



Betriebsanleitung
eddyNCDT 3020

ES-U1-C-CAx/mB0 ES-S1-C-CAx/mB0 ES-S2-C-CAx/mB0 ES-U3-T-CAx/mB0 ES-U6-C-CAx/mB0
ES-U1-T-CAx/mB0 ES-U2-C-CAx/mB0 ES-U3-C-CAx/mB0 ES-S4-C-CAx/mB0 ES-U8-C-CAx/mB0

Berührungsloses Kompakt-Wegmesssystem auf Wirbelstrombasis

MICRO-EPSILON MESSTECHNIK
GmbH & Co. KG
Königbacher Str. 15

94496 Ortenburg / Deutschland

Tel: +49 (0) 8542 / 168-0
Fax: +49 (0) 8542 / 168-90
info@micro-epsilon.de
<https://www.micro-epsilon.de>

Inhaltsverzeichnis

1	Sicherheit.....	5
1.1	Verwendete Zeichen.....	5
1.2	Warnhinweise.....	5
1.3	Hinweise zur Produktkennzeichnung.....	5
1.3.1	CE-Kennzeichnung.....	5
1.3.2	UKCA-Kennzeichnung.....	5
1.4	Bestimmungsgemäße Verwendung.....	6
1.5	Bestimmungsgemäßes Umfeld.....	6
2	Funktionsprinzip, Technische Daten.....	7
2.1	Anwendungsgebiet.....	7
2.2	Messprinzip.....	7
2.3	Aufbau des kompletten Messsystems.....	7
2.4	Begriffsdefinition, Analogausgang Weg.....	8
2.5	Technische Daten.....	8
2.5.1	DT3020.....	8
2.5.2	Sensoren DT3020.....	9
3	Lieferung.....	11
3.1	Lieferumfang.....	11
3.2	Lagerung.....	11
4	Installation und Montage.....	12
4.1	Allgemein.....	12
4.1.1	Modelle.....	12
4.1.2	Messbereichsanfang.....	12
4.2	Einbausituation Sensor.....	12
4.2.1	Standardmontage.....	13
4.2.2	Flächenbündige Montage.....	15
4.3	Messaufbau, Einsatz mehrerer Sensoren.....	15
4.4	Maßzeichnungen Sensoren.....	16
4.5	Sensorkabel.....	18
4.6	Maßzeichnung Controller.....	20
4.7	Messobjektgröße.....	20
4.8	Elektrische Anschlüsse.....	21
4.8.1	Anschlussmöglichkeiten.....	21
4.8.2	Anschlussbelegung.....	21
4.8.3	Versorgungsspannung.....	21
4.8.4	Analogausgang, Weg.....	22
4.8.5	Schaltausgang.....	23
5	Inbetriebnahme über die Software sensorTOOL.....	24
5.1	Beschreibung, Download.....	24
5.2	Messsystemaufbau prüfen.....	24
5.3	Messobjekt platzieren.....	24
5.4	Singlesensor Mode.....	25
5.5	Multisensor Mode.....	25
5.6	Baudrate einstellen.....	26
5.6.1	Baudrate konfigurieren.....	26
5.6.2	Baudrate ändern.....	26
5.7	Menüführung.....	27
6	Datenaufnahme im sensorTOOL.....	29
6.1	Datenaufnahme.....	29
6.2	Signalverarbeitung.....	30
6.2.1	Gleitender Mittelwert.....	30
6.2.2	Gleitender Median.....	31
6.2.3	Rekursiver Mittelwert.....	32
6.2.4	Trigger, Unterabtastung, Master.....	32
6.3	Messung aufzeichnen und speichern.....	33
7	Einzelwert im sensorTOOL.....	35
8	Einstellungen am Controller.....	36
8.1	Allgemeine Informationen zu den Controller-Einstellungen.....	36

8.2	Feldlinearisierung.....	36
8.2.1	Skalierung des Messbereichs.....	36
8.2.2	Feldlinearisierung.....	37
8.2.2.1	1-Punkt (Nullpunktverschiebung).....	38
8.2.2.2	2-Punkt (Nullpunktverschiebung, Empfindlichkeit).....	39
8.2.2.3	3-Punkt (Nullpunktverschiebung, Empfindlichkeit, Linearität).....	40
8.3	Schaltausgang: Grenzwertüberwachung.....	41
8.3.1	Detektor-Einstellungen.....	41
8.3.1.1	Absolutwert.....	41
8.3.1.2	Peak to peak.....	41
8.3.1.3	Dynamisch.....	42
8.3.2	Grenzwerttyp-Einstellungen.....	43
8.3.2.1	Grenzwertüberwachung.....	43
8.3.2.2	Fenster.....	43
8.3.2.3	Hysterese.....	44
8.3.3	Zeitbedingungs-Einstellungen.....	44
8.3.4	Ausgangstyp-Einstellungen.....	45
8.4	Status Bits: Temperatur Warnschwellen.....	47
8.5	Filter und Messraten.....	47
8.5.1	Filter und Mittelung im DT3020.....	47
8.5.2	Datenübertragung RS485.....	49
8.5.3	Datenübertragungsdiagnose sensorTOOL.....	50
9	Info im sensorTOOL.....	51
10	Config Parameter.....	52
10.1	Feldlinearisierung mit Config Parametern durchführen.....	52
10.2	Signalverarbeitung: Unterabtastung.....	54
10.3	Liste verfügbarer Config Parameter.....	54
10.4	Digitalwerte.....	60
10.4.1	Ausgabedatenformate.....	60
10.4.2	Berechnung des Abstandswertes.....	61
10.4.2.1	int16 distance (Config Parameter 754 = 0x70, 0x73, 0x74).....	62
10.4.2.2	int32 distance (Config Parameter 754 = 0x71, 0x72).....	62
10.4.3	Berechnung der Sensor- und Elektroniktemperatur.....	63
10.5	Bedeutung des Status.....	63
11	Softwareunterstützung mittels MEDAQLib.....	64
11.1	Config Parameter lesen und schreiben.....	64
11.2	Messwerte lesen.....	65
11.3	Feldlinearisierung.....	65
12	Diagnosemeldungen.....	66
12.1	Fehlercodes.....	66
12.2	Warnungen.....	67
13	Fehlerbehebung.....	68
14	Haftungsausschluss.....	69
15	Service, Reparatur.....	70
16	Außerbetriebnahme, Entsorgung.....	71
17	Anhang.....	72
17.1	Optionales Zubehör.....	72
17.2	Modellbezeichnung Sensor.....	73
17.3	Modellbezeichnung Sensorkabel.....	73
17.4	Stabilität gegenüber Störeinstrahlung.....	73
17.5	Softwarebeispiele für gängige Programmiersprachen.....	74
	Index.....	75

1 Sicherheit

1.1 Verwendete Zeichen

Die Systemhandhabung setzt die Kenntnis der Betriebsanleitung voraus.

In dieser Betriebsanleitung werden folgende Bezeichnungen verwendet:

 VORSICHT	Zeigt eine gefährliche Situation an, die zu geringfügigen oder mittelschweren Verletzungen führt, falls diese nicht vermieden wird.
HINWEIS	Zeigt eine Situation an, die zu Sachschäden führen kann, falls diese nicht vermieden wird.
▶	Zeigt eine ausführende Tätigkeit an.
i	Zeigt einen Anwendertipp an.
Messung	Zeigt eine Hardware oder eine(n) Schaltfläche/Menüeintrag in der Software an.

1.2 Warnhinweise

 VORSICHT	<p>Schließen Sie die Spannungsversorgung nach den Vorschriften für elektrische Betriebsmittel an.</p> <ul style="list-style-type: none"> • Verletzungsgefahr • Beschädigung oder Zerstörung des Sensors und/oder des Controllers
HINWEIS	<p>Vermeiden Sie Stöße und Schläge auf den Sensor und auf den Controller.</p> <ul style="list-style-type: none"> • Beschädigung oder Zerstörung des Sensors und/oder des Controllers • Versorgungsspannung darf angegebene Grenzen nicht überschreiten. • Beschädigung oder Zerstörung des Sensors und/oder des Controllers <p>Schützen Sie das Sensorkabel vor Beschädigung.</p> <ul style="list-style-type: none"> • Zerstörung des Sensors • Ausfall des Messgerätes

1.3 Hinweise zur Produktkennzeichnung

1.3.1 CE-Kennzeichnung

Für das Produkt gilt:

- Richtlinie 2014/30/EU („EMV“)
- Richtlinie 2011/65/EU („RoHS“)

Produkte, die das CE-Kennzeichen tragen, erfüllen die Anforderungen der zitierten EU-Richtlinien und der jeweils anwendbaren harmonisierten europäischen Normen (EN).

Das Produkt ist ausgelegt für den Einsatz im Industrie- und Laborbereich.

Die EU-Konformitätserklärung und die technischen Unterlagen werden gemäß den EU-Richtlinien für die zuständigen Behörden bereitgehalten.

1.3.2 UKCA-Kennzeichnung

Für das Produkt gilt:

- SI 2016 No. 1091 („EMC“)
- SI 2016 No. 1101 („Low Voltage“)
- SI 2012 No. 3032 („RoHS“)

Produkte, die das UKCA-Kennzeichen tragen, erfüllen die Anforderungen der zitierten Richtlinien und der jeweils anwendbaren Normen.

Das Produkt ist ausgelegt für den Einsatz im Industrie- und Laborbereich.

Die UKCA-Konformitätserklärung und die technischen Unterlagen werden gemäß der UKCA-Richtlinien für die zuständigen Behörden bereitgehalten.

1.4 Bestimmungsgemäße Verwendung

Eingesetzt wird das DT3020 zur

- Weg-, Abstands-, Dicken- und Verschiebungsmessung
- Positionsmessung von Bauteilen oder Maschinenkomponenten

Das Messsystem ist für den Einsatz im Industrie- und Laborbereich konzipiert.

- ▶ Setzen Sie das Messsystem so ein, dass bei Fehlfunktionen oder Totalausfall des Sensors keine Personen gefährdet oder Maschinen beschädigt werden.
- ▶ Treffen Sie bei sicherheitsbezogener Anwendung zusätzlich Vorkehrungen für die Sicherheit und zur Schadensverhütung.

1.5 Bestimmungsgemäßes Umfeld

Schutzart:	
- Sensor, Sensorkabel:	IP68 (angeschlossen)
- Controller:	IP67 (gesteckt)
Temperaturbereich Sensor, Sensorkabel:	
- Betrieb:	-20 ... +180 °C, gültig für ES-U1, ES-U1-T -20 ... +200 °C
- Lager:	-20 ... +180 °C, gültig für ES-U1, ES-U1-T -50 ... +200 °C
Temperaturbereich Controller:	
- Betrieb:	-20 ... 105 °C (nicht kondensierend)
- Lager:	-20 ... 105 °C (nicht kondensierend)
Luftfeuchtigkeit:	5 % RH ... 95 % RH (nicht kondensierend)
Umgebungsdruck:	Atmosphärendruck

2 Funktionsprinzip, Technische Daten

2.1 Anwendungsgebiet

Das berührungslos arbeitende Kompaktwegmesssystem ist konzipiert für den industriellen Einsatz in Produktionsanlagen, zur Maschinenüberwachung und zum Messen und Prüfen in der Inline-Qualitätssicherung.

2.2 Messprinzip

Das Wegmesssystem eddyNCDT (Non-Contacting Displacement Transducers) arbeitet berührungslos auf Wirbelstrombasis. Es wird für Messungen an Objekten aus elektrisch leitenden Werkstoffen verwendet, die ferromagnetische und nichtferromagnetische Eigenschaften haben können. Hochfrequenter Wechselstrom durchfließt eine in ein Sensorgehäuse montierte Spule. Das elektromagnetische Spulenfeld induziert im leitfähigen Messobjekt Wirbelströme, dadurch ändert sich der Wechselstromwiderstand der Spule. Diese Impedanzänderung liefert ein elektrisches Signal, proportional zum Abstand des Messobjekts vom Sensor. Ein elektronisches Kompensationsverfahren reduziert temperaturabhängige Messfehler auf ein Minimum.

