



Hochschule für Technik,
Wirtschaft und Kultur Leipzig

FAKULTÄT INGENIEURWISSENSCHAFTEN

5420 - EMBEDDED SYSTEMS

Dokumentation 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

1	Vorwort	1
2	Datenblatt des Sensors	1
3	Pinout RPi Pico H	5
4	Schaltplan	5
5	Programm	6
6	Anwendung	8
7	Materialliste	10

1 Vorwort

Die nachfolgende Dokumentation soll eine Hilfe dafür sein, den Sensor in Betrieb zu nehmen und einen Abstand zu messen.

2 Datenblatt des Sensors

Microphotonic Devices (Micro-displacement Sensor)

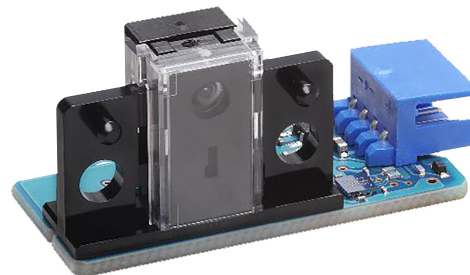
Z4D-C01

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 = $6.5 \pm 1 \text{ mm}$.

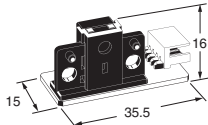


Be sure to read *Safety Precautions* on page 3.



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	$^\circ\text{C}$	No freezing or condensation
Storage temperature	T_{stg}	-25 to +80	$^\circ\text{C}$	---

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.

$$\text{Sensitivity variation} = \frac{V_1 - V_0}{2000} \quad (\text{mV}/\mu\text{m})$$

V_0 : Output voltage at the point d_0
 V_1 : Output voltage at the point d_1
 d_0, d_1 : Distance between datum clamp face and detective object

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 $1.4 \text{ mV}/\mu\text{m}$)

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 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)

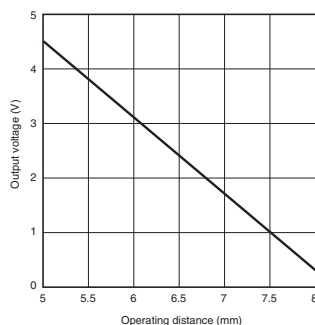


Fig 2. Dependency of Object on Reflection Factor (Typical)

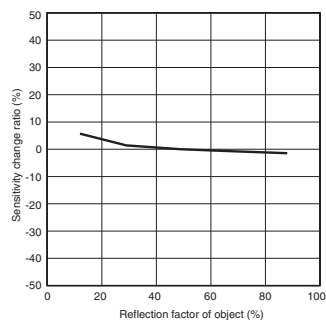


Fig 3. Temperature Characteristics (Typical)

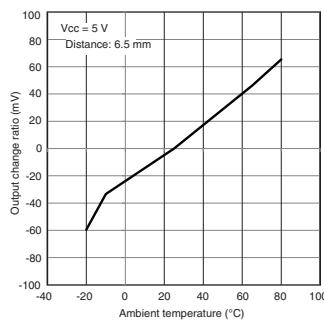
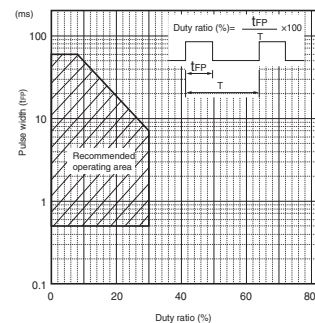
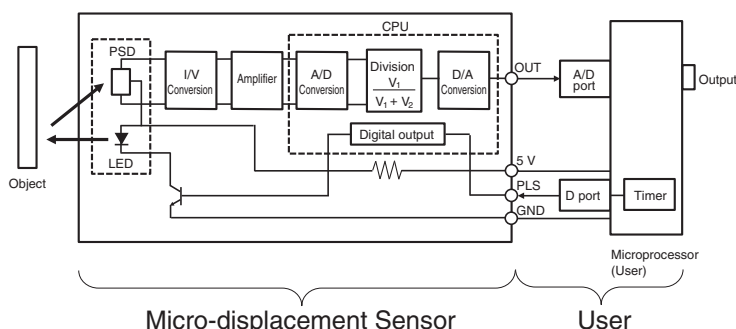


Fig 4. Pulsed Forward Current Rated Curve



Circuit diagram/Connection diagram



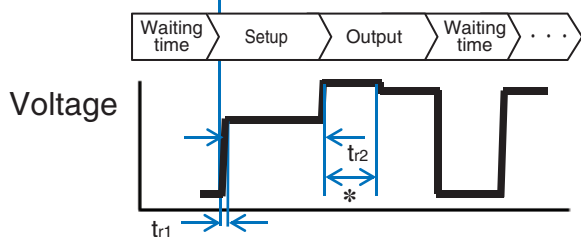
- Output voltage signal is obtained by applying the pulse signal at PLS terminal. Please be careful, it is not obtained by DC voltage applying.

