



Hochschule für Technik,
Wirtschaft und Kultur Leipzig

FAKULTÄT INGENIEURWISSENSCHAFTEN

5420 - EMBEDDED SYSTEMS

Beleg des Projekts Abstandssensor

Autor Thomas Diner
 Christian Richter
Betreuer Marco Braun
 Andreas Reinhold

GitHub

https://github.com/c-rich-de/Projekt_Abstandssensor

15. März 2024

Inhaltsverzeichnis

Abbildungsverzeichnis	2
1 Einleitung	1
2 Allgemeine Informationen	1
2.1 Vorüberlegungen	1
2.2 Abstandssensor	1
3 Vorbereitungen	2
3.1 Sensor und Anschluss	2
3.2 Erstes Programm	3
3.3 Funktionstest	5
3.4 Inbetriebnahme der Teststation	5
3.4.1 Vorüberlegungen	5
3.4.2 Umsetzung	6
3.5 Versuchsaufbau	9
3.5.1 Drehscheibe	9
3.5.2 Teststation	9
3.5.3 Platine	11
3.5.4 Spannungsteiler	13
3.5.5 Reset Knopf, Display und Datenlogger	13
3.6 Schaltplan	14
3.6.1 Finales Programm	15
4 Aufnahme der Messreihen	17
4.1 Ermittlung der Vergleichswerte	18
4.1.1 Maximaler Abstand (Grundebene)	18
4.1.2 Minimaler Abstand	20

4.1.3	Drei Papierschichten	22
4.1.4	Neun Papierschichten	23
4.2	Aufzeichnung von dynamischen Versuchen	24
4.2.1	Durchlauf mit 200 1/min	24
4.2.2	Durchlauf mit 700 1/min	25
4.2.3	Durchlauf mit 700 1/min und 100 Hz Abtastfrequenz	26
4.2.4	Durchlauf mit 700 1/min und 1000 Hz Abtastfrequenz	27
5	Zusammenfassung und Fazit	28
5.1	Abstand	28
5.2	Frequenz	28
5.3	Anwendung	29
6	Anhang	30
7	Materialliste	34

Abbildungsverzeichnis

1	Omron Z4D-C01 Abstandssensor	1
2	modifiziertes Jumperwire mit Vergleich	2
3	Pico Breadboard	3
4	Quellcode mit Kommentaren	3
5	Quellcode der Ausgabe	4
6	Teststation im Ursprung	6
7	Vorrichtung zur Befestigung der Sensorhalterung	7
8	Sensorhalterung einzeln	7
9	Sensorhalterung am Motorgestell	8
10	Gesamtbild mit Sensorhalterung am Motorgestell	8

11	Drehscheibe mit den beiden Kreisfragmenten	9
12	Kreisfragmente auf Oszilloskop	10
13	Kreisfragmente mit Markierung	11
14	Draufsicht auf das Breadboard	12
15	Umbau auf die Lochrasterplatine	12
16	Schaltplan	14
17	Weitester Abstand	18
18	Messung des Maximalen Abstandes	19
19	Abstandsanzeige auf dem Display	20
20	Messung des Minimalen Abstandes	21
21	Abstandsanzeige auf dem Display	21
22	Messung bei einer Dicke von drei Papierschichten	22
23	Messung bei einer Dicke von neun Papierschichten	23
24	200 1/min	24
25	700 1/min	25
26	700 1/min und $f = 100$ Hz	26
27	700 1/min und $f = 1000$ Hz	27
28	Endgültige Platine mit weißer Markierung für Verpolungsschutz	28
29	Pinbelegung Raspberry Pi Pico H	30

1 Einleitung

Im Rahmen des Moduls *Embedded Systems* besteht die Aufgabe darin, einen Abstandssensor auf seine Tauglichkeit zu überprüfen. Der Anwendungsbereich des Sensors sollte wie folgt aussehen: Eine Metallplatte soll mithilfe eines Elektromagneten in der Luft gehalten werden. Die Platte wurde mit Führungsschienen so fixiert, dass wir nur auf einen Freiheitsgrad Acht geben müssen. Nun soll festgestellt werden, ob der uns zur Verfügung stehende Sensor dazu geeignet ist, die Höhe der Platte aufnehmen zu können.

2 Allgemeine Informationen

2.1 Vorüberlegungen

Nachdem uns der Sensor und der Mikrocontroller vorlagen, haben wir uns mit dem Datenblatt des Sensors vertraut gemacht und anschließend das Pin-Out des Mikrocontrollers heruntergeladen. Der Sensor kann laut dem Datenblatt in einem Bereich von $(6,5 \pm 1)$ mm arbeiten und besitzt eine Auflösung von $\pm 10 \mu\text{m}$. Damit scheint der Sensor ideal für die Aufgabe gemacht zu sein.

2.2 Abstandssensor

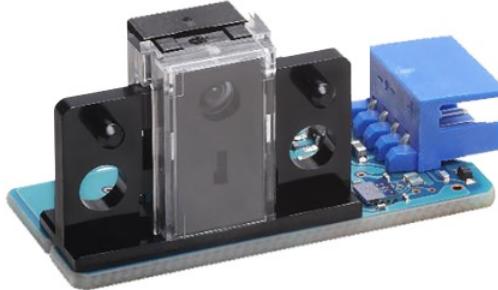


Abbildung 1: Omron Z4D-C01 Abstandssensor

Der Omron Z4D-C01 Abstandssensor besitzt 4 Anschlüsse:

- 1 = PLS: hier wird der benötigte Puls angelegt, welcher die Infrarot LED arbeiten lässt
- 2 = V_{cc} : hier wird die Versorgungsspannung angelegt, die der Sensor benötigt
- 3 = OUT: hier werden die analogen Daten (eine Spannung) ausgegeben, die der Sensor erfassst

-
- 4 = GND (Ground): hier wird die Masse angeschlossen, um einen geschlossenen Stromkreis herzustellen

Als Mikrocontroller haben wir den Raspberry Pi Pico H (2021) eingesetzt und für die Programmierung die Thonny IDE benutzt. Damit konnten wir ein Echtzeitfeedback erhalten und die Daten überwachen. Als Programmiersprache nutzen wir Python.

3 Vorbereitungen

3.1 Sensor und Anschluss

Der Sensor verfügt über eine Buchse, wo ein proprietäres Kabel anzuschließen ist. Der Abstand der Pins entspricht nicht dem sonst üblichen Standard. Wir schließen daher modifizierte Jumperwires an: wir entfernen von den beiden äußeren Jumperwires die Schutzverkleidung, um den Sensor erfolgreich verkabeln zu können. Da nur die äußeren Leitungen keine Schutzverkleidung besitzen, wird ein Kurzschlusschutz weiterhin gewährleistet.

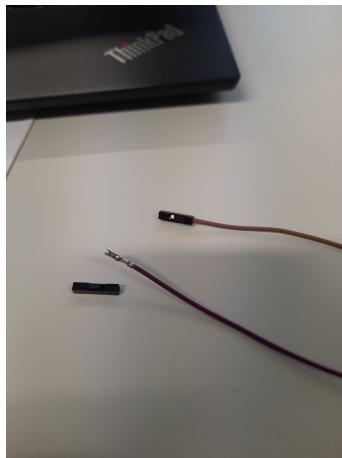


Abbildung 2: modifiziertes Jumperwire mit Vergleich

Anschließend haben wir die Jumperwires angesteckt und die benutzten Farben notiert:

- PIN 1 = PLS = blau
- PIN 2 = V_{cc} = schwarz
- PIN 3 = OUT = weiß
- PIN 4 = GND = lila

Danach wurde der Raspberry Pi Pico auf ein Breadboard gesteckt, um möglichst flexibel an die Anschlüsse zu gelangen.

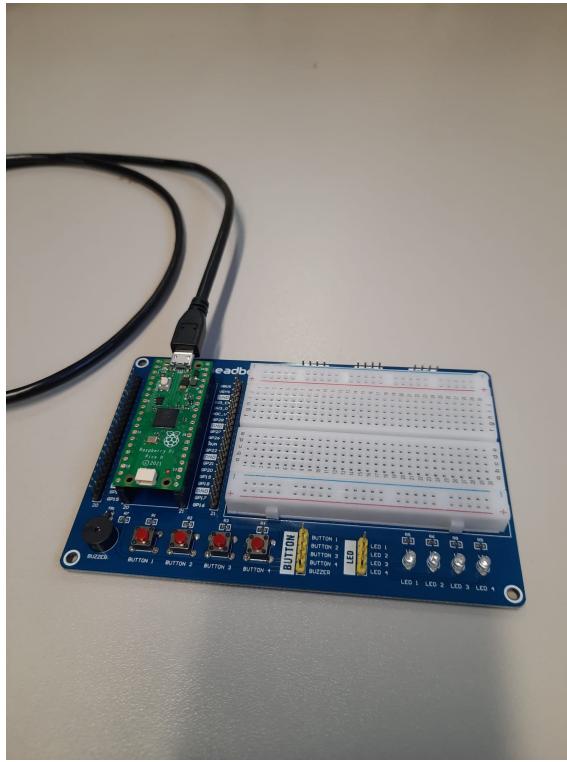


Abbildung 3: Pico Breadboard

Laut Datenblatt benötigt der Sensor eine Versorgungsspannung von 5 V und eine Pulsspannung von 3,5 V bis 5V. Die Versorgungsspannung stellt der Mikrocontroller durch seinen Anschluss von der microUSB Buchse zur Verfügung. Dafür haben wir den PIN 40 (VBUS) genutzt, da dieser direkt an den 5 V anliegt. Die Pulsspannung haben wir mit Hilfe der integrierten Pulsweitenmodulation (PWM) des Picos sichergestellt und legten dafür den PIN 25 (GP19) fest, da dieser die Voraussetzungen aufweist (siehe Pin-Out des Picos). Das Ausgangssignal des Sensors haben wir an den PIN 31 (GP 26) angeschlossen, um diesen über die Thonny IDE auslesen zu können.

3.2 Erstes Programm

Nun haben wir ein kurzes Programm geschrieben, um die Funktion des Sensors zu überprüfen. Im folgenden wird das benutze Programm aufgeschlüsselt und erklärt:

```
from machine import Pin, PWM #Initialisierung der PWM- Funktionseinheit
pwm = PWM(Pin(19))           #und des GPIO-Pins
pwm.freq(10)                 #Einstellen der Frequenz für die PWM -> hier 10Hz
pwm.duty_u16(58986)          #Eistellen vom Tastgrad (Duty Cycle) -> hier 90%
```

Abbildung 4: Quellcode mit Kommentaren

Für die Initialisierung benötigen wir die Funktion der Pulsweitenmodulation (PWM) und den Zugriff auf die GPIO-Pins (General Purpose Input/Output), welche in den ersten beiden Zeilen

erfolgt. Hierbei haben wir uns für den GPIO PIN 19 entschieden, da dieser einer der vielen Pins ist, welche die PWM-Funktion besitzt. Die benötigte Frequenz haben wir experimentell herausgefunden und legten diese auf **10 Hz** fest. Den erforderlichen Duty Cycle von 90% entnahmen wir dem Datenblatt. Eine Frequenz von 10 Hz bedeutet für den Sensor, dass dieser die Entfernung zu seinem Objekt zehn mal pro Sekunde misst und weiter gibt. Mit einem Duty Cycle von 90% erreicht man, dass die Flanke 90% der Zeit des Pulses über null liegt und die verbleibenden 10% gleich null sind. Um im Programm den benötigten Duty Cycle zu erreichen, teilt man die 100 möglichen Prozent in die 65.536 möglichen Werte auf (da der Mikrocontroller eine 16 bit ADU besitzt). Es ergeben sich folgende Werte:

- 0%: 0
- 10%: 6.554
- 20%: 13.108
- 30%: 19.662
- 40%: 26.216
- 50%: 32.268
- 60%: 39.324
- 70%: 45.878
- 80%: 52.432
- 90%: 58.986
- 100%: 65.535

Somit ergibt sich die Zahl 58.986 für den benötigten Duty Cycle von 90%.

```
from machine import ADC      #Initialisieren des ADC
from time import sleep       #Initialisieren Sleep - Befehls aus dem 'time' Moduls

pot = ADC(Pin(26))          #Erstellt ADC - Objekt am GPIO-Pin 26

while True:                  #begin der while - Schleife
    pot_value = pot.read_u16() #liest den analogen Wert (Abstand) und speichert in 'pot_value'
    print(pot_value)         #Ausgabe des Wertes
    sleep(0.1)               #pausiert um 0,1 s
```

Abbildung 5: Quellcode der Ausgabe

An dieser Stelle wird nun der Analog-Digital-Wandler (ADC) initialisiert, welcher den analogen Messwert aufnehmen und umwandeln soll. Zudem wird eine 'Wartezeit' (sleep) benötigt damit die Anzahl der ausgegebenen Werte eingeschränkt wird, um den Pico nicht zu überlasten und die

Daten noch mit dem Auge verfolgen zu können.

Nun wird ein ADC-Objekt namens 'pot' erstellt und mit dem GPIO-Pin 26 verknüpft, da dieser einen internen ADC besitzt. Danach beginnt die while-Schleife, wobei der Wert des analogen Abstandes gelesen und in ein digitales 16-bit-Signal umgewandelt wird. Anschließend erfolgt die Speicherung in die Variable 'pot_value'. Danach wird der Wert der Variable als Digitalwert (0 - 65535) ausgegeben und eine Pause von 0,1 s eingestellt.