2.3 Aufbau des kompletten Messsystems

Das berührungslos arbeitende Einkanal-Wegmesssystem besteht aus:

- Sensor
- Sensorkabel
- Anschlusskabel
- Controller
- Werksabgleich



i Die Komponenten sind durch einen werksseitigen Abgleich aufeinander abgestimmt. Die Zuordnung von Sensor und Controller bestimmt die Serien-Nummer. Die abgestimmten Komponenten und Seriennummern sind mit Labels seitlich am Controller gekennzeichnet.



eddyNCDT 3020 Controller und Sensoren

2.4 Begriffsdefinition, Analogausgang Weg

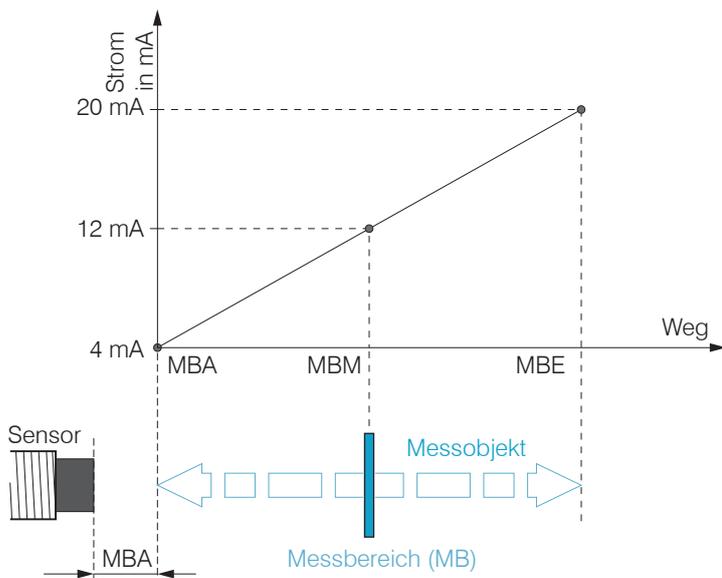


Abb. 2.1: Messbereich und Ausgangssignal Messsystem

- MBA Messbereichsanfang
Minimaler Abstand zwischen Sensorstirnfläche und Messobjekt, sensorspezifisch
- MBM Messbereichsmitte
- MBE Messbereichsende (Messbereichsanfang + Messbereich)
Maximaler Abstand zwischen Sensorstirnfläche und Messobjekt
- MB Messbereich

2.5 Technische Daten

2.5.1 DT3020

Modell		DT3020
Auflösung ^[1]	statisch	0,004 % d.M.
	dynamisch	0,01 % d.M.
Grenzfrequenz (-3dB) ^[2]		9-stufig einstellbar: 10 Hz ... 5 kHz
Messrate	Analogausgang	80 kSa/s
	Digitalausgang	10 kSa/s
Linearität ^[3]		< ±0,2 % d.M.
Temperaturstabilität ^[4]		< 0,025 % d.M. / K
Temperaturkompensation		10 ... 105 °C
Messobjektmaterial ^[5]		Stahl, Aluminium
Anzahl Kennlinien		1
Versorgungsspannung		12 ... 32 VDC
Leistungsaufnahme		< 1,7 W
Digitale Schnittstelle ^[6]		RS485 / USB / Ethernet / EtherCAT / PROFINET / EtherNet/IP
Analogausgang		4 ... 20 mA (max. 500 Ω Bürde, frei skalierbar 0 ... 20 mA)

[1] d.M. = des Messbereichs, RMS Rauschen bezogen auf Messbereichsmitte, statisch: 20 Hz, dynamisch: 5 kHz

[2] Werkseinstellung 5 kHz

[3] Wert gültig mit 3-Punkt-Linearisierung

[4] Wert gültig im temperaturkompensierten Bereich

[5] Stahl: St37 1.0037; Aluminium: AlMg3 3.3535

[6] Für USB, Ethernet, EtherCAT, PROFINET und EtherNet/IP ist die Anbindung über ein Schnittstellenmodul erforderlich

Modell		DT3020	
Schaltausgang		wählbar: NPN, PNP, Push-Pull	
Anschluss		Sensor: Steckverbinder triaxiale Buchse; Versorgung/Signal: Steckverbinder 8-polig M12	
Montage		Durchgangsbohrungen (Ø 4,4 mm)	
Temperaturbereich	Lagerung	-20 ... 105 °C (nicht kondensierend)	
	Betrieb	-20 ... 105 °C (nicht kondensierend)	
Schock (DIN EN 60068-2-27)		15 g / 6 ms in 3 Achsen, je 2 Richtungen und je 1000 Schocks	
Vibration (DIN EN 60068-2-6)		5 g / 10 ... 500 Hz in 3 Achsen, je 2 Richtungen und je 10 Zyklen	
Schutzart (DIN EN 60529)		IP67 (gesteckt)	
Material		Alu-Druckguss	
Gewicht		ca. 190 g	
Bedien- und Anzeigeelemente ^[7]		Konfigurierbar über sensorTOOL Software: 3-Punkt-Linearisierung, Skalierung des Analogausgangs, Filter & Mittelung, Schnittstellenauswahl	

2.5.2 Sensoren DT3020

Modell		ES-U1	ES-U1-T	ES-S1	ES-U2	ES-S2
Messbereich		1 mm		1 mm	2 mm	2 mm
Messbereichsanfang		0,1 mm		0,1 mm	0,2 mm	0,2 mm
Auflösung ^[8]		0,04 µm		0,04 µm	0,08 µm	0,08 µm
Linearität ^[9]		< ±2 µm		< ±2 µm	< ±4 µm	< ±4 µm
Temperaturstabilität ^[10]		< 0,25 µm / K		< 0,25 µm / K	< 0,5 µm / K	< 0,5 µm / K
Temperaturkompensation		10 ... 180 °C				
Sensortyp		ungeschirmt		geschirmt	ungeschirmt	geschirmt
Empfohlene Größe Messobjekt (flach)		Ø 18 mm		Ø 12 mm	Ø 24 mm	Ø 18 mm
Anschluss ^[11]		integriertes Kabel, axial, Standardlänge 3 m; 1 m, 6 m, 9 m optional				
Montage		Verschraubung (M6x0,5)	Klemm-Montage (Ø 6 mm)	Verschraubung (M8x1)	Verschraubung (M8x1)	Verschraubung (M12x1)
Temperaturbereich	Lagerung	-20 ... 180 °C		-20 ... 200 °C		
	Betrieb	-20 ... 180 °C		-20 ... 200 °C		
Druckbeständigkeit		20 bar front- und rückseitig				
Schock (DIN EN 60068-2-27)		15 g / 6 ms in 3 Achsen, je 2 Richtungen und je 1000 Schocks				
Vibration (DIN EN 60068-2-6)		15 g / 49,85 ... 2000 Hz in 3 Achsen ±3 mm / 10 ... 49,85 Hz in 3 Achsen				
Schutzart (DIN EN 60529)		IP68 (gesteckt)				
Material		Edelstahl und Kunststoff				
Gewicht ^[12]		ca. 2,4 g		ca. 2,4 g	ca. 4,7 g	ca. 11 g

[7] Zugriff auf sensorTOOL erfordert Anschluss an PC über Schnittstellenmodul

[8] RMS-Wert des Signalrauschens, statisch (20 Hz); gültig bei Betrieb mit DT3020; bezogen auf den nominalen Messbereich und in Messbereichsmittle; im kompensierten Temperaturbereich

[9] Gültig bei Betrieb mit DT3020 und 3-Punkt-Linearisierung; bezogen auf den nominalen Messbereich

[10] Gültig bei Betrieb mit DT3020; bezogen auf den nominalen Messbereich und in Messbereichsmittle; im kompensierten Temperaturbereich

[11] Längentoleranz Kabel: Nominalwert 0 % / + 30 %

[12] Gewicht nur Sensor ohne Muttern, ohne Kabel

Modell		ES-U3	ES-U3-T	ES-S4	ES-U6	ES-U8
Messbereich		3 mm		4 mm	6 mm	8 mm
Messbereichsanfang		0,3 mm		0,4 mm	0,6 mm	0,8 mm
Auflösung ^[8]		0,12 µm		0,16 µm	0,24 µm	0,32 µm
Linearität ^[9]		< ±6 µm		< ±8 µm	< ±12 µm	< ±16 µm
Temperaturstabilität ^[10]		< 0,75 µm / K		< 1 µm / K	< 1,5 µm / K	< 2 µm / K
Temperaturkompensation		10 ... 180 °C				
Sensortyp		ungeschirmt		geschirmt	ungeschirmt	ungeschirmt
Empfohlene Größe Messobjekt (flach)		Ø 36 mm		Ø 27 mm	Ø 54 mm	Ø 72 mm
Anschluss ^[11]		integriertes Kabel, axial, Standardlänge 3 m; 1 m, 6 m, 9 m optional				
Montage		Verschraubung (M12x1)	Klemm-Montage (Ø 12 mm)	Verschraubung (M18x1)	Verschraubung (M18x1)	Verschraubung (M24x1,5)
Temperaturbereich	Lagerung	-20 ... 200 °C				
	Betrieb	-20 ... 200 °C				
Druckbeständigkeit		20 bar front- und rückseitig				
Schock (DIN EN 60068-2-27)		15 g / 6 ms in 3 Achsen, je 2 Richtungen und je 1000 Schocks				
Vibration (DIN EN 60068-2-6)		15 g / 49,85 ... 2000 Hz in 3 Achsen ±3 mm / 10 ... 49,85 Hz in 3 Achsen				
Schutzart (DIN EN 60529)		IP68 (gesteckt)				
Material		Edelstahl und Kunststoff				
Gewicht ^[12]		ca. 12 g		ca. 30 g	ca. 33 g	ca. 62 g

[8] RMS-Wert des Signalrauschens, statisch (20 Hz); gültig bei Betrieb mit DT3020; bezogen auf den nominalen Messbereich und in Messbereichsmittle; im kompensierten Temperaturbereich

[9] Gültig bei Betrieb mit DT3020 und 3-Punkt-Linearisierung; bezogen auf den nominalen Messbereich

[10] Gültig bei Betrieb mit DT3020; bezogen auf den nominalen Messbereich und in Messbereichsmittle; im kompensierten Temperaturbereich

[11] Längentoleranz Kabel: Nominalwert 0 % / + 30 %

[12] Gewicht nur Sensor ohne Muttern, ohne Kabel

3 Lieferung

3.1 Lieferumfang

- 1 Sensor inkl. Sensorkabel
 - 1 Controller inkl. Werksabgleich
 - 1 Prüfprotokoll
 - 1 Montageanleitung
- ▶ Nehmen Sie die Teile des Messsystems vorsichtig aus der Verpackung und transportieren Sie sie so weiter, dass keine Beschädigungen auftreten können.
 - ▶ Prüfen Sie die Lieferung nach dem Auspacken sofort auf Vollständigkeit und Transportschäden.
 - ▶ Wenden Sie sich bitte bei Schäden oder Unvollständigkeit sofort an den Hersteller oder Lieferanten.

Rücknahme von Verpackungen

Die Micro-Epsilon Messtechnik GmbH & Co. KG bietet Kunden die Möglichkeit, Verpackung von Produkten, die sie bei Micro-Epsilon erworben haben, nach vorheriger Abstimmung zurückzugeben, damit diese der Wiederverwendung oder einer Verwertung (Recycling) zugeführt werden kann.

Um die Rückgabe von Verpackung zu veranlassen, bei Fragen zu den Kosten und / oder dem genauen Ablauf der Rücknahme, wenden sie sich bitte direkt an

info@micro-epsilon.de

3.2 Lagerung

Temperaturbereich:	-20 ... 105 °C (nicht kondensierend)
- Sensoren	-20 ... +180 °C, gültig für ES-U1, ES-U1-T -20 ... +200 °C
- Controller	-20 ... 105 °C (nicht kondensierend)
Luftfeuchtigkeit:	5 % RH ... 95 % RH (nicht kondensierend)

4 Installation und Montage

4.1 Allgemein

Auf die Kabelmantel von Sensor-, Versorgungs- und Ausgangskabel durfen keine scharfkantigen oder schweren Gegenstande einwirken.

i Ein beschadigtes Kabel kann nicht repariert werden. Zugkraft am Kabel ist unzulassig!

4.1.1 Modelle

Das Messsystem eddyNCDT wird mit ungeschirmten oder geschirmten Sensoren eingesetzt.



Abb. 4.1: Ungeschirmte Sensoren mit Gewinde (links), ohne Gewinde (Mitte), geschirmte Sensoren (rechts)

Ungeschirmte Sensoren

- Typenbezeichnung: ES-Ux
- Aufbau: Die Sensorkappe mit eingebetteter Spule besteht aus elektrisch nichtleitenden Werkstoffen.

i In radialer Richtung konnen Metallteile in der Nahe wie ein Messobjekt wirken und das Messergebnis verfalschen. Beachten Sie dies bei der Auswahl des Materials fur die Sensormontage und deren Aufbau.