Pulse Input and Sensor Output

<Pulse Input>



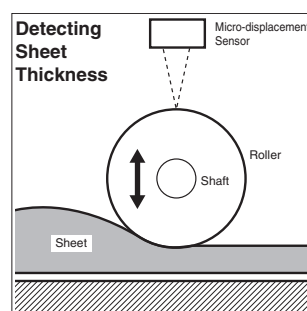
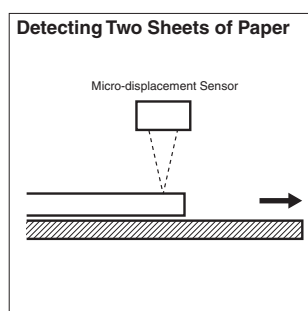
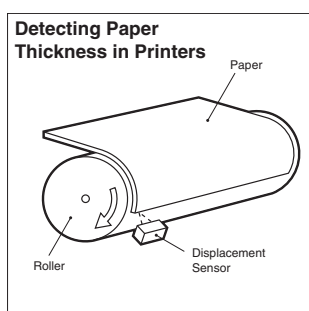
<Sensor Output>



- 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 micro seconds.

* Update every 80 micro seconds. 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.

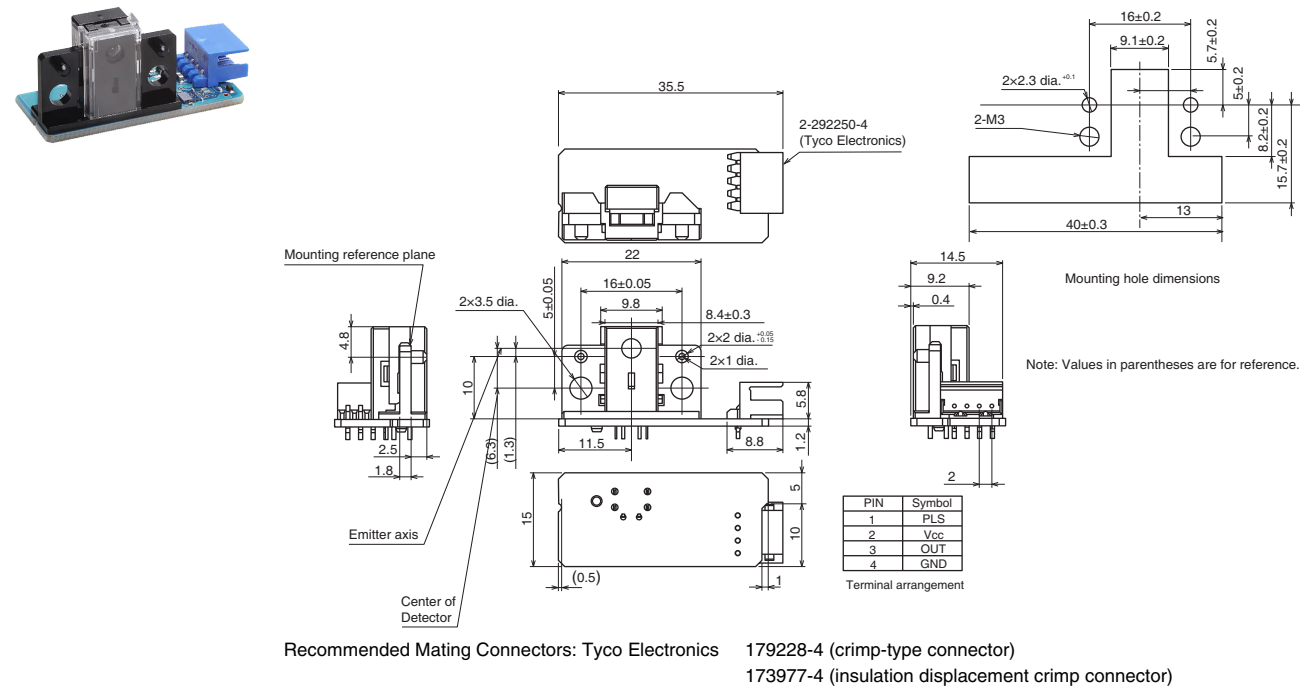
The ripple noise of the power supply should be 10mVp-P or less. Add averaging and filters as needed to reduce the effects of noise.

