https://moodle.tu-dortmund.de/pluginfile.php/2745358/mod_resource/content/0/Steuerungs-%20und%20Regelungstechnik%203_2_IV.pdf.
- [9] Vishay Siliconix. *IRF540*. URL <https://www.vishay.com/docs/91021/irf540.pdf>.