Geschirmte Sensoren

- Typenbezeichnung: ES-Sx
- Aufbau: Der Sensor ist bis zur Stirnflache von einem Metallgehause mit Montagegewinde umgeben. Der Sensor ist damit gegen die Beeinflussung durch radial nahegelegene Metallteile besser abgeschirmt.

4.1.2 Messbereichsanfang

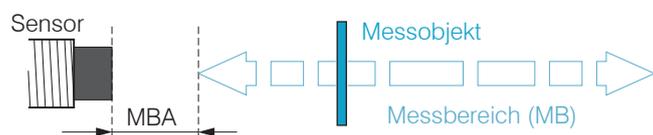


Abb. 4.2: Messbereichsanfang (MBA), der minimale Abstand zwischen Sensorstirnflache und Messobjekt

Fur jeden Sensor muss ein minimaler Abstand zum Messobjekt eingehalten werden. Damit wird eine Messunsicherheit durch Andruck des Sensors an das Messobjekt und mechanische Zerstorung des Sensors/Messobjektes vermieden. Der Messbereichsanfang betragt im Standardwerksabgleich 10 % des nominellen Messbereichs.

4.2 Einbausituation Sensor

Wirbelstrom-Wegsensoren konnen in ihrem Messverhalten von einer metallischen Halterung beeinflusst werden. Bevorzugen Sie die Sensormontage entsprechend dem verwendeten Sensortyp.

4.2.1 Standardmontage

Die Sensoren ragen über die metallische Halterung hinaus. Die dargestellte Einbausituation wird zur Werkskalibrierung der Sensoren bei Micro-Epsilon verwendet. Die Technischen Daten der Sensoren beziehen sich auf die Standardmontage. Für das Erreichen der im Datenblatt angegebenen Werte wird daher empfohlen, den Sensor in gleicher Weise einzubauen wie dies während dessen Kalibrierung der Fall war.

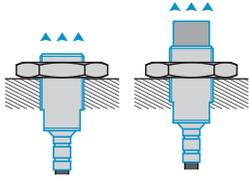


Abb. 4.3: eddyNCDT Standardmontage

Sensoren mit Gewinde

- ▶ Stecken Sie den Sensor durch die Bohrung in der Sensorhalterung.
- ▶ Schrauben Sie den Sensor fest.
- ▶ Drehen Sie dazu die Montagemuttern aus dem Lieferumfang beidseitig über das aus der Halterung ragende Gewinde.
- ▶ Ziehen Sie die Montagemuttern vorsichtig an, um Beschädigungen, vor allem der kleineren Sensoren, zu vermeiden.

i Bevorzugen Sie die Standardmontage des Sensors, da Sie mit dieser Methode optimale Messergebnisse erzielen! Halten Sie bei der Kalibrierung dieselbe relative Position des Sensors zur Halterung wie bei der Messung ein. Abweichende Einbausituationen können mittels Feldlinearisierung zum Teil kompensiert werden.

Bei der werksseitigen Kalibrierung der Sensoren befindet sich die Sensorstirnfläche in einem definierten Abstand A zur Montagemutter. Für die Anwendung ist dieser Abstand A zu berücksichtigen, um eine maximale Linearität zu erreichen.

Ungeschirmter Sensor mit Gewinde in Standardmontage

Ungeschirmter Sensor mit Gewinde in Standardmontage	Sensor	Maß A
	ES-U1	8 mm
	ES-U2	8 mm
	ES-U3	10 mm
	ES-U6	20,4 mm
	ES-U8	24,6 mm

Tab. 4.1:

Geschirmter Sensor mit Gewinde in Standardmontage

Geschirmter Sensor mit Gewinde in Standardmontage	Sensor	Maß A
	ES-S1	4 mm
	ES-S2	4 mm
	ES-S4	4 mm

Tab. 4.2:

Klemmbare Sensoren ohne Gewinde

- ▶ Montieren Sie Sensoren ohne Gewinde vorzugsweise mit einer Umfangsklemmung. Montieren Sie die Sensoren alternativ mit einem Gewindestift.

Umfangsklemmung mit Spannzange

Diese Art der Sensormontage bietet die höchste Zuverlässigkeit, da der Sensor über sein zylindrisches Gehäuse flächig geklemmt wird. Sie ist bei schwierigen Einbaumgebungen, zum Beispiel an Maschinen und Produktionsanlagen zwingend erforderlich.

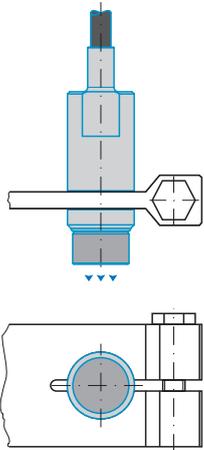


Abb. 4.4: Montage mit Spannzange

Radiale Punktklemmung mit Gewindestift

Diese einfache Befestigungsart ist nur bei kraft- und vibrationsfreiem Einbauort zu empfehlen. Die Punktklemmung kann mittels Gewindestift oder Druckschraube erfolgen.

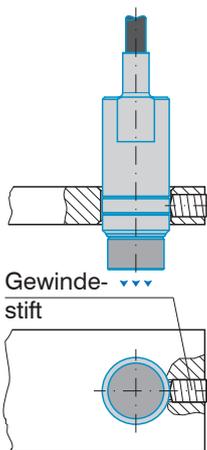


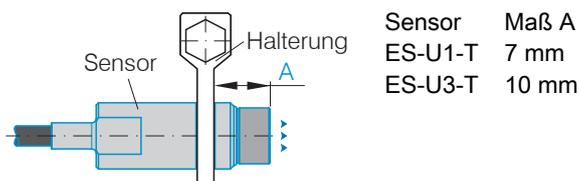
Abb. 4.5: Montage mit Gewindestift

Hinweis

► Achten Sie darauf das Sensorgehäuse nicht zu beschädigen!

Abstand Sensorstirnfläche und Halterung Sensoren ohne Gewinde in Standardmontage

Bei der werksseitigen Kalibrierung der Sensoren befindet sich die Sensorstirnfläche in einem definierten Abstand A zur Sensorhalterung. Für die Anwendung ist dieser Abstand A zu berücksichtigen, um eine maximale Linearität zu erreichen.



4.2.2 Flächenbündige Montage

Die flächenbündige Montage entspricht nicht der Werkskalibrierung. Micro-Epsilon empfiehlt eine 3-Punkt-Feldlinearisierung durchzuführen.

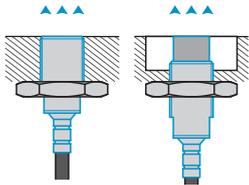


Abb. 4.6: Bündige Montage

- i Linearisieren Sie das Messsystem, wenn möglich in einer exakten Messanordnung wie in der späteren Messung selbst!

Sensoren mit Gewinde

- ▶ Montieren Sie geschirmte oder ungeschirmte Sensoren bündig in die Sensorhalterung aus Isoliermaterial (Kunststoff, Keramik etc.).
- ▶ Montieren Sie ungeschirmte Sensoren bündig in die metallische Sensorhalterung. Achten Sie dabei auf eine Aussparung an der Halterung in der Größe des dreifachen Sensordurchmessers.
- ▶ Montieren Sie geschirmte Sensoren bündig in die metallische Sensorhalterung.
- ▶ Drehen Sie die Sensoren in allen Montagefällen in die Gewindebohrung und kontern Sie die Sensoren mit der Montagemutter.
- ▶ Ziehen Sie diese vorsichtig an, um Beschädigungen, vor allem der kleineren Sensoren, zu vermeiden.

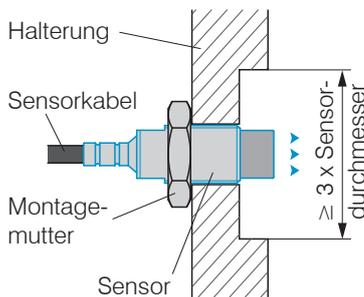


Abb. 4.7: Flächenbündige Montage eines ungeschirmten Sensors in einer metallischen Halterung mit Aussparung

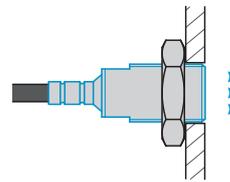


Abb. 4.8: Flächenbündige Montage eines geschirmten Sensors in einer metallischen Halterung

4.3 Messaufbau, Einsatz mehrerer Sensoren

Wirbelstromsensoren erzeugen magnetische Felder, die sich überlagern können, wenn die Sensoren zu nahe aneinander platziert werden (sogenanntes Cross-Talk). Um dies zu vermeiden, gibt es zwei Lösungen:

- Montage mit ausreichend Mindestabstand
- Montage von Sensoren mit unterschiedlichen Frequenzen, LF (low frequency) und HF (high frequency)



Abb. 4.9: Sensor ES-Ux, ungeschirmt

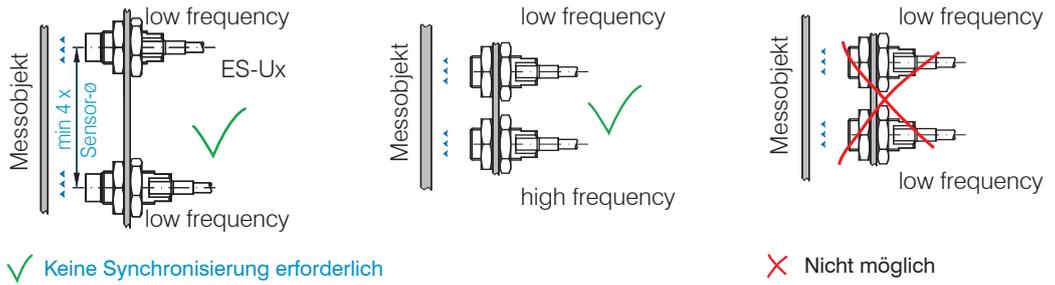
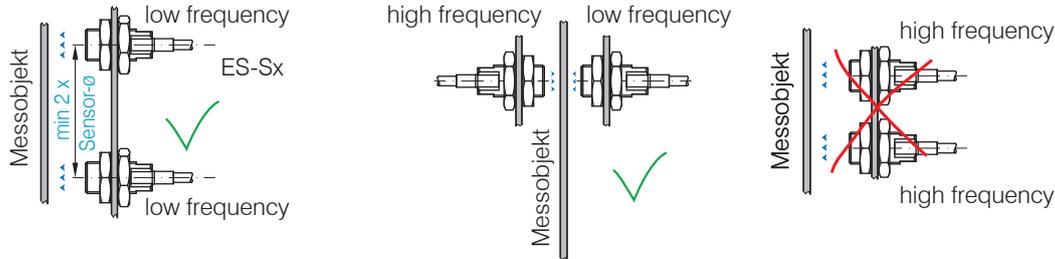


Abb. 4.10: Sensor ES-Sx, geschirmt



Beim Betrieb mehrerer Messsysteme können diese mit einer Frequenztrennung (LF/HF) geliefert werden. Die Frequenztrennung ermöglicht einen Mehrkanalbetrieb ohne gegenseitige Beeinflussung. Mit dieser Funktion ist eine Synchronisation nicht erforderlich.

Bei mehr als zwei Sensoren ist die abwechselnde Reihenfolge LF-HF-LF-HF- ... bzw. HF-LF-HF-LF- ... zu beachten. Die Wahl von LF oder HF Sensoren betrifft nur die Frequenz des elektrischen Feldes und hat keine Auswirkung auf die Genauigkeit, die max. Grenzfrequenz oder Messrate des Controllers.