3.3 Funktionstest

Nachdem das oben erklärte Programm auf den RPi Pico überspielt wurde, haben wir zunächst überprüft, ob der Puls auch ordnungsgemäß ausgegeben wird und diesen auf eine LED des Breadboards angeschlossen. Der Test verlief erfolgreich. Jetzt wurde der Sensor richtig angeschlossen und eingeschaltet. Nun konnte man in der Thonny IDE erkennen, dass der Sensor auch Werte ausgibt. Daraufhin haben wir den Sensor mit dem Finger abgedeckt, ihn sehr langsam davon entfernt und dabei die ausgegebenen Werte beobachtet. Die Beobachtung ergab, dass sich die Werte wie gewünscht ändern und wir damit den ersten Funktionstest erfolgreich abschließen können.

3.4 Inbetriebnahme der Teststation

3.4.1 Vorüberlegungen

Nachdem der Sensor grob auf seine Funktionalität überprüft worden war, haben wir uns der Teststation zugewendet. Diese soll dazu dienen den Sensor auf seine Grenzen zu überprüfen. Hierbei lag das wesentliche Augenmerk auf den folgenden Punkten:

- Wie schnell reagiert der Sensor bei schnellen Änderungen des Abstands?
- Wie nah darf ich mit dem Sensor an das Objekt heran?
- Wie weit darf ich von dem Objekt entfernt sein?

Dabei entstand, gemeinsam mit Andreas Reinhold, ein Aufbau, der wie folgt aussieht: Ein runde Drehscheibe wird als Grundfläche benutzt. Auf dieser werden Fragmente mit unterschiedlicher Höhe platziert. Die Höhen sind vorher bekannt und es wird mit dem Sensor gemessen. Die Drehscheibe soll sich gleichmäßig drehen, um herauszufinden wie schnell der Sensor die Höhenänderungen aufnehmen kann.

3.4.2 Umsetzung

Für die Umsetzung der Teststation bekamen wir eine bereits vorhandene Konstruktion aus dem Labor. Dabei handelt es sich um einen Motor, welcher mit einem Gestell umgeben ist, auf welchem sich ein Frequenzumrichter und eine Bedieneinheit befindet. Mit dem Bedienelement kann man die Drehrichtung, sowie die Drehgeschwindigkeit von 0 bis $700 \frac{U}{min}$ problemlos steuern.



Abbildung 6: Teststation im Ursprung

Nun bestand die Aufgabe darin, die Teststation zu erweitern, sodass diese optimal geeignet dafür ist, den Sensor an seine Grenzen zu bringen. Hierfür benötigten wir eine Konstruktion, welche den Sensor genau auf die Drehscheibe richtet. Dabei sollte die Erweiterung stabil in der Konstruktion und flexibel in der Anwendung sein, damit eine genaue Aufnahme der Grenzen der Sensors sicher gestellt werden kann.

Mit Hilfe von Herrn Recklies und CAD-Programmen haben wir eine Konstruktion entworfen, die uns es ermöglicht den Abstandssensor in einem Bereich von 6,5 mm von der Drehscheibe zu positionieren. Die Nachfolgenden Abbildungen sollen die Konstruktion beschreiben:

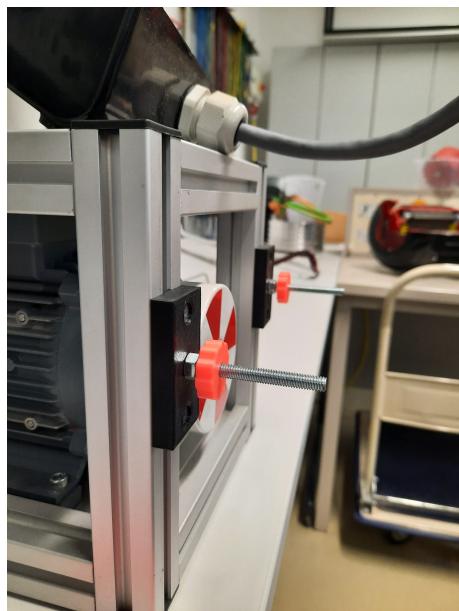


Abbildung 7: Vorrichtung zur Befestigung der Sensorhalterung

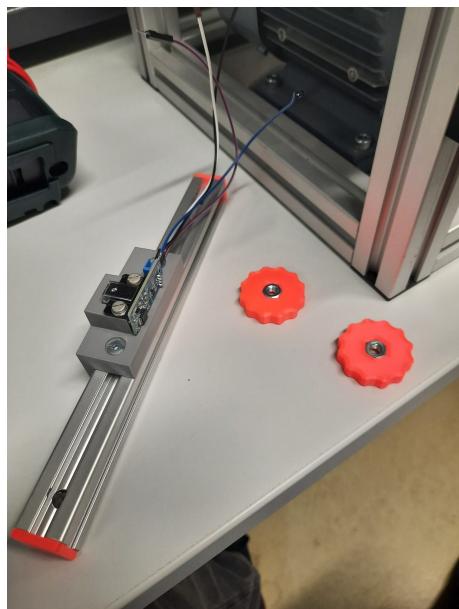


Abbildung 8: Sensorhalterung einzeln

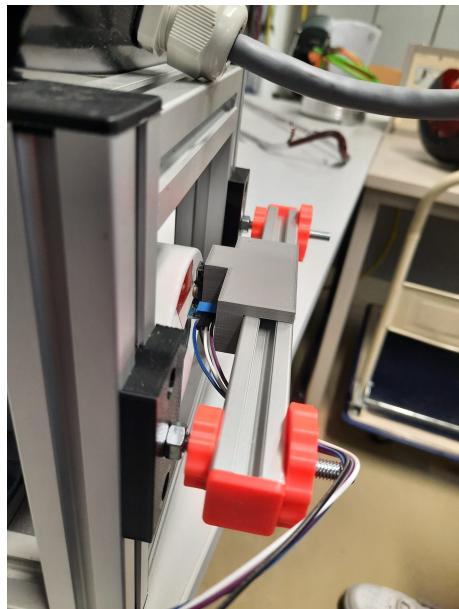


Abbildung 9: Sensorhalterung am Motorgestell

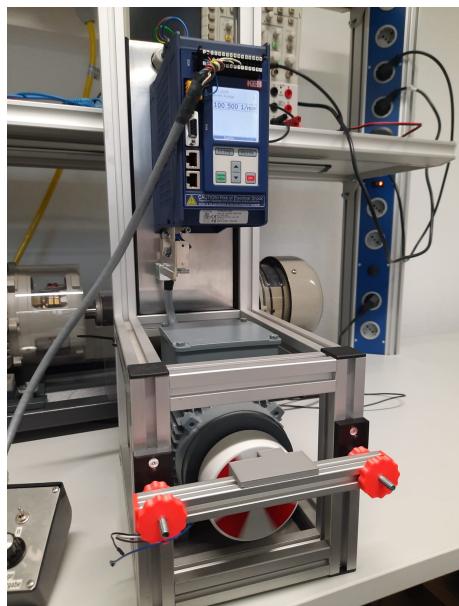


Abbildung 10: Gesamtbild mit Sensorhalterung am Motorgestell

Nach Montage der Abstandshalterung ging es darum herauszufinden, an welchem Punkt der Sensor ideal platziert ist; also wie nah bzw. wie weit weg wir den Sensor platzieren dürfen. Das Datenblatt gibt dabei an, dass der Arbeitsbereich $(6,5 \pm 1)$ mm beträgt. Die Halterung ist vertikal, horizontal und in der Tiefe verstellbar. Nach der erstmaligen Einstellung der Horizontalen und Vertikalen wurden diese nicht weiter verändert, wobei die Tiefe je nach Testlauf des Öfteren variiert wurde. Des Weiteren fiel uns auf dem Datenblatt auf, dass die Werte mit einem 'munsell paper with a reflection factor of 70%' aufgenommen wurden. Dies bedeutete nichts anderes als, dass der Sensor mit herkömmlichem weißen (Drucker)-Papier aufgenommen wurde. Deshalb haben wir kurzer

Hand die gesamte Fläche der Drehscheibe mit einem Kreis aus Papier beklebt. Hier galt besondere Vorsicht, da man unbedingt darauf achten musste die Fläche gleichmäßig auf die Drehscheibe zu kleben und Unebenheiten zu vermeiden. Jede kleine Unebenheit könnte das Messergebnis in der Rotation beeinflussen.

Nun konnte die Testreihe beginnen. Dafür nahmen wir uns ein Oszilloskop zur Hilfe und überwachten den eingehenden Puls und den ausgegebenen Spannungswert (OUT). Letzteren griffen wir noch *vor* dem Mikrocontroller ab.

3.5 Versuchsaufbau

3.5.1 Drehscheibe

Wir beklebten die Fläche der Drehscheibe mit zwei Kreisfragmenten aus Papier. Eines besteht aus drei, das andere aus neun Papierschichten.

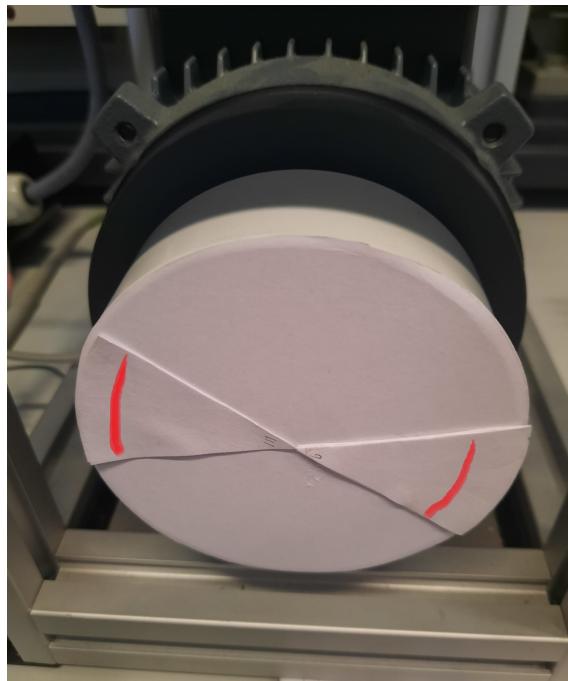


Abbildung 11: Drehscheibe mit den beiden Kreisfragmenten

Die rot markierten Stellen markieren den Bereich, an welchem der Sensor misst.

3.5.2 Teststation

Um den Abstand verändern zu können haben wir zunächst die Einstellschrauben der Tiefeneinstellung abgenommen. Anschließend haben wir vorsichtig, mit beiden Händen die Schiene mit dem Sensor vor und zurück bewegt. Dabei galt es gleichzeitig den Monitor des Oszilloskops zu beobach-

ten, um ein Gefühl dafür zu bekommen, wo die Grenzen ungefähr liegen.

Nun haben wir den Motor rotieren lassen, damit die Kreisfragmente in immer gleichen Abständen vor dem Sensor entlanglaufen.

Jetzt haben wir den Sensor vorsichtig in Richtung Drehscheibe bewegt und dabei das Oszilloskop beobachtet. Wir haben mit gezieltem Vor- und Zurückschieben des Sensors die Grenzwerte von 0,2 V und 4,4 V ermittelt und diese jeweils im Oszilloskop mit einem Cursor (Y1 = 0,2 V und Y2 = 4,4 V) veranschaulicht. Die Messung erfolgte noch vor dem Spannungsteiler, damit wir die genauen Spannungswerte aufnehmen konnten.

Dann haben wir den Sensor etwa in der Mitte dieser Grenzwerte platziert, wobei folgendes Bild entstand:

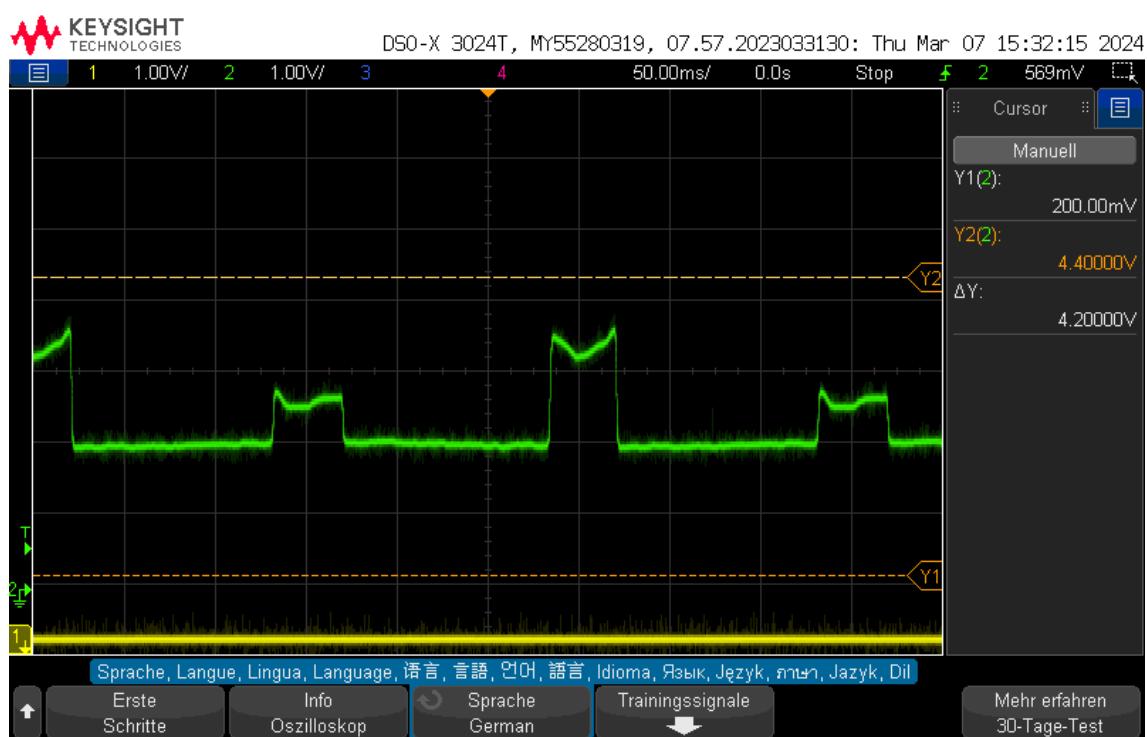


Abbildung 12: Kreisfragmente auf Oszilloskop

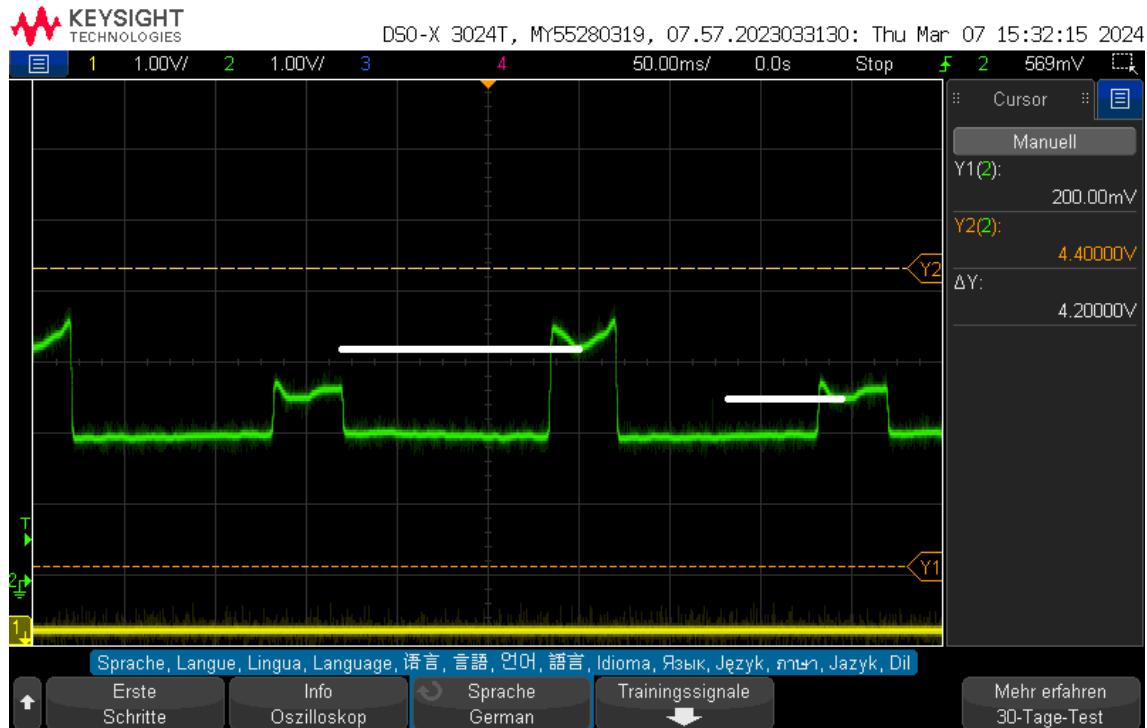


Abbildung 13: Kreisfragmente mit Markierung

Da die Skalierung 1 V/div beträgt, können wir bereits einen Spannungswert ablesen. Für die kleinere Flanke (drei Schichten Papier) entspricht dies etwa 0,6 V und für die größere Flanke (neun Schichten Papier) circa 1,2 V. Auf dem Bild ist sehr gut zu erkennen, wie der Sensor die Wölbungen der Kreisfragmente problemlos erkennen kann. Die weißen Linien markieren den Bereich von dem wir ausgehen, dass er die 'echte' Dicke der drei bzw. neun Schichten darstellt. Bevor wir nun mit der Ermittlung der Grenzwerte beginnen wollten, haben wir uns zunächst dem Platinenaufbau gewidmet.

3.5.3 Platine

In unserem ersten Durchlauf haben wir die Kommunikation des Mikrocontrollers mit dem Sensor mittels Breadboard durchgeführt, da dort Veränderungen sehr einfach realisierbar sind. Mit diesem Versuchsaufbau haben wir auch, wie bereits erwähnt, einen Spannungsteiler eingeführt, welcher den Ausgang des Sensors (5 V) auf 3,3 V für den Eingang des ADC aufteilt. Damit ist es möglich das volle Spektrum des Sensors zu nutzen. Allerdings hat der Aufbau aber den Nachteil, dass die Leitungen sehr locker angeschlossen sind und die Messgrößen vor Allem durch die Leitungswiderstände, aber auch durch magnetische Felder, beeinflusst werden. Somit lies sich ein Umbau nicht vermeiden.

Daher wurde der Versuchsaufbau auf eine Lochrasterplatine bewegt, welche vor allem die Leitungswiderstände minimieren sollte. Ebenso ist damit ein versehentliches Lösen der Kabelverbindungen nicht mehr möglich und die Bedienung ist nutzerfreundlicher geworden. Auch ein Reset-Knopf für

das schnelle Zurücksetzen des Mikrocontrollers, ebenso wie ein Display zur Anzeige der Momentanwerte wurden eingeführt.

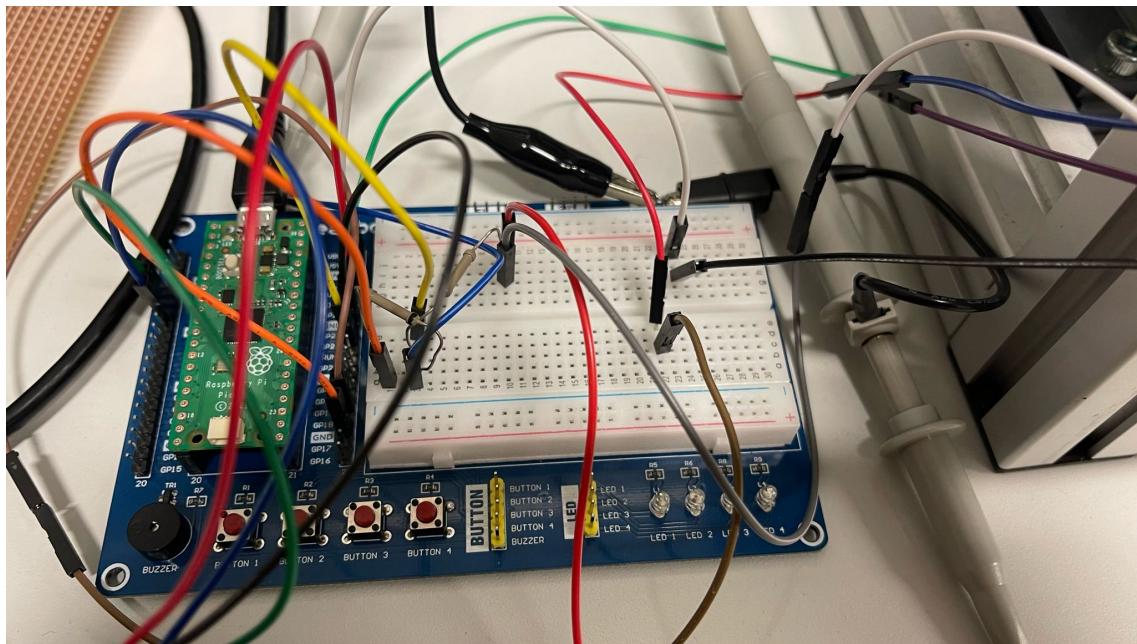


Abbildung 14: Draufsicht auf das Breadboard

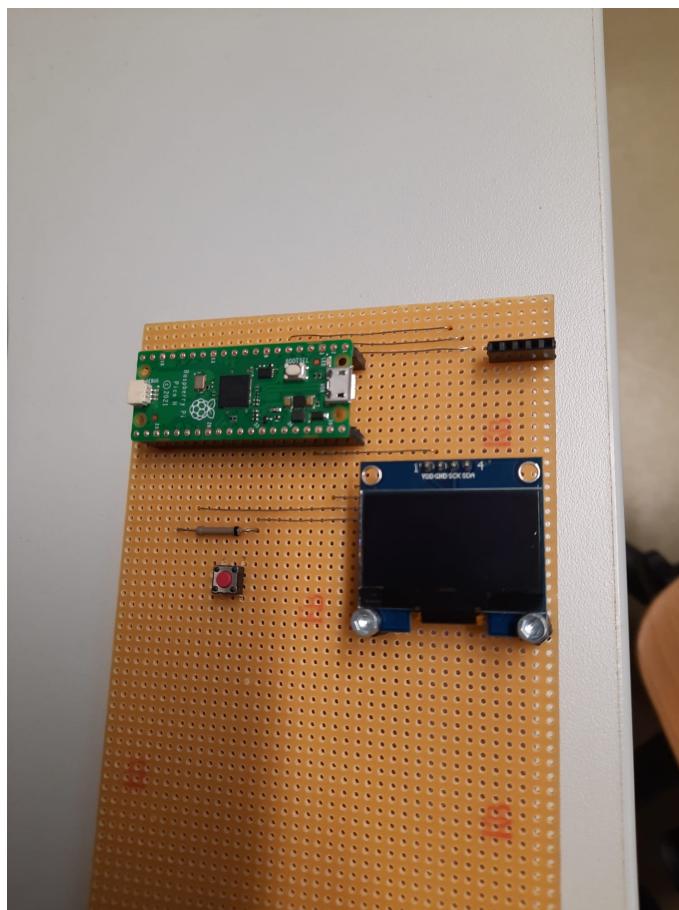


Abbildung 15: Umbau auf die Lochrasterplatine

3.5.4 Spannungsteiler

Für unseren Spannungsteiler haben wir $R_1 = 40 \text{ k}\Omega$ und $R_2 = 75 \text{ k}\Omega$ gewählt. Diesen haben wir zunächst auf dem Breadboard getestet und mit dem Oszilloskop das Teilverhältnis verifiziert. Nun haben wir bei einem 5 V Ausgang am Sensor, einen 3,3 V Eingang beim ADC. Um nun die vollständigen Werte ordentlich auslesen zu können haben wir das Teilverhältnis in das Programm mit eingearbeitet.

Zur Veranschaulichung ist unten der Rechenweg für den Spannungsteiler gegeben. Die Werte für x_1 und x_2 haben wir ermittelt, indem wir mit dem Oszilloskop und einem Messschieber zwei Wertepaare aufgenommen haben, um eine lineare Beziehung (eine Gerade) aufzunehmen. Jede Spannung bekam damit einen Abstandswert zugewiesen.

$$U = \frac{\Delta u}{\Delta x} \cdot x + U_0$$

mit $x_1 = 2,5 \text{ mm}$; $x_2 = 5 \text{ mm}$ und $u_1 = 4,4 \text{ V}$; $u_2 = 0,2 \text{ V}$

$$U = \frac{u_2 - u_1}{x_2 - x_1} \cdot x + U_0 = \frac{-4,2V}{2,5\text{mm}} \cdot x + U_0$$

$$\rightarrow U(x_1, u_1) = 4,4 \text{ V} = -\frac{4,2V}{2,5\text{mm}} \cdot 2,5 \text{ mm} + U_0$$

$$U_0 = 8,6 \text{ V}$$

$$u' = k \cdot U = \frac{R_2}{R_1 + R_2} \cdot U \text{ mit } R_1 = 40 \text{ k}\Omega \text{ und } R_2 = 75 \text{ k}\Omega$$

$$u' = 0,65 \cdot U \rightarrow U = -\frac{4,2V}{2,5\text{mm}} \cdot x + 8,6 \text{ V}$$

$$u' = k \cdot U = 0,65 \cdot \left(-\frac{4,2V}{2,5\text{mm}} \cdot x + 8,6 \text{ V} \right) = -1,1 \frac{V}{\text{mm}} \cdot x + 5,61 \text{ V}$$

$$\text{nach x umstellen} \rightarrow x = \frac{u' - 5,61V}{-1,1 \frac{V}{\text{mm}}}$$

$$\text{für den ADC: } \frac{3,3V}{65535} = \frac{u'}{y}$$

$$u' = \frac{3,3V}{65535} \cdot y \text{ einsetzen in } \rightarrow x = \frac{u' - 5,61V}{-1,1 \frac{V}{\text{mm}}}$$

y = der angezeigte Digitalwert im Programm

3.5.5 Reset Knopf, Display und Datenlogger

Das oben erwähnte Display wird über I2C angesprochen. Dafür wurden die Pins konfiguriert, die Anzeige gedreht und verschiedene Elemente zur Anzeige erstellt. Wie dies im Code aussieht ist in der nachfolgenden Sektion genauer aufgelistet.

Der Reset-Knopf stellt sicher, dass der Mikrocontroller jederzeit in seinen anfänglichen Programmzustand gebracht werden kann. Damit wird auch das Display zurückgesetzt und die Log-Datei neu erstellt. Für die Umsetzung des Reset musste man lediglich den RUN- Pin mit GND (Ground) verbinden.

Die Log-Datei kann zur Auswertung der digitalen Werte und deren Umrechnung in Spannung und Abstand genutzt werden. Diese wird nach Beendigung des Programms auf den Raspberry Pi Pico geschrieben und kann z.B. über die Thonny IDE auf den lokalen PC übertragen werden.