Dimensions

(Unit: mm)

Micro-displacement Sensor

Z4D-C01

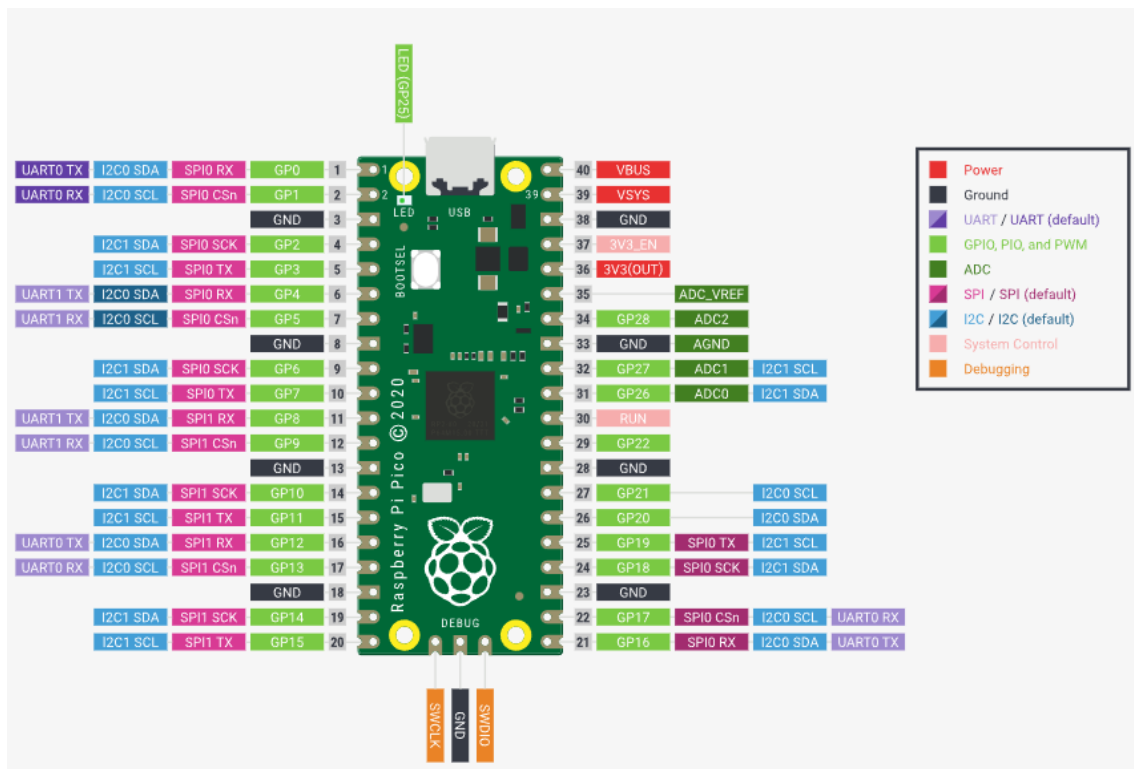


Der Omron Z4D-C01 Abstandssensor besitzt 4 Anschlüsse:

- 1 = PLS (blau): hier wird der benötigte Puls angelegt, welcher die Infrarot LED arbeiten lässt
- 2 = V_{cc} (schwarz): hier wird die Versorgungsspannung angelegt, die der Sensor benötigt
- 3 = OUT (weiß): hier werden die analogen Daten (eine Spannung) ausgegeben, die der Sensor erfasst
- 4 = GND (lila) (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. Als Programmiersprache nutzen wir Python.