4.4 Maßzeichnungen Sensoren

Messrichtung

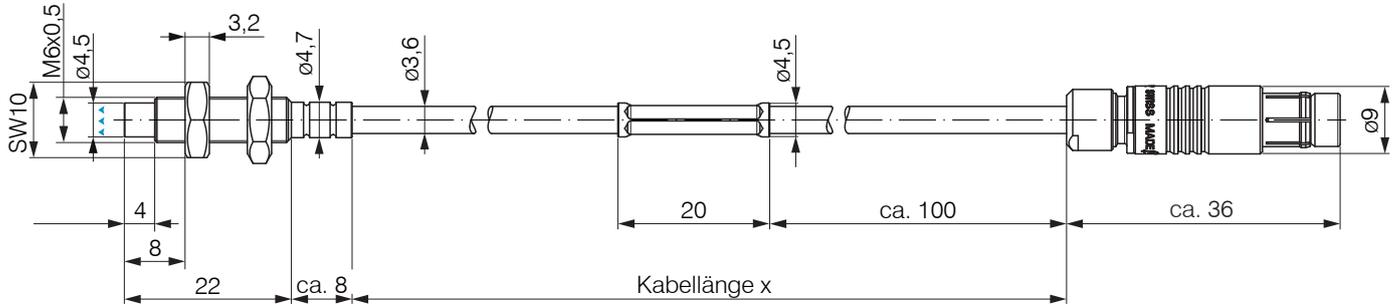


Abb. 4.11: Maßzeichnung Sensoren ES-U1-C-CAX/mB0, Abmessung in mm

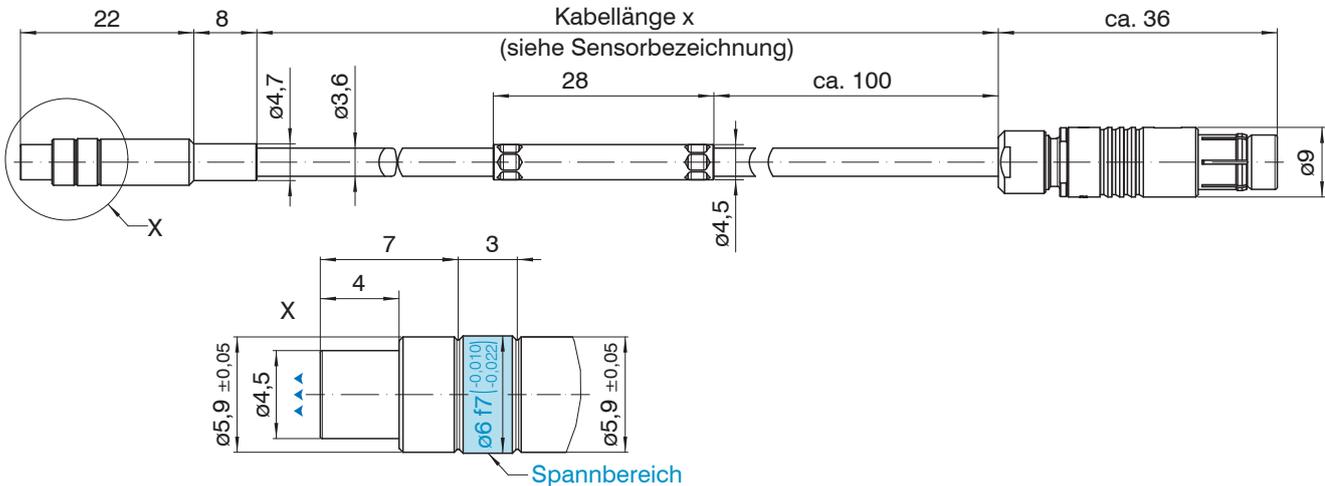


Abb. 4.12: Maßzeichnung Sensoren ES-U1-T-CAX/mB0, Abmessung in mm

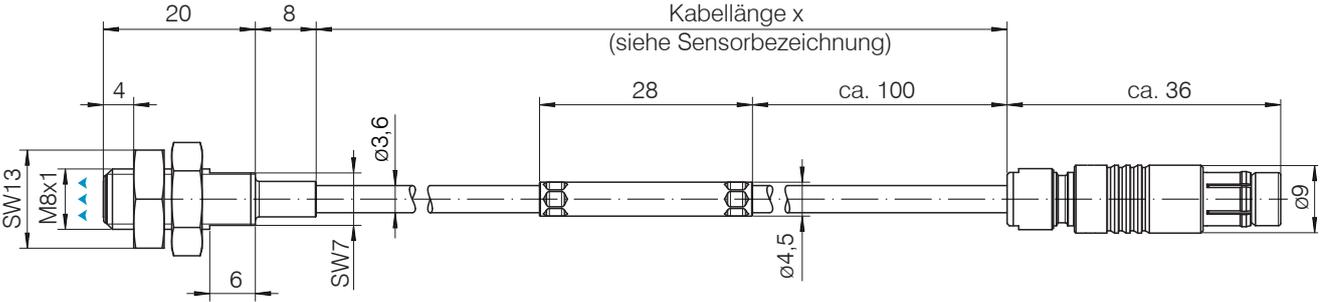


Abb. 4.13: Maßzeichnung Sensoren ES-S1-C-CAX/mB0, Abmessung in mm

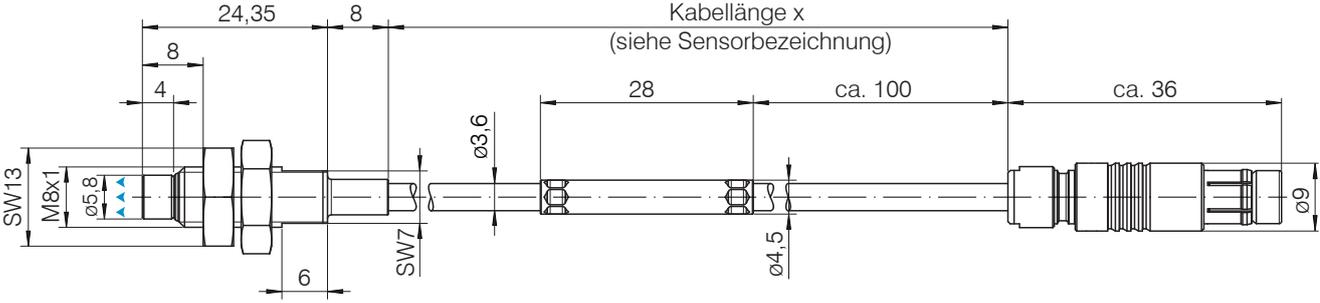


Abb. 4.14: Maßzeichnung Sensoren ES-U2-C-CAX/mB0, Abmessung in mm

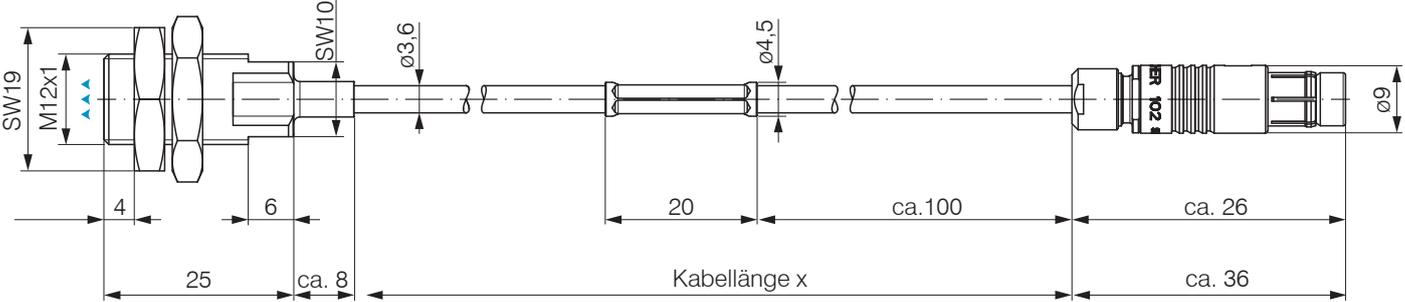


Abb. 4.15: Maßzeichnung Sensoren ES-S2-C-CAX/mB0, Abmessung in mm

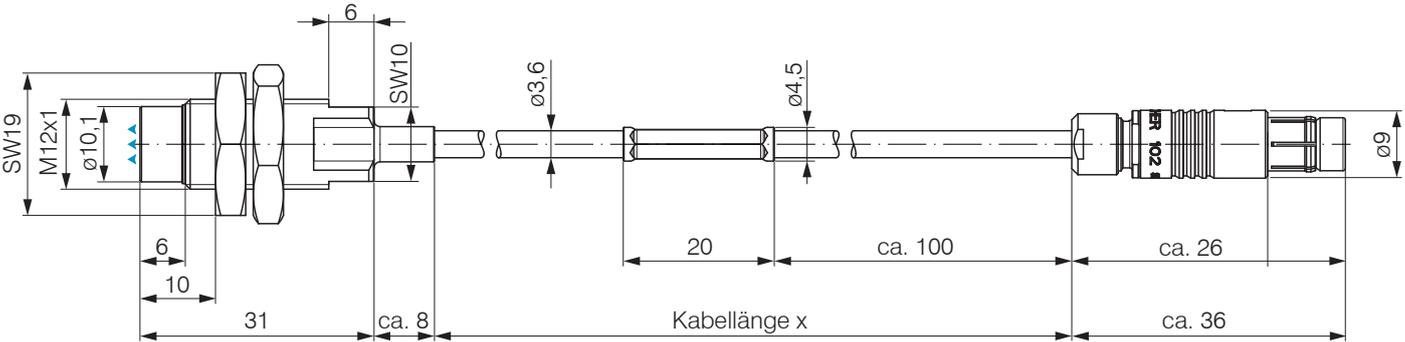


Abb. 4.16: Maßzeichnung Sensoren ES-U3-C-CAX/mB0, Abmessung in mm

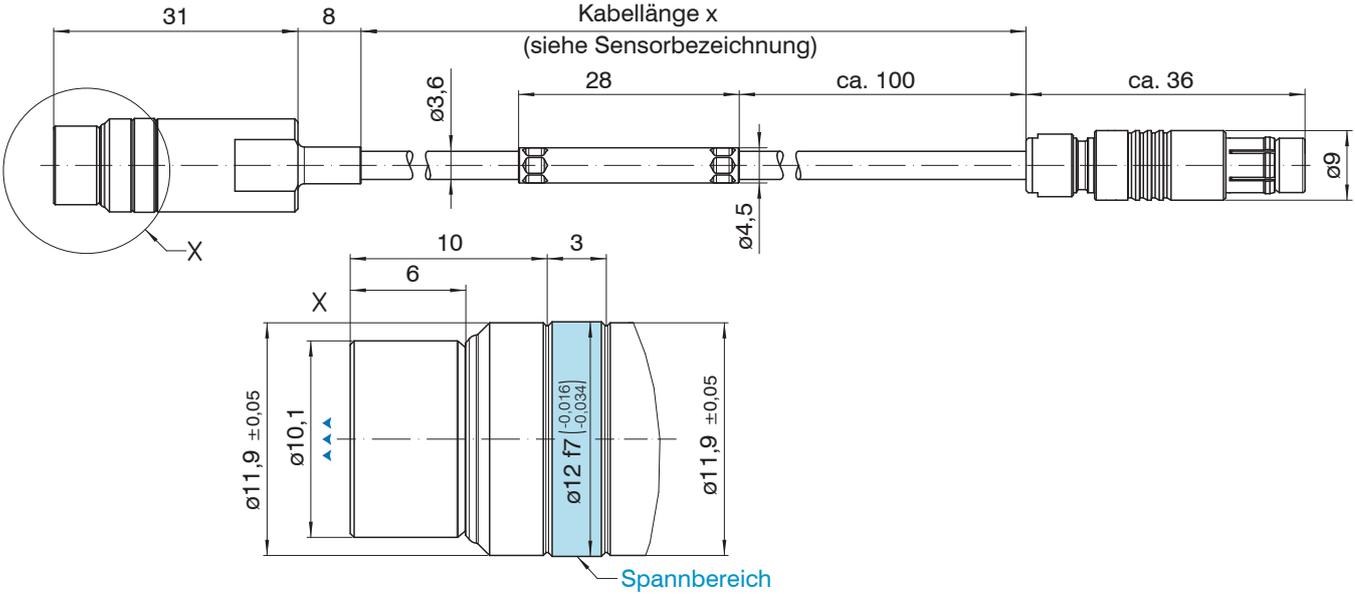


Abb. 4.17: Maßzeichnung Sensoren ES-U3-T-CAx/mB0, Abmessung in mm

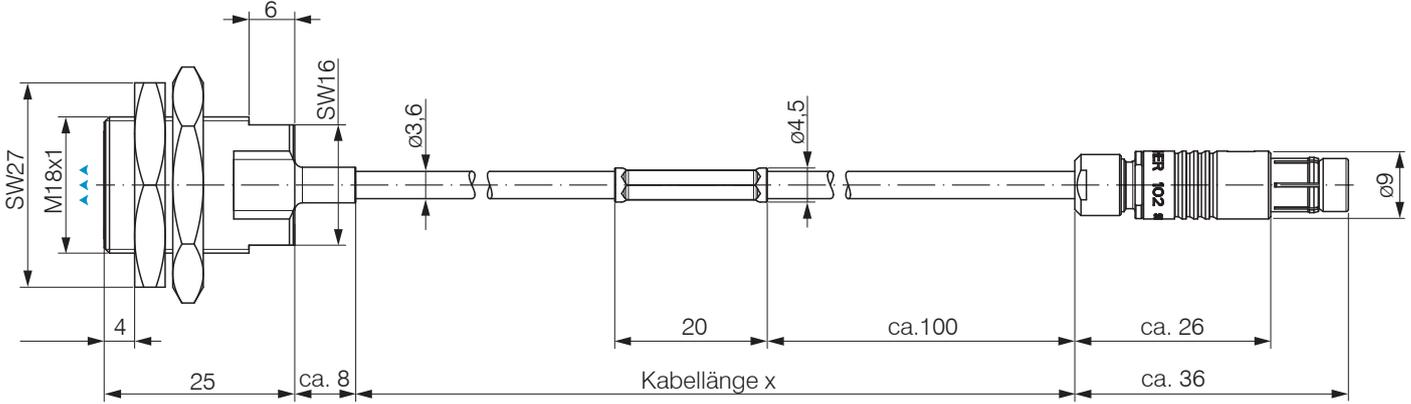


Abb. 4.18: Maßzeichnung Sensoren ES-S4-C-CAx/mB0, Abmessung in mm

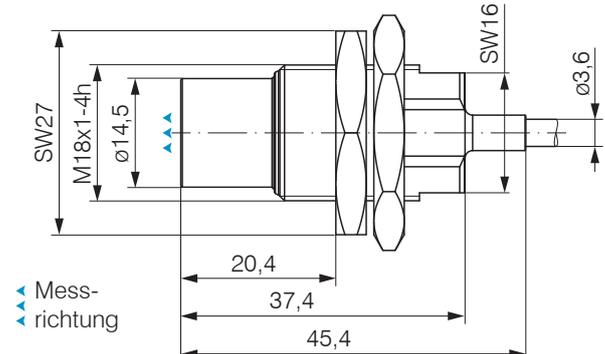


Abb. 4.19: Maßzeichnung Sensoren ES-U6-C-CAx/mB0

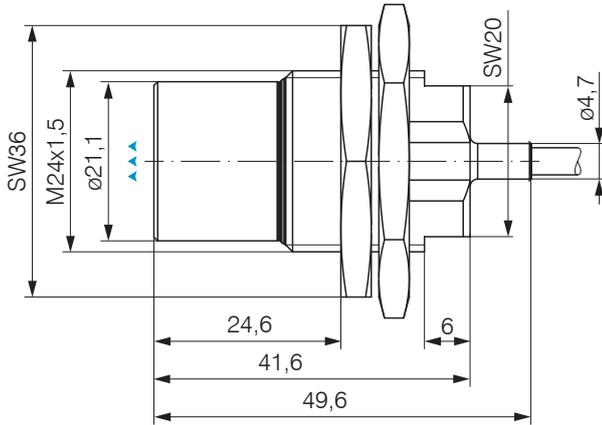


Abb. 4.20: Maßzeichnung Sensoren ES-U8-C-CAx/mB0

4.5 Sensorkabel

- ▶ Knicken Sie das Kabel nicht ab. Beachten Sie die minimalen Biegeradien.

Kabel- \varnothing 3,6 mm		
ES-U1-C-CAx/mB0	festverlegt, statisch	27 mm
ES-S1-C-CAx/mB0		dynamisch
ES-U1-T-CAx/mB0		
ES-U2-C-CAx/mB0		
ES-S2-C-CAx/mB0		
ES-U3-C-CAx/mB0		
ES-U3-T-CAx/mB0		
ES-S4-C-CAx/mB0		
ES-U6-C-CAx/mB0		
ES-U8-C-CAx/mB0		

Tab. 4.3: Minimale Biegeradien der Sensoren bzw. Sensorkabel

- ▶ Verlegen Sie das Sensorkabel so, dass keine scharfkantigen oder schweren Gegenstände auf den Kabelmantel einwirken.
- ▶ Schließen Sie das Sensorkabel an den Controller an.

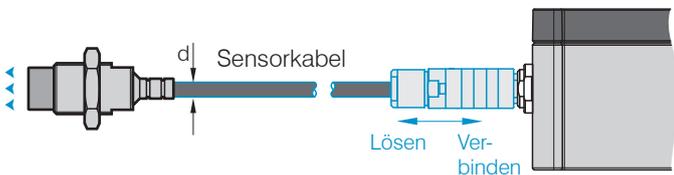


Abb. 4.21: eddyNCDT Sensorkabel

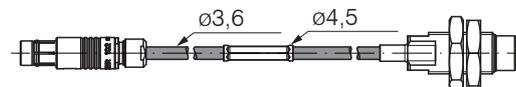
Zum Lösen der Steckverbindung werden die Steckverbinder an den gerillten Griffstücken (Außenhülsen) angefasst und gerade auseinandergezogen.