3.6 Schaltplan

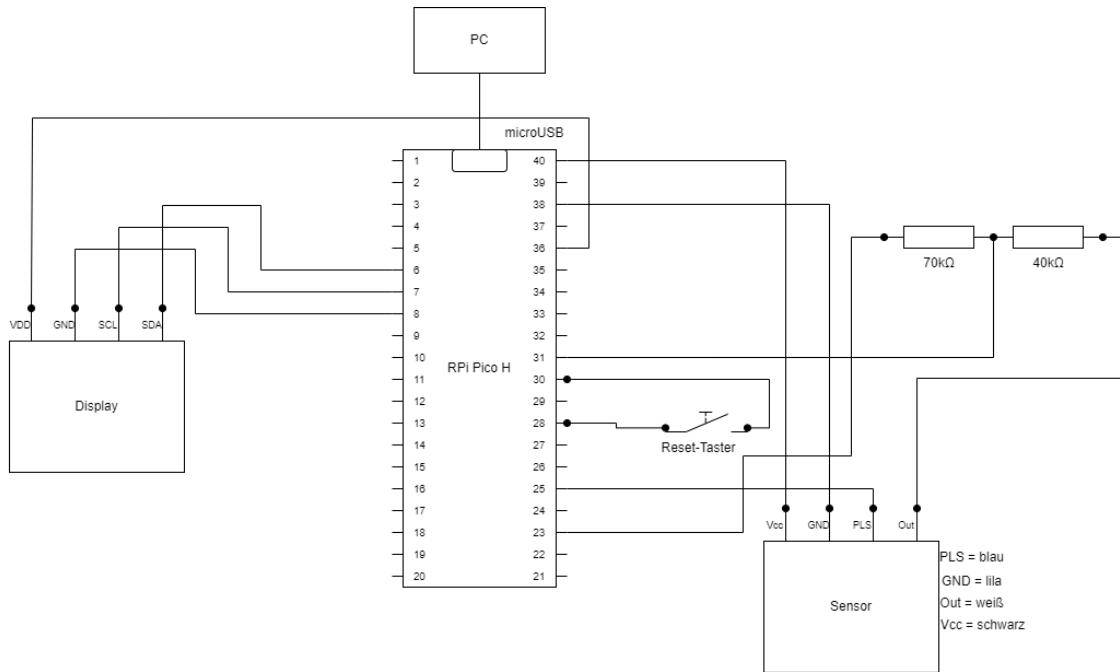


Abbildung 16: Schaltplan

3.6.1 Finales Programm

Um den RPi Pico H auch nutzen zu können, muss dieser vorher ausgewählt werden:

Run → Configure Interpreter → Auswählen von Micropython und dem angeschlossenen Port.

```
from machine import ADC      #Initialisieren des ADC
from time import sleep       #Initialisieren Sleep-Befehls aus dem 'time' Modul

#I2C betriebenes SH1106 OLED-Display
from machine import I2C, ADC
from sh1106 import SH1106_I2C
import framebuf

WIDTH  = 128                  #Breite des OLED-Displays
HEIGHT = 64                   #Höhe des OLED-Displays

i2c = I2C(0)                 #I2C mit I2C0 Standardwerten, SCL=Pin(GP5), SDA=Pin(GP4), freq=400000

print("I2C Address      : "+hex(i2c.scan()[0]).upper()) #Display Adresse
print("I2C Configuration: "+str(i2c))                      #Display I2C Konfig.

oled = SH1106_I2C(WIDTH, HEIGHT, i2c)                     #Initialisierung des OLED-Displays
oled.rotate(True)                                         #Drehen des Bildes um 180 Grad

#Sensor-Anschluss
from machine import Pin, PWM                               #Initialisierung der PWM-Funktionseinheit
pwm = PWM(Pin(19))                                       #und des GPIO-Pins
pwm.freq(13)                                            #Einstellen der Frequenz für die PWM
pwm.duty_u16(65535)                                      #Einstellen vom Tastgrad (Duty Cycle)
pot = ADC(Pin(26))                                       #Festlegung des Pins für den ADC

#log-file erstellen
file = open("data.csv", "w")

#Definition von max und min
max_abstand = 0                                         #umgekehrte Proportionalität
min_abstand = 65535

#Durchlauf mit Anzeige von Werten
```

```
while True:
    #ADC
    pot_value = pot.read_u16()
    #Berechnung der Anzeigewerte
    pot_spannung = (3.3 * pot_value)/65535
    abstand = (pot_spannung - 5.61)/(-1.05)
    max_abstand = max(abstand, max_abstand)
    min_abstand = min(abstand, min_abstand)
    diff = max_abstand - min_abstand
    #Ausgabe in der Shell zur Kontrolle
    print(pot_value)
    print(pot_spannung)
    print(abstand)
    #OLED-Display (links/rechts, oben/unten)
    #Abstandsanzeige
    oled.text("Abstand:",5,5)
    oled.text(str(abstand)[:5],5,15)
    oled.text("mm",55,15)
    #Füller
    oled.text("-----",0,23)
    #Max/Min Anzeige
    oled.text("max:    min:",5,30)
    oled.text(str(max_abstand)[:5],5,40)
    oled.text(str(min_abstand)[:5],60,40)
    oled.text("mm",110,40)
    #Differenzanzeige
    oled.text("Diff.:",5,55)
    oled.text(str(diff)[:5],60,55)
    oled.text("mm",110,55)
    #Ausgabe des Displays
    oled.show()
    sleep(0.08)      #für weniger häufige Aktualisierung der Messwerte (auf pwm.freq abgestimmt!)
    oled.fill(0)      #reset des OLED-Displays auf schwarz
    #Füllen des log-files
    file.write("Wert: " + str(pot_value) + " | Spannung (V): " + str(pot_spannung) +
              " | Abstand (mm): " + str(abstand) + "\n")
    file.flush()
```

Das Programm kann unter Anderem die digitalen Messwerte, durch das Verhältnis des Spannungsteilers und dem Verhältnis von digital zu analog, umrechnen, um welchen Abstand es sich in diesem Moment handelt.

Wie man auch sieht wurde der Duty-Cycle auf 100% angehoben. Dies ist geschehen, da beim Messen ansonsten 10% aller Messwerte fehlen (bei einem Duty Cycle von 90%). Die digitalen Werte des Sensors bzw. das Oszilloskop haben bei einem geringeren Wert an dieser Stelle 'ungewünschte Werte' aufgenommen, welche bei der Auswertung gestört hätten.

Bei der Wahl der Abtastfrequenz ist darauf zu achten, dass die PWM und die sleep-time aufeinander abstimmt werden (z.B.: 50Hz entsprechen einem sleep von 0,02s). Dafür wird einfach die Frequenz umgerechnet.

4 Aufnahme der Messreihen

Nachdem das Programm sowie die Platine ihren finalen Zustand erreicht haben, konnten wir nun mit der Aufnahme der Grenzwerte beginnen. Dabei musste man nun einen Messwert für den nahesten Abstand aufnehmen, den dazugehörigen Digitalwert und den tatsächlichen Abstand messen.

Für die Versuchsreihe haben wir immer die gleichen Parameter des PWM benutzt:

$f=100\text{Hz}$, Duty-Cycle = 100% (65535), sleep 0,01s und eine Umdrehungsgeschwindigkeit von 200 1/min. Der Spannungshöchstwert des ADC von 3,3 V wurden dabei nicht überschritten. Wir haben die Grenzwerte mit Hilfe des Oszilloskops 'Keysight InfiniiVision DSO-X 3024T' aufgenommen.

4.1 Ermittlung der Vergleichswerte

4.1.1 Maximaler Abstand (Grundebene)

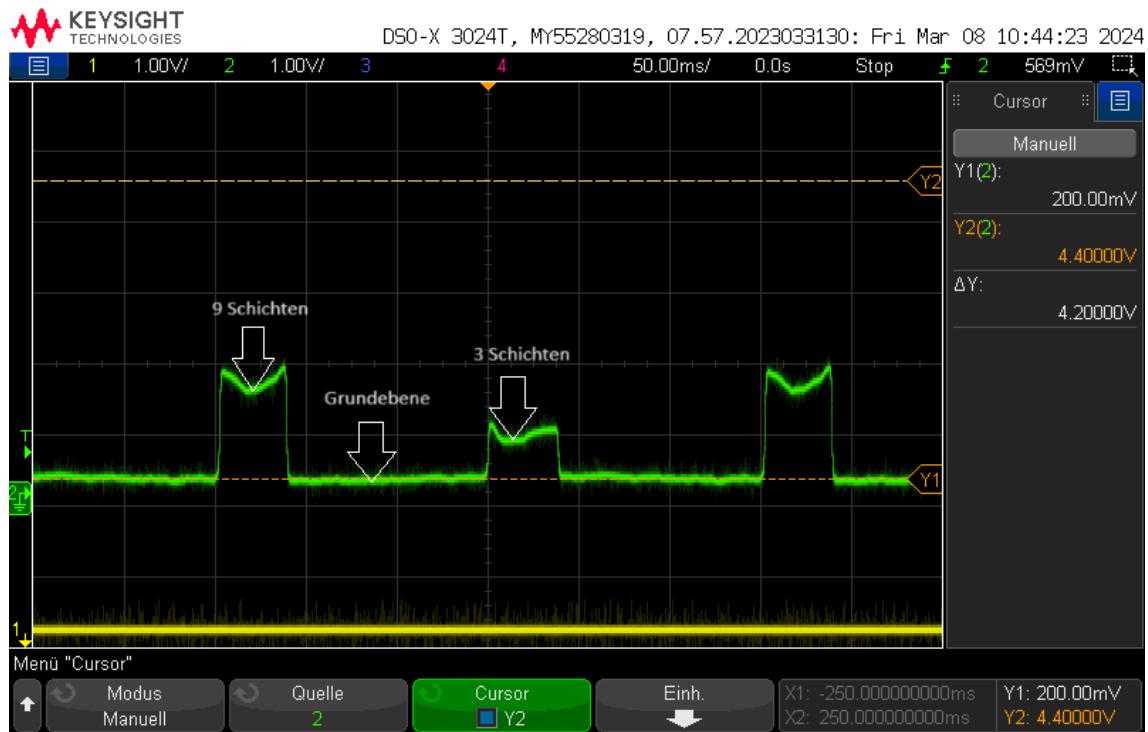


Abbildung 17: Weitester Abstand

Die Abbildung zeigt, wie wir methodisch den maximalen Abstand zur Messebene ermittelt haben. Wird die Grenze von ca. 0,2 V unterschritten, so werden die Amplituden der Messwerte gedämpft, bis es zum vollständigen Einbruch kommt.

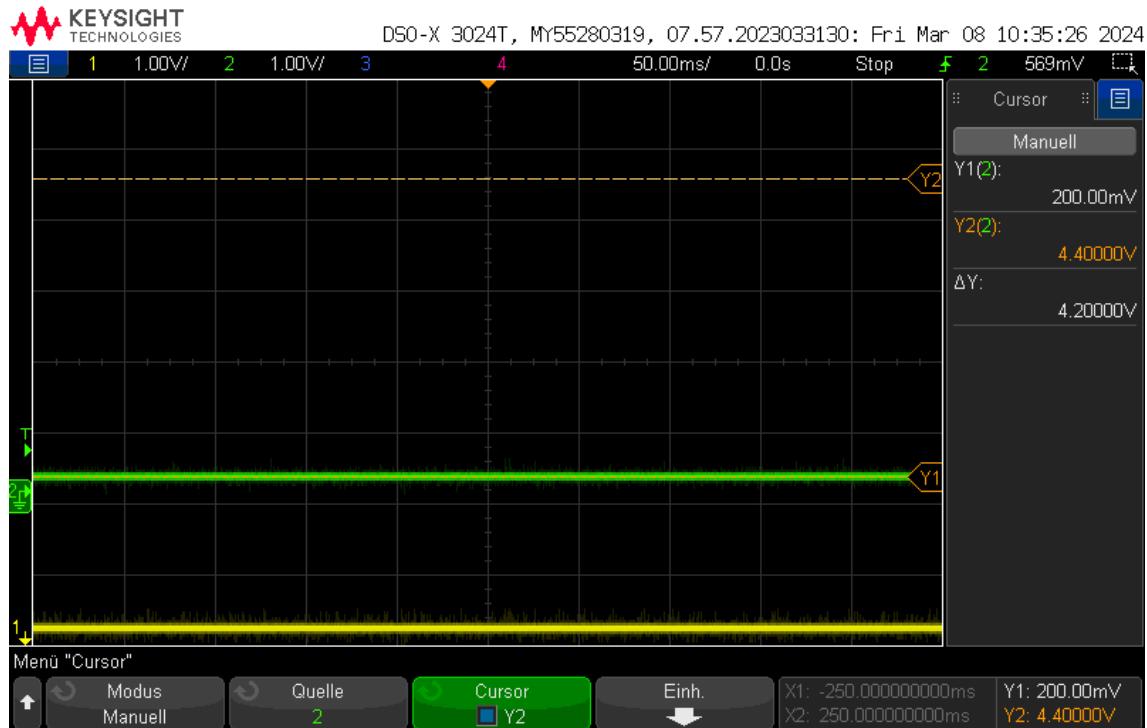


Abbildung 18: Messung des Maximalen Abstandes

Die digitalen Messwerte für diesen Abstand wurden, so wie alle nachfolgenden, mit dem Datenlogger ausgewertet. Dieser gibt die Daten in einer Tabelle aus. Zu beachten ist dabei, dass der errechnete Spannungswert nur zur Umrechnung in den Abstand mit ausgegeben wird. Für die weitere Verwendung des Sensors als Abstandsanzeige ist die Spannung nicht von Bedeutung und wird daher nicht weiter mit erwähnt. Nachfolgend sind die Mittelwerte der jeweiligen Werte angegeben. Hier wurden diese über einen Zeitraum von 10 s gemessen, um das Hintergrundrauschen zu kompensieren.