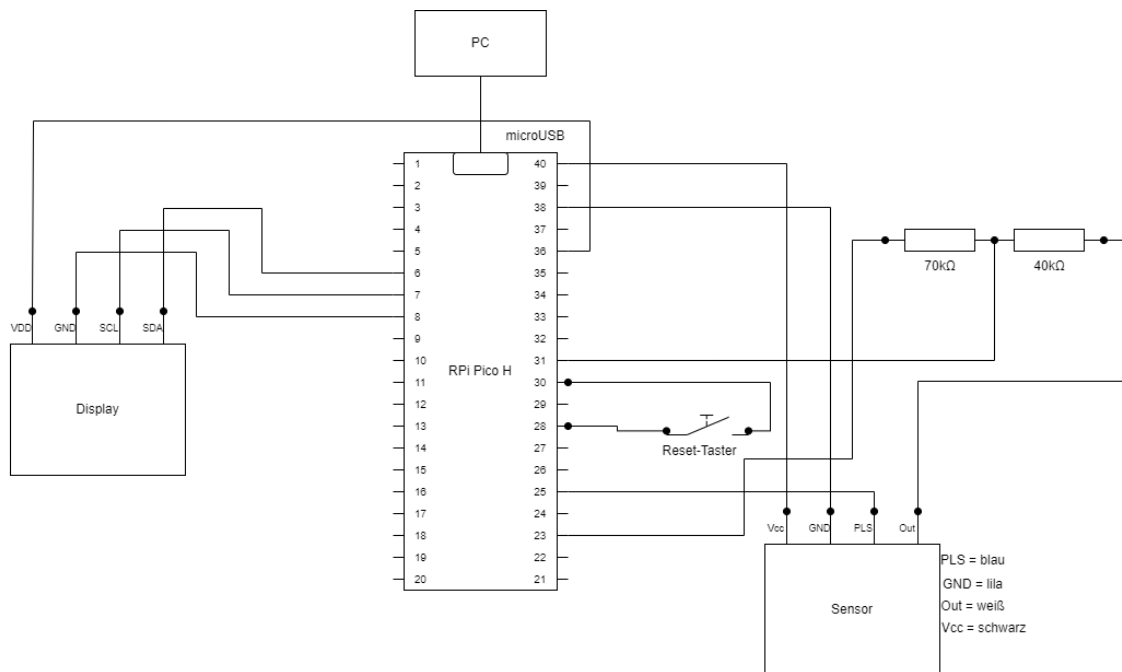
3 Pinout RPi Pico H



4 Schaltplan

Es ist darauf zu achten, dass der ADC Pin des Raspberry Pi Pico nur maximal 3,3 V aufnehmen kann. Somit ist ein geeigneter Spannungsteiler **vor** dem ADC pin und **nach** dem 'Out' des Sensors

zu schalten. In diesem Fall wurde $40k\Omega$ und $70k\Omega$ benutzt um das Ausgangssignal des Sensors einzuschränken (siehe Schaltplan).



5 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.

Der Output ist im Verzeichnis des RPi zu finden, unter dem Namen 'data.csv'. Dieses wird bei jedem Programmneustart überschrieben.

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

#I2C betriebebenenes SH1106 OLED-Display
from machine import I2C, ADC
from sh1106 import SH1106_I2C
import framebuffer

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

```

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

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

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 (l/r, o/u)
    #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()
```

6 Anwendung

Wenn man nun das obige Programm und die Betriebsmittel nach dem Schaltplan anschließt und aufbaut, ist man in der Lage mit jenem Abstandssensor einen Abstand zu messen, welcher jedoch stark von den Angaben des Datenblatts abweicht.

- 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 nur einen Arbeitsbereich von etwa 2,65 mm nachweisen können.
- Wir haben festgestellt, dass man auch nur maximal $(2,55 \pm 0,15)$ mm an das zu messende Objekt herankommt und nicht weiter als $(5,2 \pm 0,15)$ mm weg darf, da der Sensor dann unverwertbare Werte ausgibt.
- Es wurde auch nicht der volle Umfang der 16 bit, also 65535 Werte ausgereizt, sondern nur

etwa 53208 Werte (56683 - 3475).

- Wichtig ist hier aber auch die Verwendung von weißem (Drucker-)Papier, da sich schnell gezeigt hat, dass stark glänzende Flächen mit einer hohen Fehlerabweichung einhergehen.
- ebenfalls empfiehlt es sich als Messpunkt eine möglichst ebene Fläche zu wählen
- Für eine genaue Analyse empfiehlt es sich ein Oszilloskop heranzuziehen. Der Sensor samt Display ist eher eine Momentanwertanzeige bzw. zum Aufzeichnen von Daten geeignet.

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-phasig, 0,37 kW, Dreieck 230 V/ 1,73 A, Stern 400 V/ 1,00 A, $\cos \phi = 0,71$; $n = 1410$ 1/min)
 - Bedienfeld für die leichte Bedienung des Frequenzumrichters
 - dazugehörige Verkabelung
 - herkömmliches Druckerpapier