i Ein Ziehen am Kabel und der Spannmutter verriegelt die Steckverbinder und führt nicht zum Lösen der Verbindung. Vermeiden Sie deshalb übermäßigen Zug auf die Kabel. Kürzen Sie nicht das Sensorkabel. Ein gekürztes oder beschädigtes Sensorkabel kann nicht repariert werden. Ein Neu-Abgleich ist erforderlich!

- ▶ Prüfen Sie die Steckverbindungen auf festen Sitz.



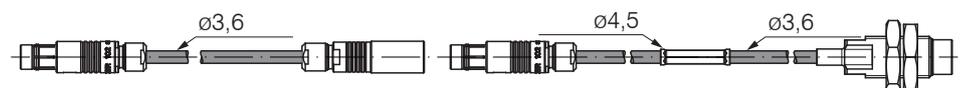
Sensoren mit integriertem Kabel: Kabeltyp ES-xx-x-CAx/mB0



Sensoren mit Buchse: Kabeltyp EC-x/mB0/mB0



Verlängerungskabel: Kabeltyp ECE-x/fB0/mB0



Tab. 4.4: Anschlusskabel für Sensoren, Reihe DT3020

Sensoren und Kabel, die am Controller DT3020 angesteckt werden sollen, tragen die Steckerbezeichnung mB0 im Namen. Siehe auch Modellbezeichnungen, [siehe Kap. 17.3](#)

4.6 Maßzeichnung Controller

Der Controller DT3020 ist in ein Aluminiumgehäuse eingebaut. Der Controller wird werkseitig auf den mitgelieferten Sensor mit Sensorkabel abgestimmt. Der Controller verfügt über zwei Durchgangsbohrungen mit $\varnothing 4,4$ mm zur sicheren Montage.

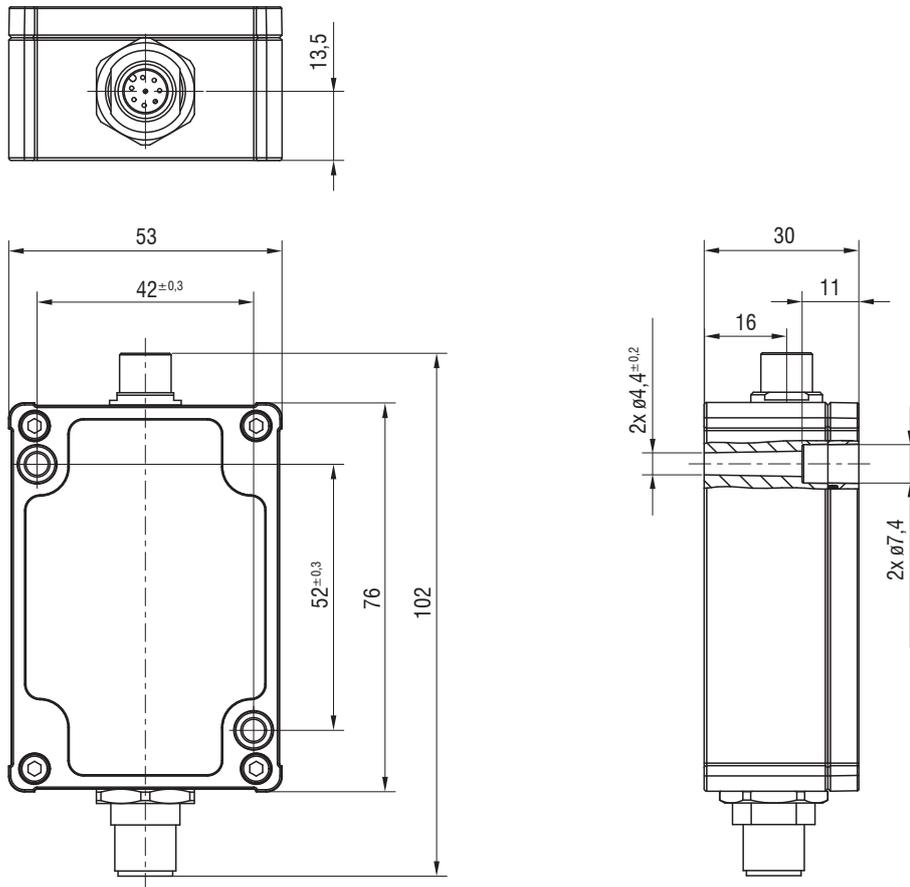


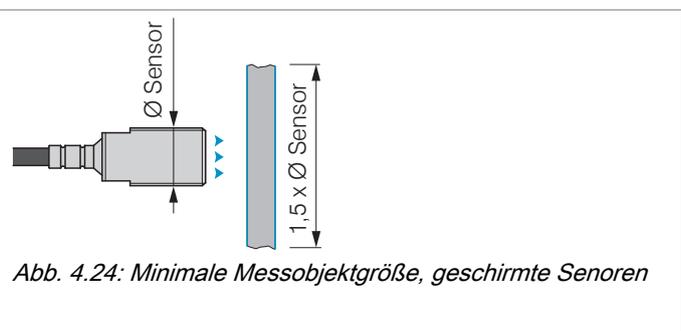
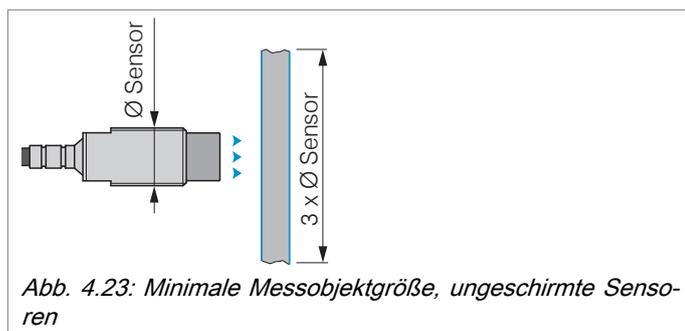
Abb. 4.22: Maßzeichnung Controller DT3020, Abmessungen in mm

4.7 Messobjektgröße

Bei Wirbelstromsensoren hat die relative Größe des Messobjekts zum Sensor Auswirkungen auf Linearitäts- und Steigungsabweichung.

Kann die geforderte Target-Mindestgröße nicht eingehalten werden, so sind für eine ausreichend hohe Linearität und Steigung folgende Aspekte zu beachten:

- Die Größe des Messobjekts darf sich nicht verändern.
- Das Target darf nicht lateral zur Sensorstirnfläche bewegt werden.

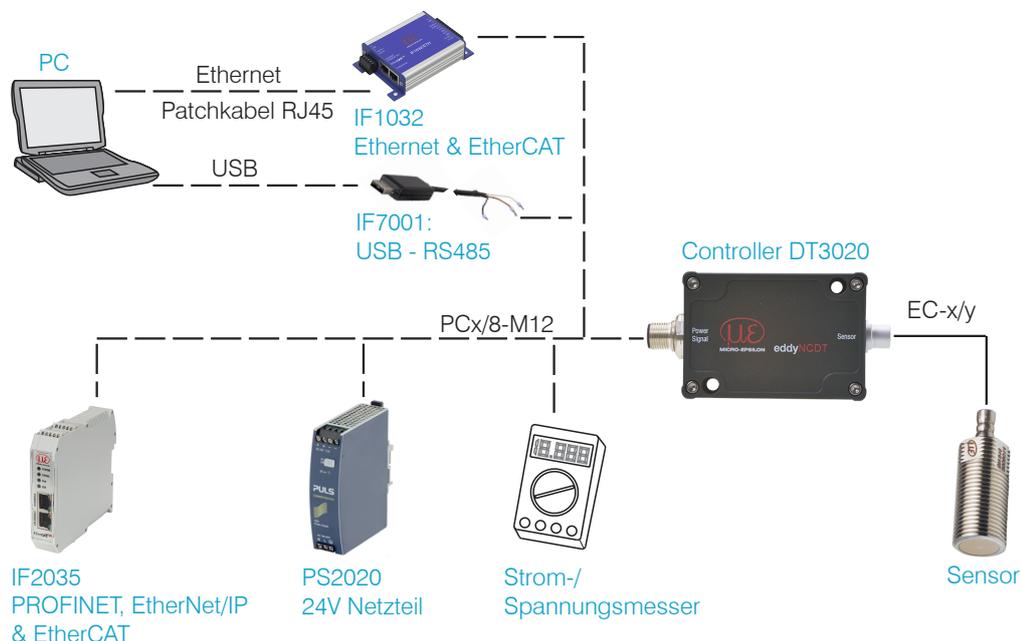


Eine erfolgreiche Kalibrierung ist Voraussetzung für möglichst kleine Linearitätsfehler. Um ein optimales Messergebnis zu erzielen, empfiehlt Micro-Epsilon eine Linearitäts-Kalibrierung auf das entsprechende Messobjekt. Eine Veränderung der Messobjektgröße beeinflusst die Güte der Messergebnisse.