Digitalwert	Abstand (mm)
3475	5,176

Zu erkennen ist, dass das Oszilloskop an der unteren Grenze einen Spannungswert von 0,2 V ausgibt. Wir haben auch den tatsächlichen Abstand mit einem Messschieber gemessen - dieser gibt uns einen Abstand von ca. 5,2 mm. Der errechnete Abstand liegt dabei mit 5,176 mm sehr nah daran. Eine geringe Abweichung ist dennoch ersichtlich.

4.1.2 Minimaler Abstand

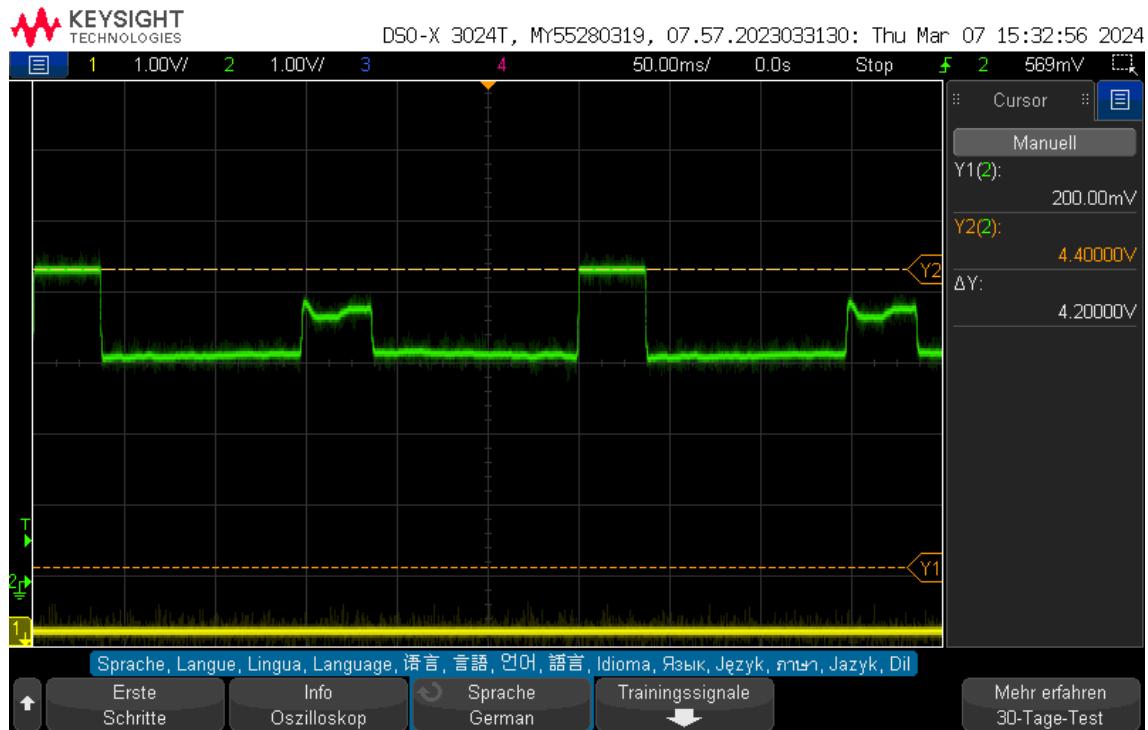


Abbildung 19: Abstandsanzeige auf dem Display

Man sieht, dass das Papier mit neun Schichten bereits nicht mehr vollständig abgebildet werden kann. Daraufhin haben wir den Abstand des Sensors zur Messebene weiter verringert, bis keine Messwerte mehr abgebildet werden konnten. Dies erfolgt bei ca. 4,4 V. Am Messschieber ist dabei ein Abstand von ca. 2,55 mm ablesbar.

Kleinere Abweichungen lassen sich durch ein gewisses Signalrauschen erklären. Der Versuchsaufbau wurde zwar verbessert, ist aber dennoch nicht vom Rest der Umwelt abgeschlossen.

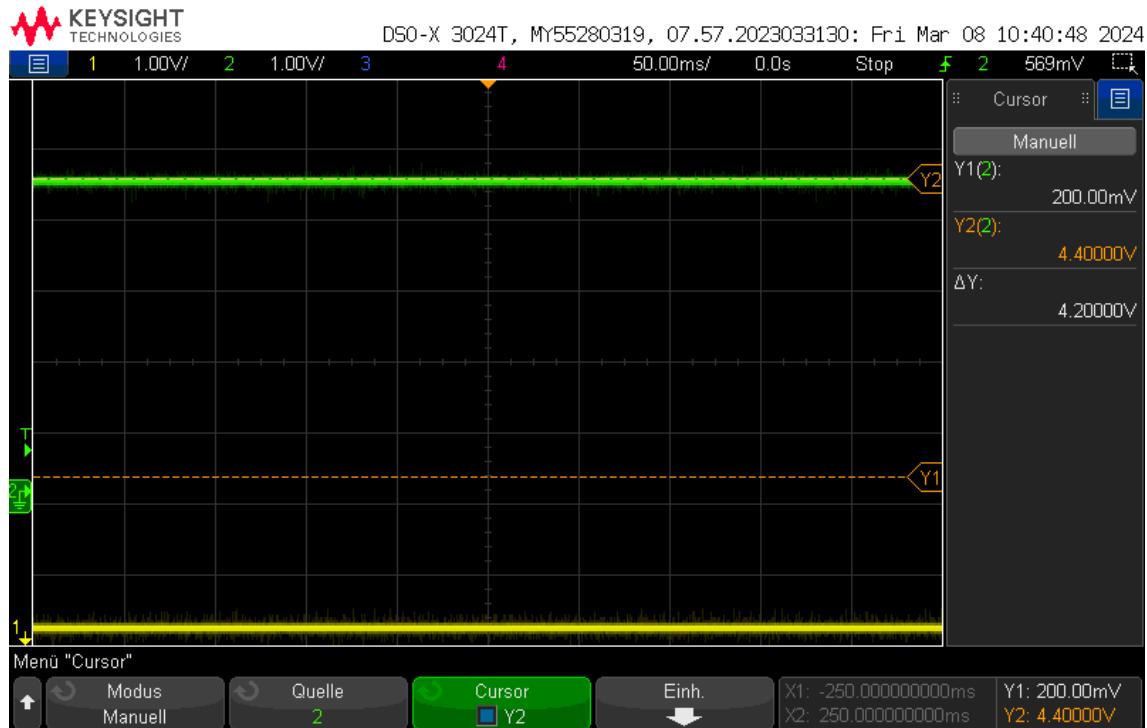


Abbildung 20: Messung des Minimalen Abstandes

Digitalwert	Abstand (mm)
56683	2,625

Auch hier ist zu sehen, dass die Umrechnung gut funktioniert. Eine Abweichung ist zu erkennen, jedoch ist diese auch nicht dramatisch hoch. Auf dem Display sind die Werte für max und min auch ablesbar. Die hier angezeigten Werte sind allerdings von einem weniger genauen Versuchsdurchlauf, weshalb sie etwas höher erscheinen, als die im Text angegebenen:

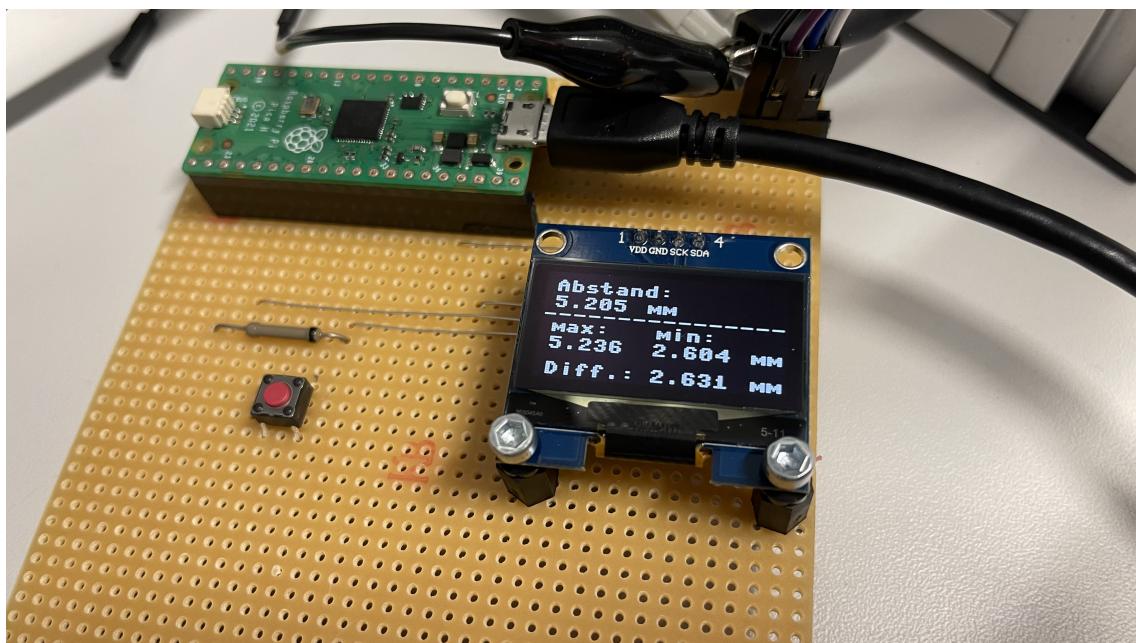


Abbildung 21: Abstandsanzeige auf dem Display

4.1.3 Drei Papierschichten

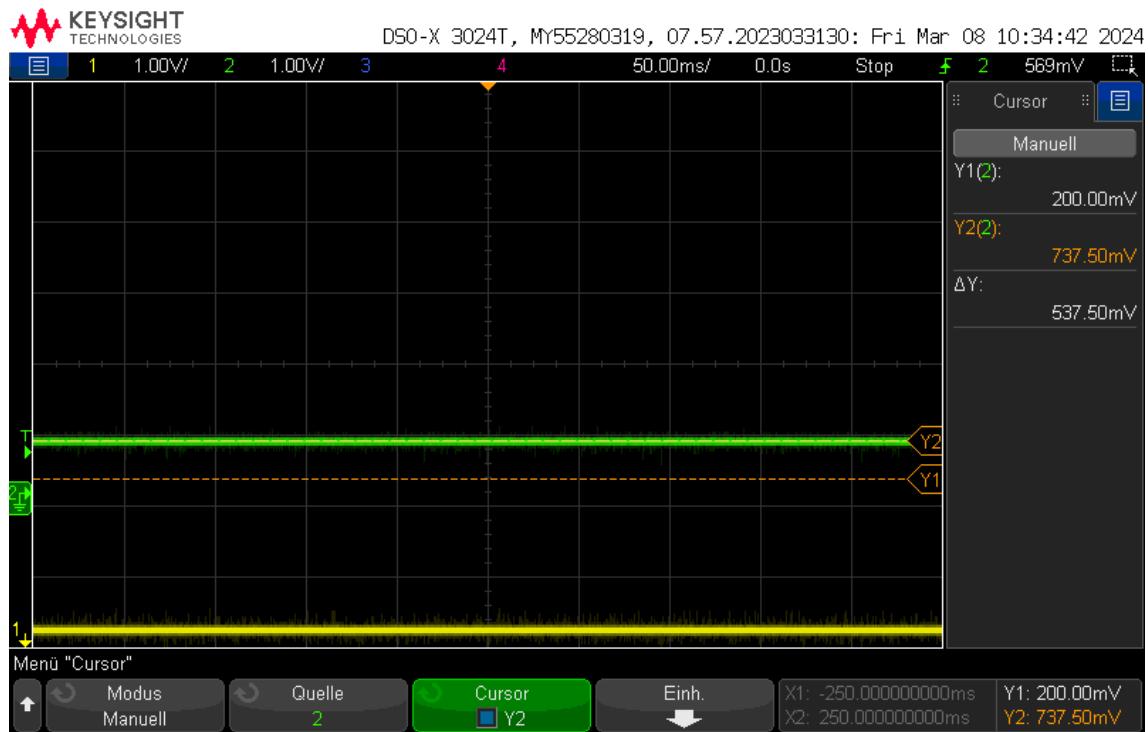


Abbildung 22: Messung bei einer Dicke von drei Papierschichten

Digitalwert	Abstand (mm)
10098	4,859

4.1.4 Neun Papierschichten

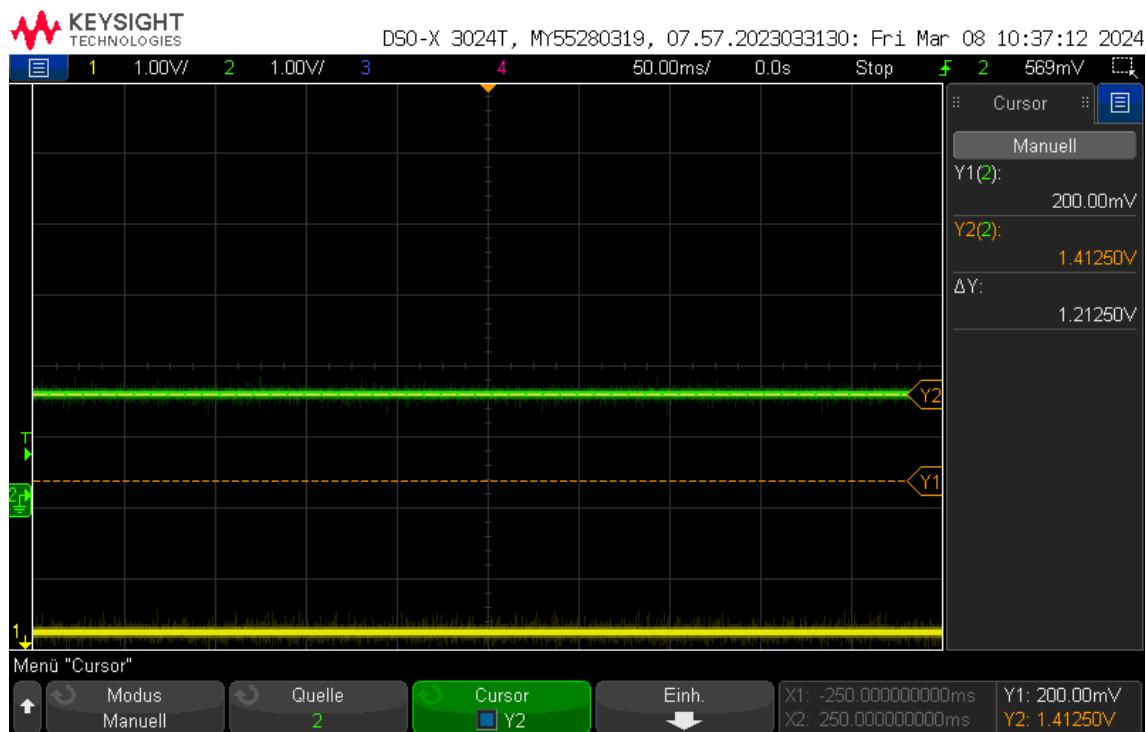


Abbildung 23: Messung bei einer Dicke von neun Papierschichten

Digitalwert	Abstand (mm)
18991	4,432

4.2 Aufzeichnung von dynamischen Versuchen

4.2.1 Durchlauf mit 200 1/min

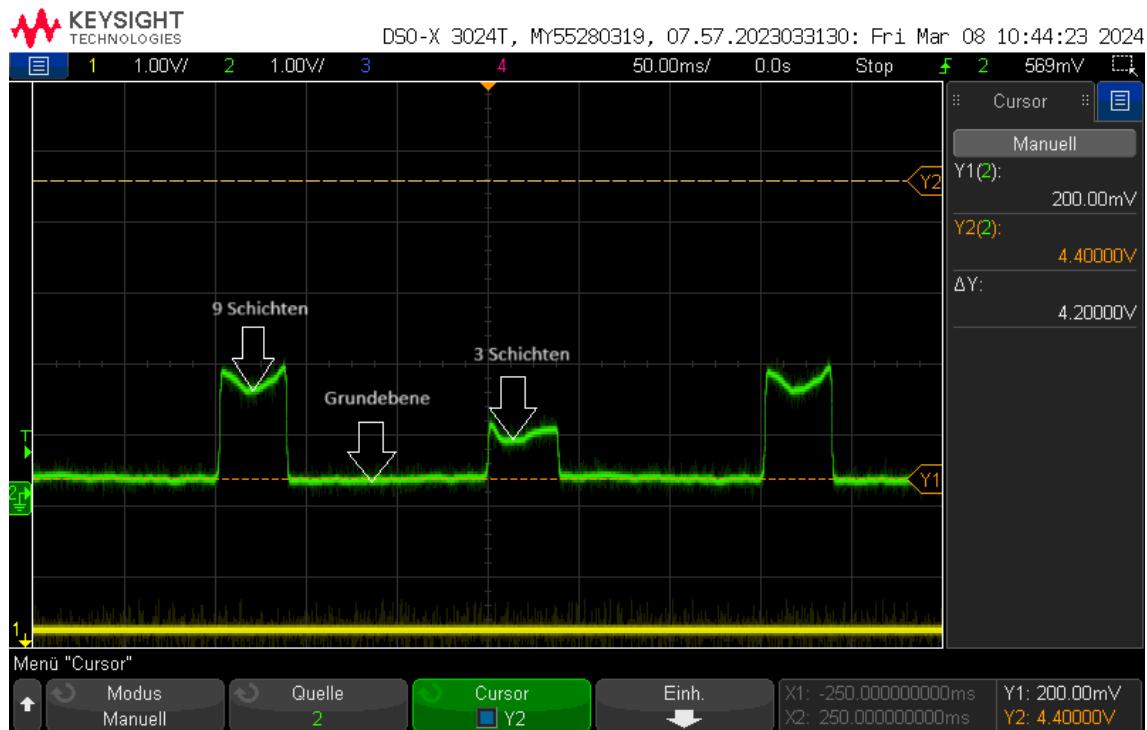


Abbildung 24: 200 1/min

Dieser Versuchsdurchlauf kombiniert die vorherigen Versuche miteinander. Es erfolgte eine dynamische Aufzeichnung der Messwerte, indem die Drehscheibe mit 200 1/min angetrieben wurde. Damit sieht man die unterschiedlichen Papierschichten sowohl auf dem Oszilloskop, als auch digital im Datenlogger. Im Folgenden soll verglichen werden, wie gut der Sensor das Signal digital abbilden kann.

Unsere Auswertung gibt uns die folgenden Daten für den Versuchsdurchlauf:

3 Schichten		9 Schichten	
Digitalwert	Abstand (mm)	Digitalwert	Abstand (mm)
11415	4,795	20520	4,359

Zu sehen ist, dass diese Werte eine geringe Abweichung zu den Messwerten aus den gesonderten Durchläufen besitzen. So weichen die dynamischen Werte der 3 Schichten um 0,064 mm gegenüber den statischen ab. Diese Diskrepanz ist allerdings wirklich gering. Ähnlich sieht es auch bei den dynamischen Werten für 9 Papierschichten aus. Der Unterschied beträgt 0,073 mm. Die Digitalwerte haben höhere Abweichungen voneinander, aber umgerechnet in den Abstand sind diese Abweichungen nicht beachtlich schlecht. Ganz im Gegenteil: Die Messung zeigt, dass der Sensor auch bei dynamischen Aufzeichnungen zuverlässig arbeiten kann. Es ist sogar möglich, dass diese

Abweichungen durch Umwelteinflüsse entstanden sind, obwohl darauf geachtet wurde den Aufbau nicht zu berühren oder in die nähere Umgebung elektronische Geräte einzuführen.

4.2.2 Durchlauf mit 700 1/min

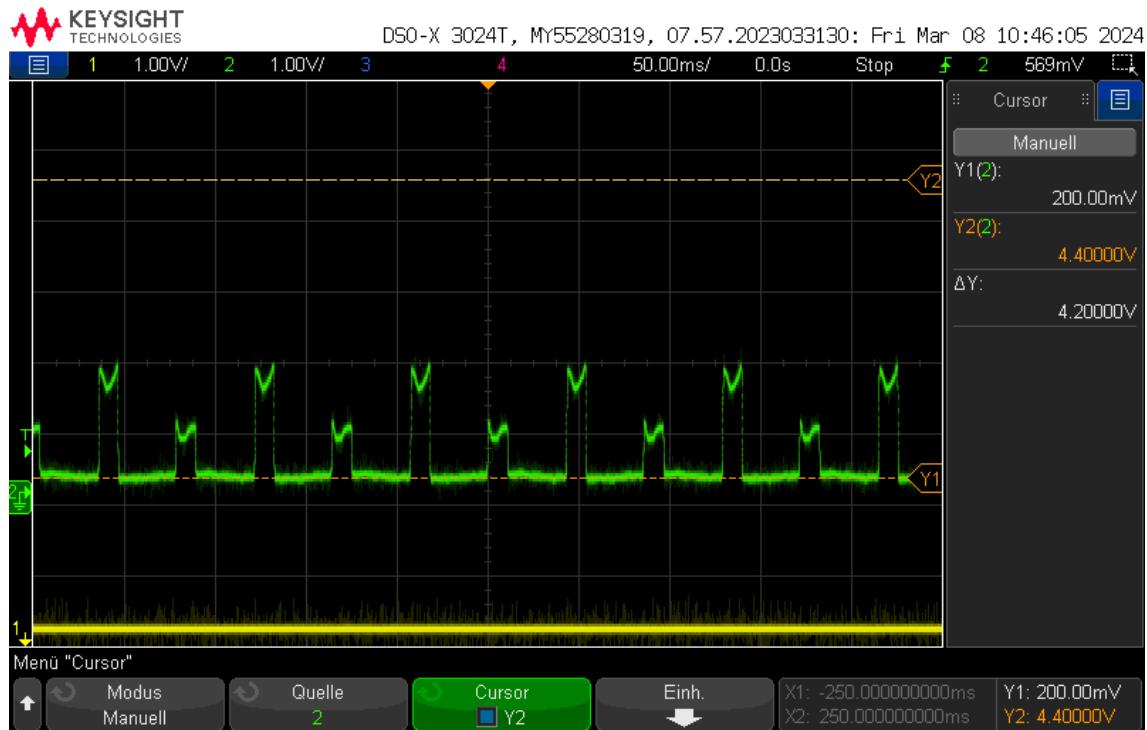


Abbildung 25: 700 1/min

Nun wurde die Drehgeschwindigkeit auf das 3,5-fache erhöht. Damit ist zu erwarten, dass der Sensor bei der gleichen Frequenz der Aufnahme der Messwerte das Signal nicht mehr vollständig digital abbilden kann bzw. viele Höhenänderungen einfach 'überspringt'.

3 Schichten		9 Schichten	
Digitalwert	Abstand (mm)	Digitalwert	Abstand (mm)
11397	4,796	20726	4,349

Aus den Mittelwerten ist keine große Veränderung zu sehen. Die Veränderungen sind mit 0,063 mm für drei und 0,083 mm für neun Schichten nicht weiter auffällig schlechter geworden. Wenn man jedoch einen Blick in die Log-Datei wirft, so wird einem sofort klar: Der Sensor bekommt nicht mehr alle Höhenänderungen mit bzw. ist zwischen diesen schlechter zu unterscheiden. Teilweise sieht man mehrere Werte nur von den neun Papierschichten, dann wieder eine dynamische Aufzeichnung usw. Es ist an der Zeit die Abtastfrequenz zu erhöhen und den sleep dementsprechend anzupassen. Natürlich hätte man dies auch schon bei den vorherigen Messungen unternehmen können, jedoch ist es nicht sinnvoll Werte übermäßig oft zu messen, da das Gesamtergebnis damit nicht beeinflusst wird.

4.2.3 Durchlauf mit 700 1/min und 100 Hz Abtastfrequenz

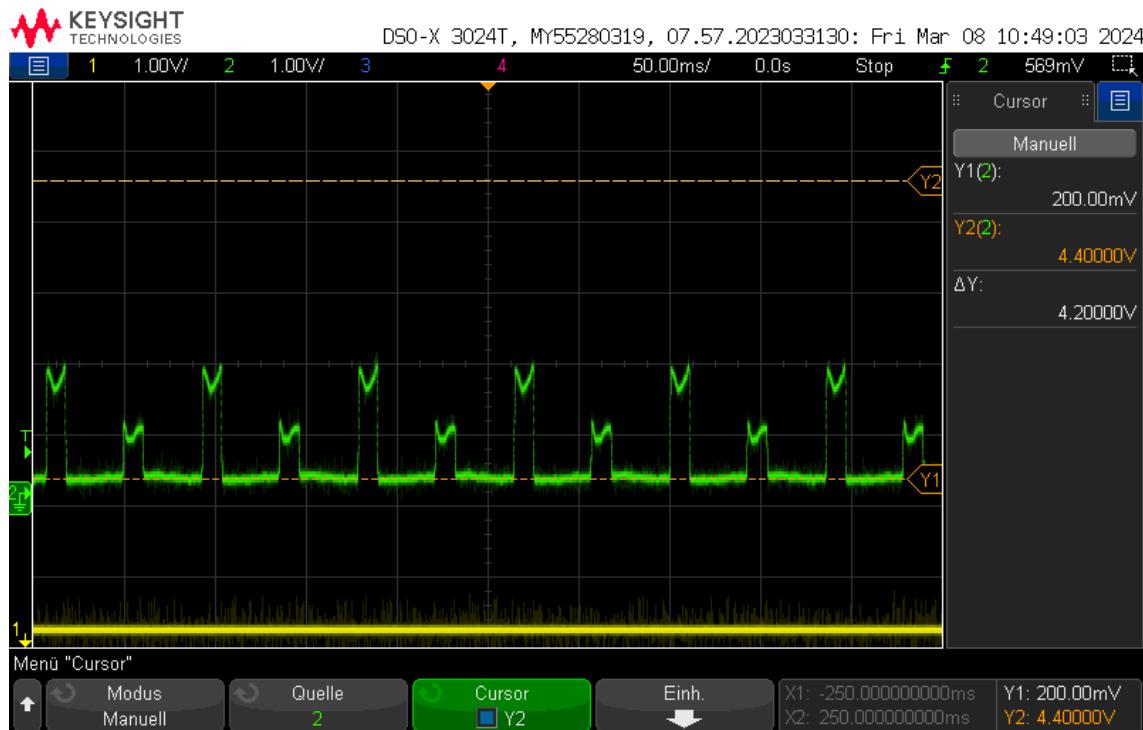


Abbildung 26: 700 1/min und $f = 100$ Hz

3 Schichten		9 Schichten	
Digitalwert	Abstand (mm)	Digitalwert	Abstand (mm)
11720	4,780	21509	4,312

Erneut ist die Diskrepanz zwischen statischem und dynamischen Messwert vernachlässigbar klein. In der Log-Datei sieht man jedoch, dass die Anzahl der mehrfachen Datenpunkte für die gleiche Papierdicke reduziert werden konnte. Dies ist insofern hilfreich, da der Aufbau somit auf dem Display immer den richtigen Abstand anzeigt und in der Log-Datei speichert, was für die spätere Weiterverwendung des Projektes von Vorteil ist. Es ist also empfehlenswert die Abtastfrequenz zu erhöhen, auch wenn damit eventuell zu viele Messwerte im Gesamten aufgenommen werden - für die Auswertung ist es durchaus von Vorteil.