4.8 Elektrische Anschlüsse

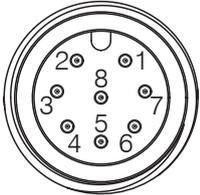
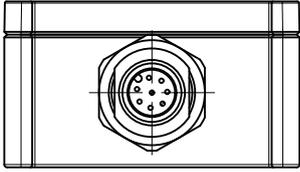
4.8.1 Anschlussmöglichkeiten

Die Spannungsversorgung und Signalausgabe erfolgen an der Vorderseite des Controllers.



Optionales Zubehör finden Sie hier, [siehe Kap. 17.1](#)

4.8.2 Anschlussbelegung

DT3020			
Pin	Beschreibung	Adernfarbe PCx/8-M12	
1	NC	Weiß	 <p>Abb. 4.25: Stiftseite 8-pol. Gehäusestecker</p>
2	12 ... 30 VDC	Braun	
3	Schaltausgang	Grün	
4	RS485 A / +	Gelb	 <p>Abb. 4.26: Versorgung und Analogausgang Controller, 8-pol. Stecker, M12 A-kodiert</p>
5	RS485 B / -	Grau	
6	GND (Weg)	Pink	
7	GND (Versorgung)	Blau	
8	Analog Ausgang	Rot	

Tab. 4.5: Anschlussbelegung und Farbcodes

Das PCx/8-M12 ist ein fertig konfektioniertes Versorgungs- und Ausgangskabel, welches in den Längen 3 m, 5 m oder 10 m optional als Zubehör mitbestellt werden kann. Die beiden Massen GND (Versorgung) und GND (Weg) sind intern miteinander verbunden. Die Ausgänge sind kurzschlussfest. Ein 5 m langes Kabel ist in einer schleppkettentauglichen Ausführung bis +105 °C erhältlich, [siehe Kap. 17.1](#)

4.8.3 Versorgungsspannung

Nennwert: 24 V DC (12 ... 32 V, Leistungsaufnahme < 1,7 W).

- Schalten Sie das Netzteil erst nach Fertigstellung der Verdrahtung ein.

- ▶ Verbinden Sie die Eingänge „2“ und „7“ am Controller mit einer 24-V-Spannungsversorgung.

Spannungsversorgung nur für Messgeräte, nicht gleichzeitig für Antriebe oder ähnliche Impulsstörquellen verwenden. MICRO-EPSILON empfiehlt die Verwendung des optional erhältlichen Netzteils PS2020 für den Controller, [siehe Kap. 17.1](#)

	Controller - Pin	PCx/8-M12 - Farbe	Versorgung
	2	Braun	V_+
	7	Blau	$GND_{\text{Versorgung}}$

Tab. 4.6: Anschluss Versorgungs-spannung

4.8.4 Analogausgang, Weg

Der Controller stellt im Auslieferungszustand einen Stromausgang 4 ... 20 mA zur Verfügung.

Der Stromausgang ist von 0 ... 20 mA skalierbar, [siehe Kap. 8.2.1](#)

- ▶ Verbinden Sie den Ausgang 8 (Rot) und 6 (Pink) am Controller mit einem Messgerät.

	Controller	
	8-pol. M12 Kabelstecker	Aderfarbe PCx/8-M12
	I_{Out} (Pin 8)	Rot
	GND_{Weg} (Pin 6)	Pink

Tab. 4.7: Beschaltung für Stromausgang

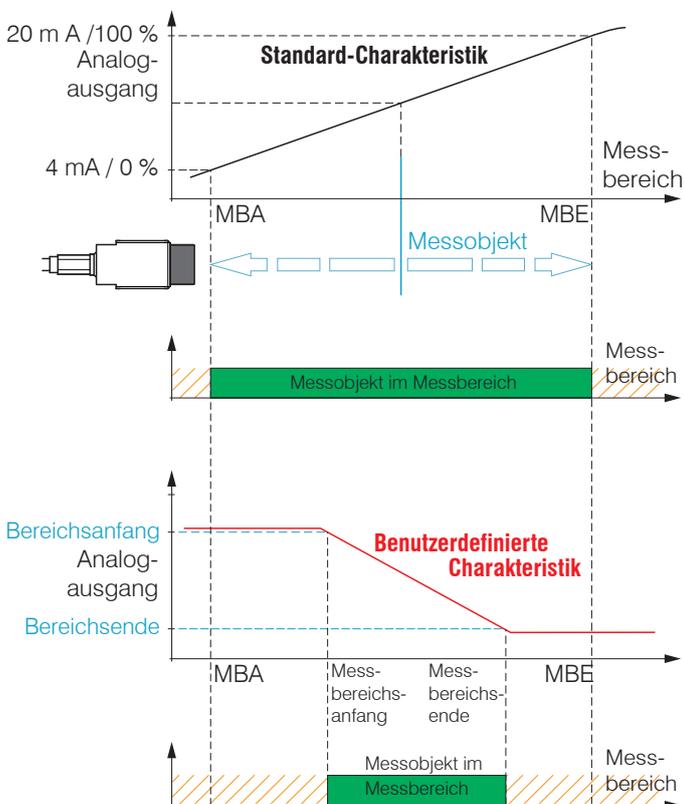


Abb. 4.27: Kennlinien Standard und benutzerdefiniert

4.8.5 Schaltausgang

Der Grenzwertausgang ist im Auslieferungszustand deaktiviert und kann z.B. über die Software `sensorTOOL` konfiguriert werden, [siehe Kap. 8.3](#)

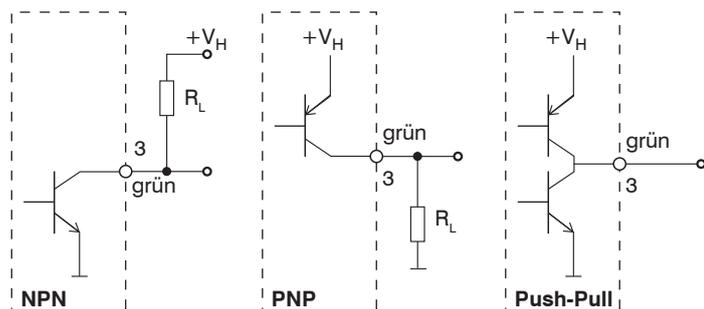


Abb. 4.28: Prinzipschaltung des Schaltausgangs

Das Schaltverhalten (NPN Öffner, NPN Schließer, PNP Öffner, PNP Schließer, Push-Pull, Push-Pull negiert), [siehe Kap. 8.3.4](#) der beiden Schaltausgänge hängt von der Einstellung des DT3020 ab. Der Schaltausgang ist geschützt gegen:

- Verpolung
- Überlastung: Das Status Bit 7, [siehe Kap. 10.5](#) ist auf 1 gesetzt. Der Schalter öffnet 14.8 ms und schließt wieder für 0.2 ms. Ist der Überlastzustand behoben, arbeitet der Schalter wieder regulär. Die Ein- und Ausschaltzyklen dienen der Erkennung, ob der Überlastzustand noch vorhanden oder behoben ist.
- Übertemperatur: Das Status Bit 7, [siehe Kap. 10.5](#) ist auf 1 gesetzt und der Schalter öffnet. Ist eine bestimmte Temperaturschwelle unterschritten, arbeitet der Schalter wieder regulär.

Die Spannungsniveaus für die Schaltzustände *high* bzw. *low* variieren je nach angeschlossener Last. Für die Spannung V_H an einem Pull-Up Widerstand muss gelten: $V_H \leq 32 \text{ V}$.

Ausgangsstrom	U_{low}	U_{high}
100 mA ^[13]	< 1,5 V	> $+V_H - 1,5 \text{ V}$
<=50 mA	< 0,5 V	> $+V_H - 0,5 \text{ V}$

Wird der Controller mit 24 V versorgt und ist der obere Schalter geschlossen. Die Ausgangsspannung des Schaltausgangs bei 50 mA Laststrom beträgt in diesem Fall mindestens $24 \text{ V} - 0,5 \text{ V} = 23,5 \text{ V}$ und bei 100 mA Laststrom mindestens $24 \text{ V} - 1,5 \text{ V} = 22,5 \text{ V}$.

- i Abschaltung bei Überlast von > 140 mA.
Das Status Bit 7 wird auf 1 gesetzt, [siehe Kap. 10.5](#), der Ausgang wechselt in den Zustand *Überlastung*.

[13] entspricht Sättigung bei I_{max}

5 Inbetriebnahme über die Software sensorTOOL

5.1 Beschreibung, Download

Mit sensorTOOL steht Ihnen eine Software zur Verfügung, mit der Sie den Controller einstellen und Messungen visualisieren sowie dokumentieren können.

Die Software sensorTOOL können Sie sich unter folgendem Link herunterladen <https://www.micro-epsilon.de/service/software-sensorintegration/sensortool/>

- ▶ Verbinden Sie den Controller DT3020 über einen freien USB-Port Ihres PCs mit einem RS485 Konverter und schließen Sie die Stromversorgung an den DT3020 an. Alternativ können Sie eine Verbindung mit dem optionalen Schnittstellenmodul IF1032 über EtherNet herstellen, [siehe Kap. 4.8.1](#)

5.2 Messsystemaufbau prüfen

- Ist der Sensor auf den Anwendungsfall (Messobjektwerkstoff) abgestimmt?
- Sind Sensor, Sensorkabellänge und Controller aufeinander abgestimmt (Typ und Seriennummer)?
- Ist der Sensor angeschlossen? Sind die Kabelverbindungen fest?

5.3 Messobjekt platzieren

- ▶ Platzieren Sie das Messobjekt innerhalb des Sensormessbereichs. Der Wert für den Messbereichsanfang (MBA) hängt vom Sensor ab, beträgt aber standardmäßig 10% des Messbereichs. Den Wert finden Sie in den technischen Daten zum Sensor. Wird der Messbereich durch den Anwender eingeschränkt, ergeben sich u. U. neue Werte für MBA, MBM und MBE.

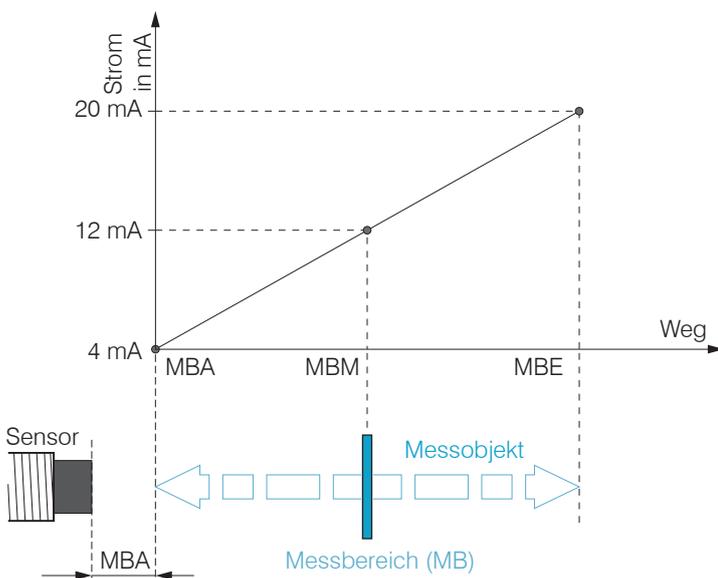


Abb. 5.1: Skalierung Analogausgang ab Werk

5.4 Singlesensor Mode

- ▶ Starten Sie das Programm `sensorTOOL`.

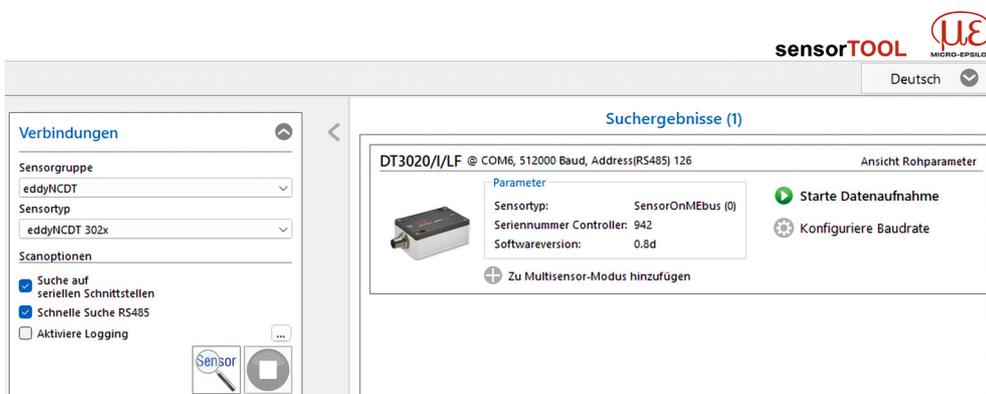


Abb. 5.2: Erste interaktive Seite nach Aufruf des Programms `sensorTOOL`

- ▶ Stellen Sie in den Dropdown-Menüs die Sensorgruppe `eddyNCDT` und den Sensortyp `eddyNCDT 3020` ein
- ▶ Wählen Sie den angeschlossenen Sensor aus.
- ▶ Setzen Sie den Haken bei `Suche auf seriellen Schnittstellen`.
- ▶ Falls nur 1 Controller am Bus betrieben wird, setzen Sie den Haken bei `Schnelle Suche RS485`.

Falls mehrere Controller am Bus betrieben werden, muss der Haken bei `Schnelle Suche RS485` deaktiviert werden. Die Suche dauert in diesem Fall länger, da das Programm den gesamten MEBus-Adressbereich scannt.

- ▶ Klicken Sie auf die Schaltfläche `Sensor` mit dem Lupensymbol, um die Suche zu starten.
- ▶ In der Übersicht `Suchergebnisse (x)` werden nun alle verfügbaren Kanäle angezeigt.
- ▶ Über die Schaltfläche `Starte Datenaufnahme` und `Konfiguriere Baudrate` können nun weitere Menüs aufgerufen werden.
- ▶ Wählen Sie den gewünschten Sensor aus den Suchergebnissen aus.

5.5 Multisensor Mode

Die Software `sensorTOOL` bietet zudem die Möglichkeit die Daten mehrerer DT302x auszugeben.

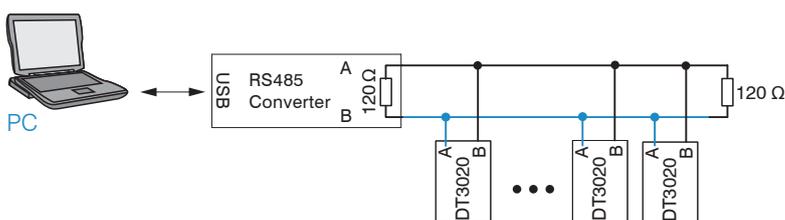


Abb. 5.3: Verbindung mehrerer DT3020 über eine RS485-Schnittstelle

Zwischen den Anschlussleitungen A und B der RS485-Schnittstelle ist am Anfang und am Ende des RS485-Busses ein Abschlusswiderstand von $120\ \Omega$ erforderlich. Im DT3020 ist kein Abschlusswiderstand der RS485-Leitung eingebaut. Daher ist der Anschluss mehrerer Controller an ein Buskabel zulässig.

Um die Daten mehrere Bus-Teilnehmer in einem Graphen auszugeben, gehen Sie bitte wie folgt vor:

Suchen Sie in `sensorTOOL` nach dem Controller, siehe *Single Sensor Mode*.

- ▶ Falls noch nicht erfolgt, konfigurieren Sie jeden einzelnen Kanal. Kehren Sie im Anschluss zur ersten interaktiven Seite nach Aufruf des `sensorTOOL` (Suchergebnisse) zurück.
- ▶ Starten Sie nun den `Multisensor-Modus`.
- ▶ Aktivieren Sie anschließend die einzelnen Checkboxes der betreffenden Kanäle.
- ▶ Mit dem grünen Play-Symbol auf der linken Seite starten Sie die Datenaufnahme.

Der Multisensor Mode funktioniert auch mit mehreren USB auf RS485 Convertern an einem PC.

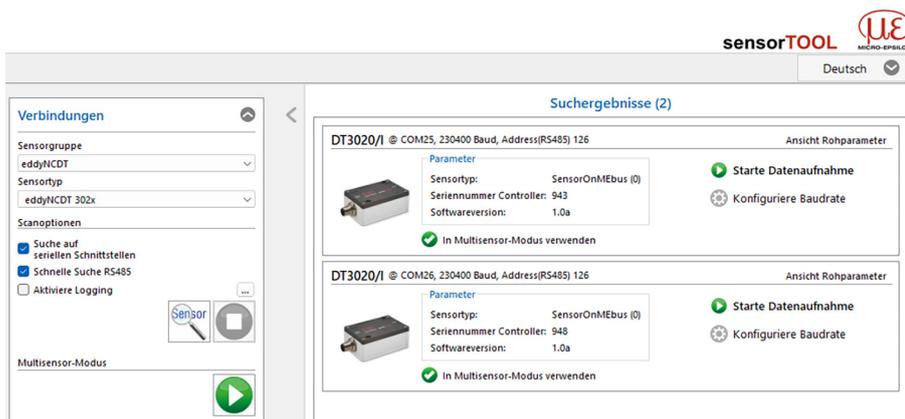


Abb. 5.4: Multisensor-Mode im sensorTOOL

5.6 Baudrate einstellen

5.6.1 Baudrate konfigurieren

- Um die aktuelle Konfiguration der seriellen Schnittstelle einzusehen und diese gegebenenfalls zu ändern, klicken Sie auf der rechten Seite die Schaltfläche `Konfiguriere Baudrate`.

Daraufhin öffnet sich das Fenster `Einstellungen serielle Schnittstelle`, siehe Abbildung. Hier kann die Baudrate geändert und eine neue Adresse für das Gerät vergeben werden.

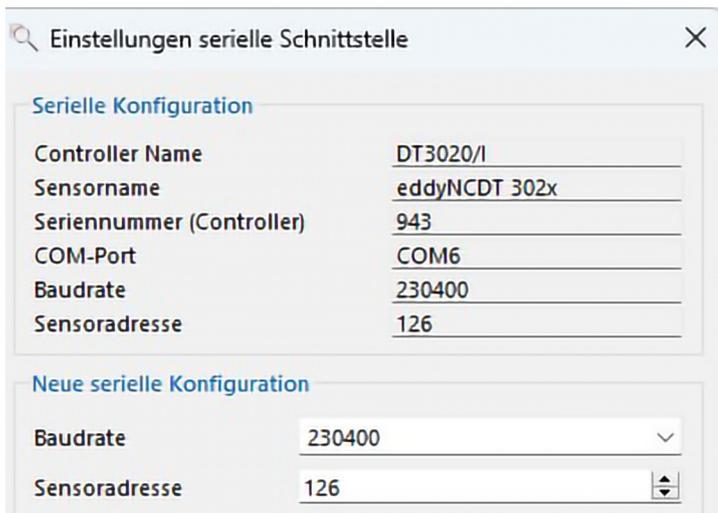


Abb. 5.5: Konfiguriere Baudrate

5.6.2 Baudrate ändern

Im Fenster `Einstellungen serielle Schnittstelle` kann die Baudrate aus einem Dropdown-Menü ausgewählt werden.

Vom DT3020 werden folgende Baudraten unterstützt, siehe Abbildung:

- 9600 Bit/s
- 230400 Bit/s
- 256000 Bit/s
- 460800 Bit/s
- 512000 Bit/s

Byte Rahmen: 1 Start Bit, 8 Daten Bits, 1 Parity Bit (parity = even), 1 Stopp Bit

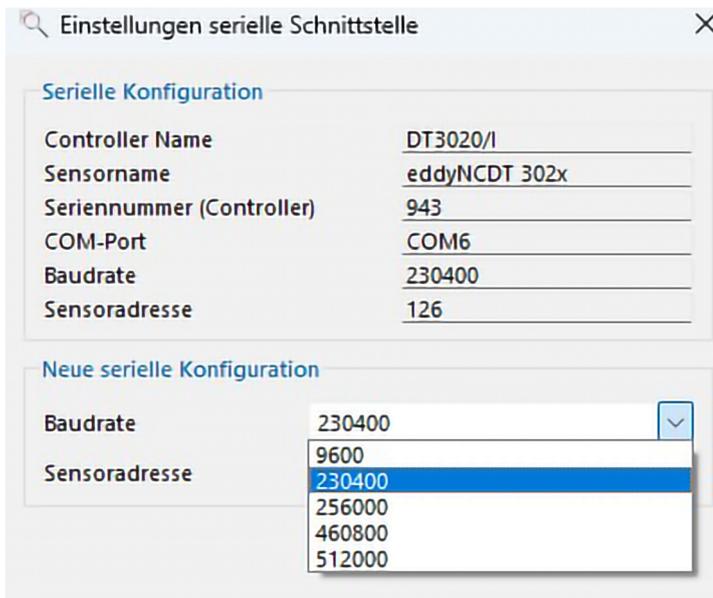


Abb. 5.6: Einstellen der Baudrate

5.7 Menüführung

Zur generellen Bedienung der Micro-Epsilon Software `sensorTOOL` ist ein Video-Tutorial verfügbar: <https://www.micro-epsilon.de/service/software-sensorintegration/sensortool/>

- Wählen Sie den Menüpunkt `Starte Datenaufnahme`.

Sie gelangen in das Hauptmenü des Programms mit der Auswahlleiste `Verbindungen`, `Datenaufnahme`, `Einzelwert`, `Einstellungen` und `Info`.



Abb. 5.7: eddyNCDT 3020 Menü sensorTOOL 120x13 TIF

`Verbindungen`

Führt Sie zurück zur Startseite des `sensorTOOL` mit der Auswahl `Starte Datenaufnahme` und `Konfiguriere Baudrate`.

`Datenaufnahme`

Start und Stopp der Datenaufnahme, Visualisierung der Messung als Signalverlauf, zeigt Elektronik- und Sensortemperatur, ermöglicht Einstellungen der Signalverarbeitung, CSV Ausgabe, InfluxDB Ausgabe, TCP/UDP Ausgabe und das Referenzzeit Setup, [siehe Kap. 6](#)

`Einzelwert`

Start und Stopp der Datenaufnahme, zeigt digitalen Messwert, Elektronik- und Sensortemperatur, ermöglicht Einstellungen der Signalverarbeitung, CSV Ausgabe, InfluxDB Ausgabe, TCP/UDP Ausgabe und das Referenzzeit Setup, [siehe Kap. 7](#)

`Einstellungen`

- `Feldlinearisierung` mit Skalierung des Messbereichs sowie Analog- und Abstandswerten und Einstellung der Feldlinearisierung, [siehe Kap. 8.2](#)
- `Schaltausgang` mit Detektor-, Grenzwerttyp- und Zeitbedingungseinstellungen, [siehe Kap. 8.3](#)
- `Filter` und `Messraten` mit Einstellungen zu Filter und Mittelung, Datenübertragung RS485 und Datenübertragungsd Diagnose, [siehe Kap. 8.5.1](#)
- `Status Bits` mit Temperatur Warnschwellen, [siehe Kap. 8.4](#)

`Info`

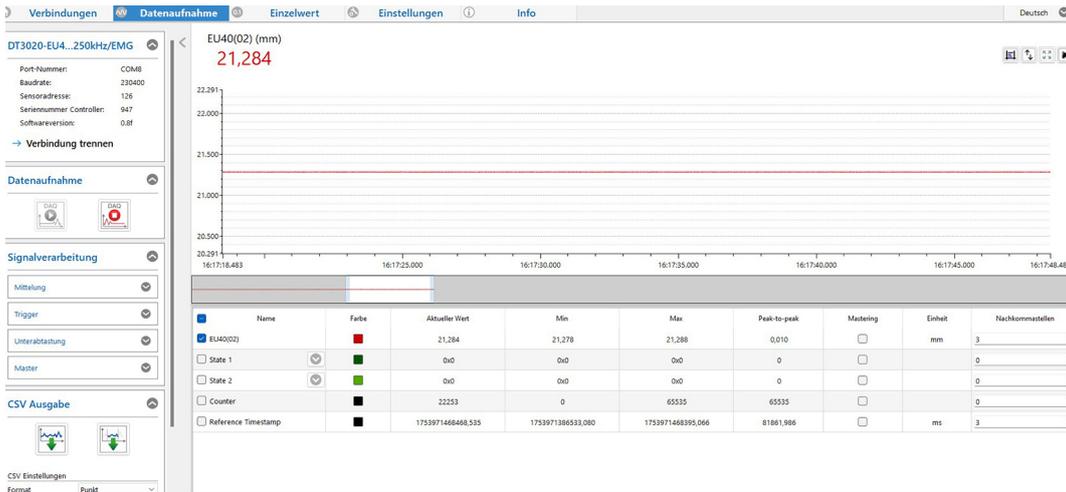
Unter dem Menüpunkt `Info` werden Ihnen Controller-, Sensor-, Diagnose- und Kennlinien-Werte angezeigt

6 Datenaufnahme im sensorTOOL

- i Einstellungen, die Sie in den nachfolgenden Punkten vornehmen, wirken sich nur auf die Verarbeitung über die Software sensorTOOL aus. Sollen Einstellungen direkt auf dem Controller wirksam werden und der Controller unter Berücksichtigung vorheriger Einstellungen die Messwerte liefern, so nutzen Sie hierfür den Menüpunkt **Einstellungen** im Hauptmenü des Programms. Mehr dazu, [siehe Kap. 8](#)

6.1 Datenaufnahme

Klicken Sie auf die Schaltfläche **Starte Datenaufnahme** oder auf das **Controllersymbol**, um weiter Einstellungen vorzunehmen und die Datenaufnahme zu starten.