4.2.4 Durchlauf mit 700 1/min und 1000 Hz Abtastfrequenz

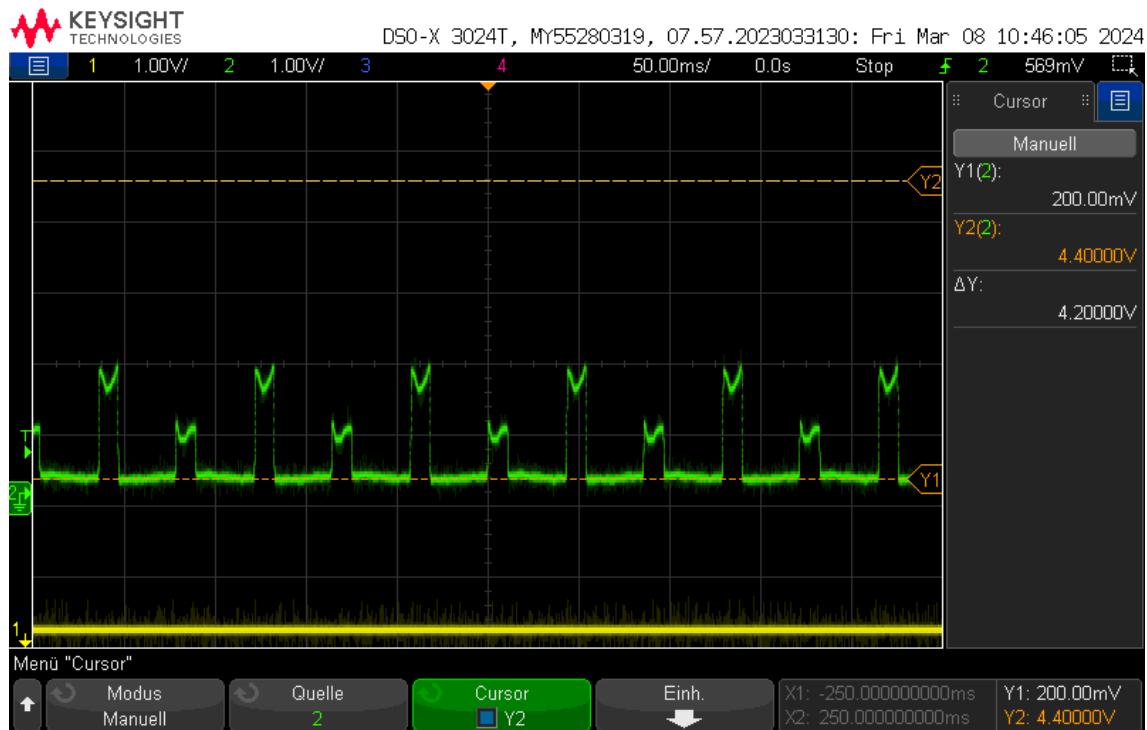


Abbildung 27: 700 1/min und $f = 1000$ Hz

3 Schichten		9 Schichten	
Digitalwert	Abstand (mm)	Digitalwert	Abstand (mm)
11620	4,785	21365	4,318

Wie es scheint ist der Sensor an seine Grenzen geraten. Da der sleep an die Abtastfrequenz angepasst wurde, werden nun wieder viele Werte für die gleiche Höhe ausgegeben. Die Datenrate ist sichtlich zu hoch und somit kann die Log-Datei nicht schnell genug erweitert werden. Es empfiehlt sich daher nicht eine so hohe Frequenz zum Abtasten zu wählen.

Für uns erweckt es den Anschein, als dass $f = 100$ Hz, sleep = 0,01 s einen guten Wert darstellt.

5 Zusammenfassung und Fazit

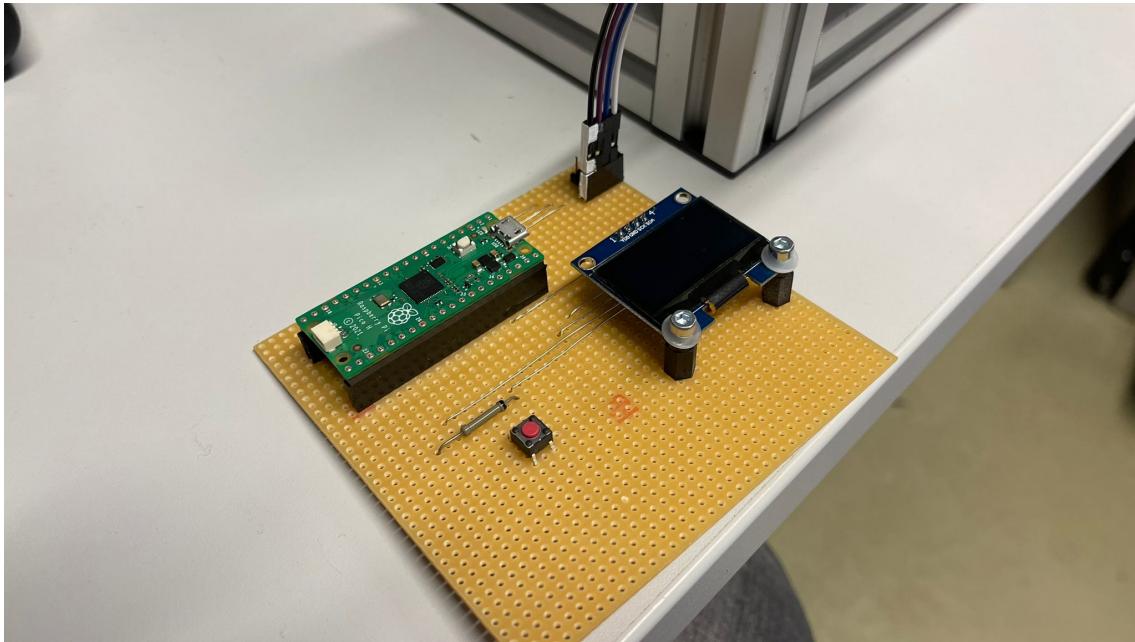


Abbildung 28: Endgültige Platine mit weißer Markierung für Verpolungsschutz

5.1 Abstand

Der im Datenblatt gegebene Arbeitsbereich von $\pm 6,5\text{mm}$ wurde bereits anhand der ersten Experimente widerlegt. Wir haben mit Hilfe der ausgegebenen Digitalwerte einen Arbeitsbereich von etwa 2,65 mm nachweisen können.

Wir haben also festgestellt, dass man maximal $(2,55 \pm 0,15) \text{ mm}$ an das zu messende Objekt herankommen kann und nicht weiter als $(5 \pm 0,15) \text{ mm}$ weg sein darf, da der Sensor sonst unverwertbare Signale sendet. Es wurde auch nicht der volle Umfang der 16 bit, also 65535 Werte ausgereizt, sondern nur etwa 53208 Werte (56683 - 3475). Natürlich ist beim Messen des Abstandes ein relativ hoher Ablesefehler vorhanden, dennoch ist es deutlich zu sehen (siehe Oszilloskop & Excel-Tabelle), dass hier die Grenzen des Sensors liegen. Wichtig ist aber auch die Verwendung von weißem Papier, da sich schnell gezeigt hat, dass stark glänzende Flächen mit einer hohen Fehlerabweichung einhergehen.

5.2 Frequenz

Die Abtastfrequenz des Sensors hat ebenfalls einen direkten Einfluss auf das Messergebnis. Wenn sich die Drehscheibe schneller dreht, als der Sensor die Höhenänderungen erfasst, sind unplausiblere Ergebnisse zu erwarten. Wir haben daher die Frequenz des Abtastens immer genau auf, oder

meistens sogar über, die Drehfrequenz der Scheibe angepasst.

Man kann sagen, dass der Sensor eine Drehung der Scheibe von 700 min^{-1} problemlos wahrnehmen kann. Dies haben wir deutlich auf dem Oszilloskop erkennen können. Jedoch muss man sehr darauf achten, dass die zu messende Fläche möglichst eben ist, da dies ansonsten zur Verfälschung der Ergebnisse führt. Wir gehen davon aus, dass der Sensor eine Abstandsänderung mit einer Frequenz von 24 Hz problemlos messen kann, wenn die idealen Bedingungen (weißes Papier und ebenes Messobjekt) eingehalten werden. Die 23 Hz ergeben sich aus dem Zusammenhang, dass eine Umdrehung von 700 min^{-1} einer Frequenz von etwa 11,7 Hz entspricht. Da wir aber zwei Kreisfragmente besitzen, ist die Abtastfrequenz mit 24 Hz zu wählen.

5.3 Anwendung

Im Allgemeinen lässt sich feststellen, dass der Sensor für seinen vorgesehenen Einsatz durchaus geeignet ist. Die Ausgabe der Digitalwerte erfolgt zwar nicht ganz so schnell und reibungslos wie gewollt, jedoch reicht die Umwandlung der Messwerte, in einen Abstand, für eine Momentanwertanzeige aus. Ebenso ist der Datenlogger sinnvoll, da somit im Nachhinein bestimmte Messpunkte überprüft werden können. Für eine genauere Auswertung empfiehlt es sich ein Oszilloskop anzuschließen, da dieses die Messwerte genauer und weniger verrauscht anzeigt. Der Versuchsaufbau gerät in dieser Hinsicht an seine Grenzen und ist nur wenig weiter zu verbessern. Eine Erklärung dafür ist der nicht-ideale Aufbau unserer Platine. Das Oszilloskop ist ein hochpräzises Messinstrument, welches die möglichen Fehlerquellen eliminiert (diese herausrechnet). Unser Versuchsaufbau ist diesen Fehlerquellen natürlich unterlegen. So können Magnetfelder, Leitungsinduktion und die Spannungsfälle über die Leitungen allesamt Einfluss auf das Messergebnis nehmen. Ebenso ist unser Spannungsteiler so konzipiert, dass wir einen rein linearen Zusammenhang zwischen den Größen vorliegen haben. In der Praxis ist dies natürlich äußerst untypisch. Dennoch sind die Werte, welche wir damit ermitteln, genau genug für unsere Anwendungen. Falls eine wirklich genaue Messung erforderlich ist, so muss man eben auf die dementsprechend genauen Messinstrumente zurückgreifen. Außerdem empfehlen wir die Verwendung von einem Tiefpass, um das Rauschen, welches die Messergebnisse markant beeinflusst, einzudämmen.

6 Anhang

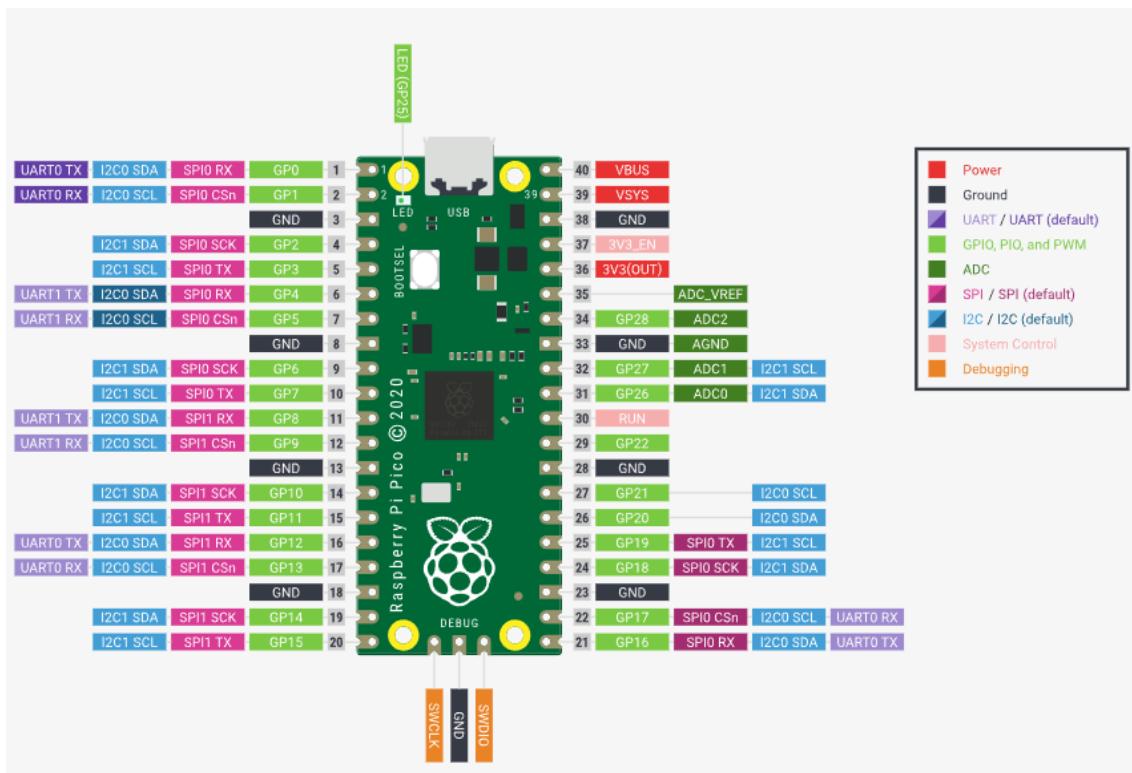
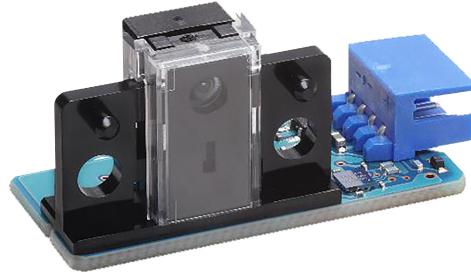


Abbildung 29: Pinbelegung Raspberry Pi Pico H

Micro-displacement Sensor

- Enable to detect $\pm 10 \mu\text{m}$ level displacement, applicable to detect duplicated paper feeding or deflection amount.
- Displacement output changes by object colors are stabilized with signal divider circuit.
- Operating area = $6.5 \pm 1 \text{ mm}$.

⚠ Be sure to read Safety Precautions on page 3.



Ordering Information

Micro-displacement Sensor

Appearance	Sensing method	Connecting method	Sensing distance	Output type	Model
	Triangulation	Connector	$6.5 \pm 1 \text{ mm}$	Analog output	Z4D-C01

Ratings, Characteristics and Exterior Specifications

Absolute Maximum Ratings ($T_a = 25^\circ\text{C}$)

Item	Symbol	Rated value	Unit	Remarks
Supply voltage	V _{cc}	7	V	---
LED pulse light emission control signal	PLS	7	V	LED
LED light emission pulse	t _{fP}	100	ms	Please refer Pulsed Forward Current Rated Curve
Operating temperature	T _{opr}	-10 to +65	°C	No freezing or condensation
Storage temperature	T _{stg}	-25 to +80	°C	---

Characteristics ($T_a = 25^\circ\text{C}$)

Object: N8.5 Munsell paper with a reflection factor of 70%.

Item	Value
Operating area *1	$6.5 \pm 1 \text{ mm}$
Sensitivity variation *2	$-1.4 \text{ mV}/\mu\text{m} \pm 10\% \text{ max.}$
Resolution *3	$\pm 10 \mu\text{m} \text{ max.}$
Linearity *4	2% F.S. (full scale) max.

*1. Distance from Mounting Reference Plane to Target.
 *2. The sensitivity is defined as slope of the line and it represents the variation in the output voltage per unit length between different products.

$$\text{Sensitivity variation} = \frac{V_1 - V_0}{2000} (\text{mV}/\mu\text{m}) \quad \begin{aligned} V_0: & \text{Output voltage at the point } d_0 \\ V_1: & \text{Output voltage at the point } d_1 \\ d_0, d_1: & \text{Distance between datum clamp face and detective object} \end{aligned}$$

d ₀	5.5 mm
d ₁	7.5 mm

*3. This is the value of the electrical noise width in the output signal converted to a distance under the following conditions.

(1) Noise width is measured in a waveform after the four times average with an oscilloscope.

(2) Ripple noise in the power supply voltage (V_{cc}): 10 mVp-p max.
 (3) Distance from mounting reference plane to target: 6.5 mm
 (4) Resolution is measured by the voltage of response delay time (t_{r2}).

*4. This is the peak-to-peak value of the deviation of the signal output from a straight line.

A linearity of 2% F.S. indicates the following value:

(1) Distance full-scale converted value: $2 \text{ mm} \times 0.02 = 0.04 \text{ mm}$ (40 μm)
 (2) Output voltage converted value: $1.4 \text{ mV}/\mu\text{m} \times 40 \mu\text{m} = 56 \text{ mV}$
 (for a sensor with a sensitivity of 1.4 mV/ μm)

Exterior Specification

Connecting method	Weight (g)	Material		
		Case	Cover	Lens
Connector	3.6	Polycarbonate	Polycarbonate	PMMA

Electrical and Optical Characteristics ($T_a = -10^\circ\text{C}$ to 65°C)

Item	Symbol	Rated value	Remarks
Power supply voltage	V _{cc}	$5 \text{ V} \pm 10\%$	Ripple (p-p): 10 mV p-p max.
Output voltage	OUT	0.2 V to (V _{cc} - 0.3) V	*1
Response delay time	t _{r1}	100 μs max.	*2
	t _{r2}	500 μs max.	*3
LED pulse light emission control signal voltage	PLS	3.5 V to V _{cc}	---

*1. Load impedance (between OUT-GND) is set at more than 10 k Ω .

*2. t_{r1}: Rise time of output voltage from 10% to 90%.

*3. t_{r2}: Setup time from PLS rising edge to measured output value rising edge.

Engineering Data (Reference value)

Fig 1. Operating Distance Characteristics (Typical)

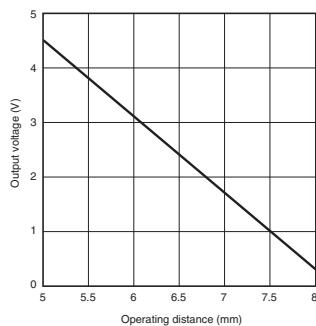


Fig 2. Dependency of Object on Reflection Factor (Typical)

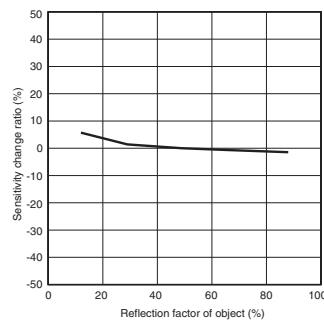


Fig 3. Temperature Characteristics (Typical)

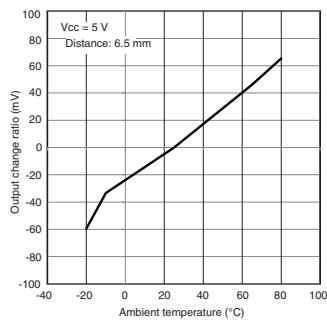
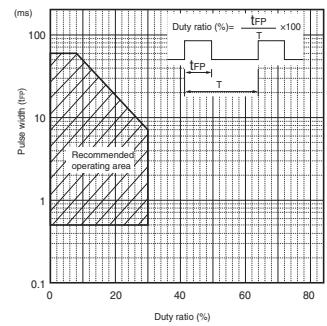
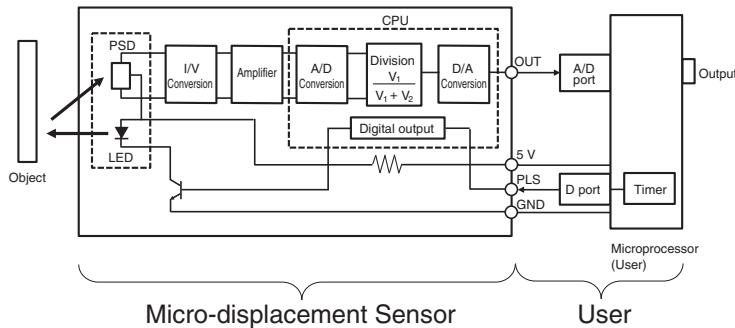


Fig 4. Pulsed Forward Current Rated Curve



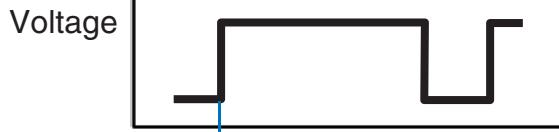
Circuit diagram/Connection diagram



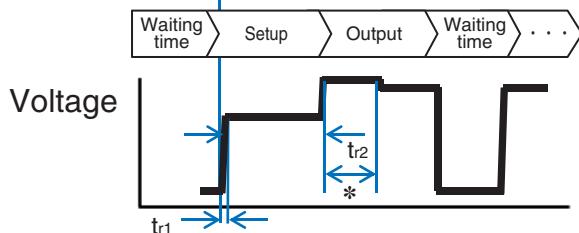
- Output voltage signal is obtained by applying the pulse signal at PLS terminal. Please be careful, it is not obtained by DC voltage applying.

Pulse Input and Sensor Output

<Pulse Input>



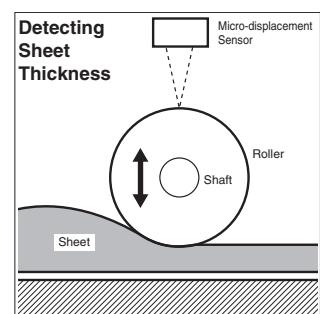
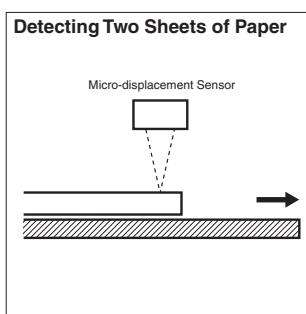
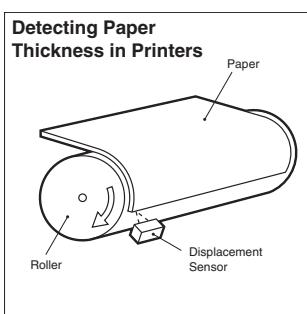
<Sensor Output>



- Output voltage depends on input. After setup time t_{r2} , output signal is obtained according to the distance.
- After setup, output signal is updated every 80 microseconds.

* Update every 80 microseconds. Output signal is reset at pulse input signal falling low level, and start setup over again after next input signal applying.

Application Examples



Safety Precautions

To ensure safe operation, be sure to read and follow the Instruction Manual provided with the Sensor.

CAUTION

This product is not designed or rated for ensuring safety of persons either directly or indirectly. Do not use it for such purposes.



Precautions for Safe Use

Do not use the product with a voltage or current that exceeds the rated range.

Applying a voltage or current that is higher than the rated range may result in explosion or fire.

Do not miswire such as the polarity of the power supply voltage.

Otherwise the product may be damaged or it may burn.

This product does not resist water. Do not use the product in places where water or oil may be sprayed onto the product.

Precautions for Correct Use

Do not use the product in atmospheres or environments that exceed product ratings.

This product is for surface mounting. Refer to Soldering Information, Storage and Baking for details.

Dispose of this product as industrial waste.

Because the resolution of this sensor is very small, output error may occur due to noise from the power supply.

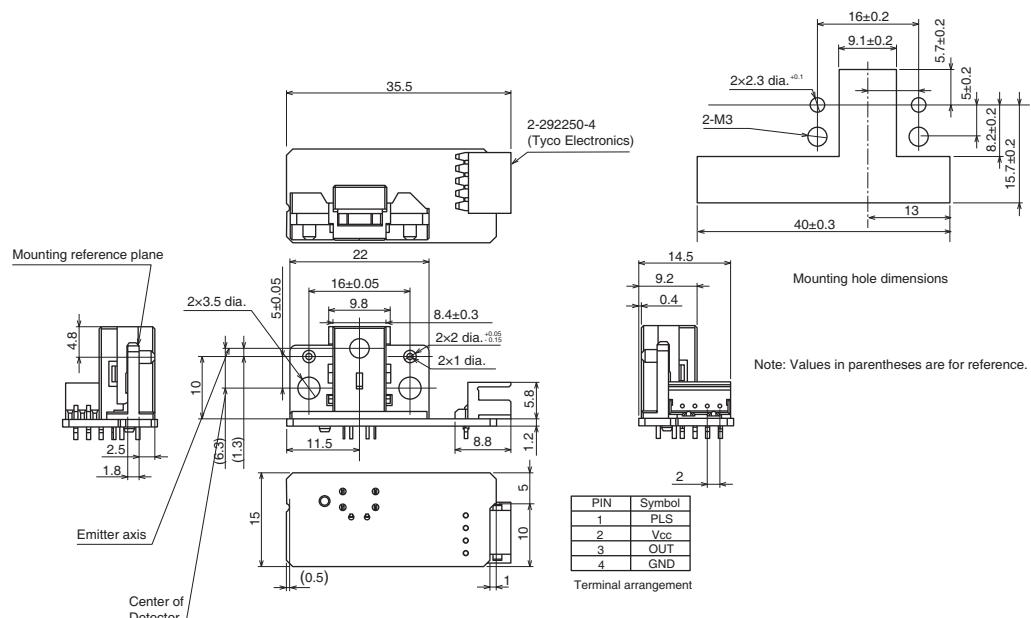
The ripple noise of the power supply should be 10mVp-P or less. Add averaging and filters as needed to reduce the effects of noise.

Dimensions

(Unit: mm)

Micro-displacement Sensor

Z4D-C01



7 Materialliste

- Mikrocontroller Raspberry Pi Pico
- 1,3 Zoll OLED I₂C 128 x 64 Pixel Display
- 75 k Ω und 40 k Ω Widerstände
- Anschlussbuchsen für Jumperwires
- Reset-Knopf
- OMRON Z4D-C01 Abstandssensor
- Pico Breadboard
- USB-A zu micro-USB Leitung
- Jumperwires
- Teststation bestehend aus:
 - Gestell aus Aluminium
 - Frequenzumrichter (Combivert G6 der Firma *KEB Automation KG*)
 - Motor (Firma: KEB, 3-phasisig, 0,37 kW, Dreieck 230 V/ 1,73 A, Stern 400 V/ 1,00 A, $\cos \phi = 0,71$; $u = 1410$ 1/min)
 - Bedienfeld für die leichte Bedienung des Frequenzumrichters
 - dazugehörige Verkabelung
 - herkömmliches Druckerpapier