- Wählen Sie vor der ersten Datenaufnahme die gewünschten Parameter über den Menüpunkt **Einstellungen**. Das Auswahlmenü in der rechten oberen Ecke ermöglicht folgende Einstellungen:

Symbol



Beschreibung

Aktiviert eine zweite Y-Achse, sobald sowohl Distanzkanäle als auch andere Signale angezeigt werden. Nicht-Distanzkanäle werden in ein andere Koordinatensystem verschoben, um eine optimale Autoskalierung zu gewährleisten.



Invertiert die Skalierung der Y-Achse.



Stellt die Y-Skala auf die ursprüngliche Einstellung zurück (z.B. nach Zoom).



Der aktuelle Signalverlauf wird angezeigt.

Einheiten ändern

Das Auswahlmenü in der rechten unteren Ecke ändert die Einheit für Abstandswerte.

Einheit	Nachkommastellen
mm	3
°C	1
°C	3
	0
	0
ms	3

Datenaufnahme starten und stoppen

Datenaufnahme



Die Aufnahme wird neu gestartet, wenn Sie diese Schaltfläche betätigen..
Die vorher angehaltene Aufnahme geht verloren.



Die Aufnahme wird gestoppt, wenn Sie diese Schaltfläche betätigen..

Tab. 6.1:

Verbindung trennen

Bei Drücken der Schaltfläche `Verbindung trennen` springt das Menü zur Einstiegsseite des `sensorTOOL` zurück.

6.2 Signalverarbeitung

Einstellung für die Signalverarbeitung im `sensorTOOL`. Diese Einstellungen beeinflussen nicht die Einstellungen im Controller DT3020 und ändern auch nicht die Verarbeitung der Messwerte im Controller.

Folgende Auswahlmöglichkeiten bei der Signalverarbeitung stehen zur Verfügung:

- Mittelung
- Trigger
- Unterabtastung: Messwertbasierend
- Master

6.2.1 Gleitender Mittelwert

Gleitender Mittelwert

Über die wählbare Filterbreite N aufeinander folgender Messwerte wird der arithmetische Mittelwert M_{gl} gebildet und ausgegeben. Jeder neue Messwert wird hinzugenommen, der erste (älteste) Messwert aus der Mittelung (aus dem Fenster) wieder herausgenommen.

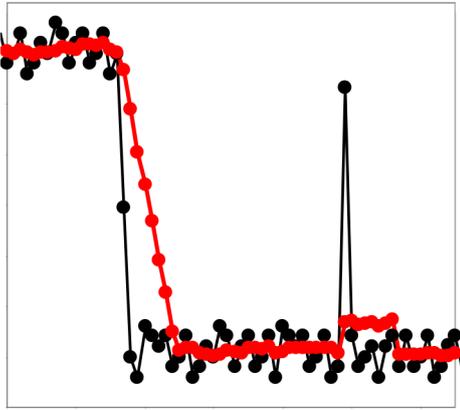
$M_{gl} = \frac{\sum_{k=1}^N MW(k)}{N}$	MW = Messwert
	N = Mittelungszahl
	k = Laufindex (im Fenster)
	M_{gl} = Mittelwert bzw. Ausgabewert

Dadurch werden kurze Einschwingzeiten bei Messwertsprüngen erzielt.

Beispiel: N=4

... 0, 1, <u>2, 2, 1, 3</u>	... 1, 2, <u>2, 1, 3, 4</u>	Messwert
↓	↓	
$\frac{2, 2, 1, 3}{4} = M_{gl}(n)$	$\frac{2, 1, 3, 4}{4} = M_{gl}(n+1)$	Ausgabewert

Abb. 6.1: Gleitender Mittelwert Formel 2 Beispiele 116x16



Anwendungshinweise

- Glätten von Messwerten
 - Die Wirkung kann fein dosiert werden im Vergleich zur rekursiven Mittelung.
 - Bei gleichmäßigem Rauschen der Messwerte ohne Spikes.
 - Auch für Messwertsprünge geeignet bei relativ kurzen Einschwingzeiten.
- Signal ohne Mittelung
— Signal mit Mittelung

Tab. 6.2:

6.2.2 Gleitender Median

Aus einer vorgewählten Anzahl von Messwerten wird der Median gebildet.

Bei der Bildung des Medians werden die einlaufenden Messwerte nach jeder Messung neu sortiert.

Der Median teilt eine Liste von Werten in zwei Hälften. Er kann auf folgende Weise bestimmt werden:

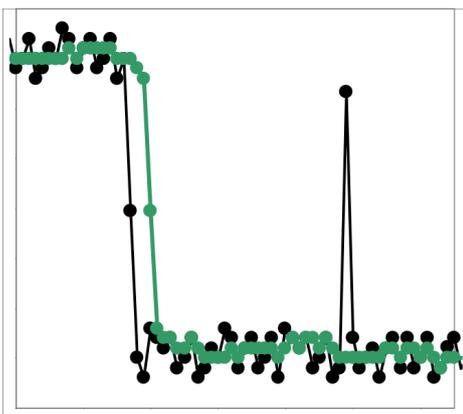
- Alle Werte werden (aufsteigend) geordnet.
- Wenn die Anzahl der Werte ungerade ist, ist die mittlere Zahl der Median.
- Wenn die Anzahl der Werte gerade ist, wird der Median als arithmetisches Mittel der beiden mittleren Zahlen berechnet.

Damit lassen sich einzelne Störimpulse unterdrücken. Die Glättung der Messwertkurven ist jedoch nicht sehr stark.

Beispiel: Median aus fünf Messwerten

... 0 1 2 4 5 1 3 → Messwerte sortiert: 1 2 **3** 4 5 Median_(n) = 3

... 1 2 4 5 1 3 5 → Messwerte sortiert: 1 3 **4** 5 5 Median_(n+1) = 4



Anwendungshinweise

- Glättung der Messwertkurve nicht sehr stark, eliminiert vor allem Ausreißer
 - Unterdrückt einzelne Störimpulse
 - Bei kurzen starken Signalpeaks (Spikes)
 - Auch bei Kantensprüngen geeignet (nur geringer Einfluss)
 - Zusätzliche Mittelung kann nach dem Medianfilter verwendet werden
- Signal ohne Mittelung
— Signal mit Mittelung

Tab. 6.3: Median, N = 7

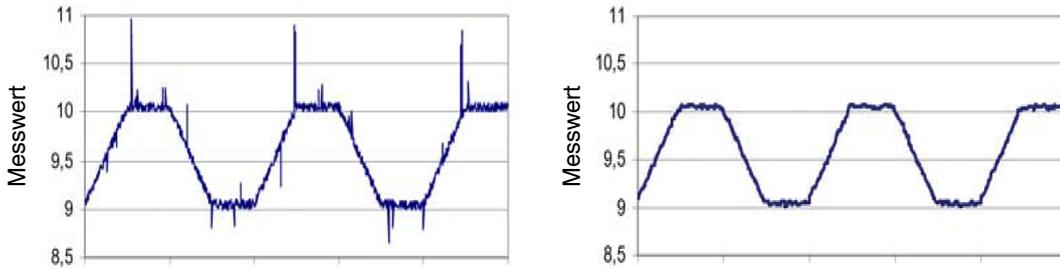


Abb. 6.2: Signalverlauf Profil ohne Median (links), mit Median N = 9 (rechts)

6.2.3 Rekursiver Mittelwert

Jeder neue Messwert $MW(n)$ wird gewichtet zum $(n-1)$ -fachen des vorherigen Mittelwertes hinzugefügt.

Formel:

$$M_{rek}(n) = \frac{MW_{(n)} + (N-1) \times M_{rek(n-1)}}{N}$$

N = Mittelungszahl

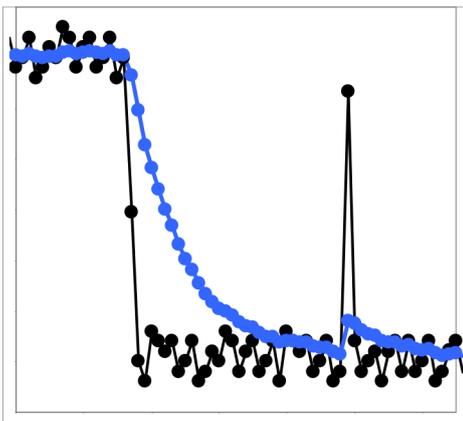
n = Messwertindex

MW = Messwert

M_{rek} = Mittelwert bzw. Ausgabewert

Tab. 6.4:

Die rekursive Mittelung erlaubt eine sehr starke Glättung der Messwerte, braucht aber sehr lange Einschwingzeiten bei Messwertsprüngen. Der rekursive Mittelwert zeigt Tiefpassverhalten.



Anwendungshinweise

- Erlaubt eine sehr starke Glättung der Messwerte. Lange Einschwingzeiten bei Messwertsprüngen (Tiefpassverhalten).
- Starke Glättung von Rauschen ohne große Spikes.
- Für statische Messungen, um das Signalrauschen besonders stark zu glätten.
- Zur Eliminierung von Strukturen, z. B. Teile mit gleichmäßigen Rillenstrukturen, gerändelte Drehteile oder grob gefräste Teile.
- Ungeeignet bei hochdynamischen Messungen.

— Signal ohne Mittelung

— Signal mit Mittelung

Tab. 6.5:

6.2.4 Trigger, Unterabtastung, Master

Erfordern eine Auswahl		Erfordern die Angabe eines Wertes
Trigger	Deaktiviert	Deaktiviert; Grundeinstellung
	Kontinuierlich	Manuelle Trigger: durch Drücken auf den Button <code>Trigger</code> wird ein Messwert aufgenommen.
	Einmalig (messwertbasierend)	Sample einstellbar; zeichnet durch Drücken des Buttons <code>Trigger</code> jeweils die eingestellte Anzahl an Samples auf; je mehr Samples, desto länger ist der Verlauf
	Einmalig (zeitbasierend)	Millisekunden einstellbar; zeichnet durch Drücken des Buttons <code>Trigger</code> die eingestellte Zeitdauer Messwerte auf.

Unterabtastung	Deaktiviert	Deaktiviert; Grundeinstellung: Jeder Messwert wird aufgezeichnet.
	Messwertbasierend	Anzahl der Samples ist einstellbar; jede x-te Messung wird erfasst.
	Zeitbasierend	Zeitbasiert: Im Abstand des eingestellten Zeitintervall wird ein Messwert aufgezeichnet.
Master	Jetzt mastern	Setzt den Master, s. Abbildung.
	Zurücksetzen	Setzt den Master wieder zurück.

Tab. 6.6: Auswahlmöglichkeiten Signalverarbeitung

Master: Ergänzung

Über die Mastering-Funktion wird der jeweilige aktuelle Messwert auf einen Vorgabewert/Masterwert gesetzt. Alle Änderungen des Wertes werden relativ zu diesem Vorgabewert angezeigt.

In der Software `sensorTOOL` ist unter dem Messsignal eine Liste, in welcher der oder die verbundenen Controller angezeigt werden.

Name	Farbe	Aktueller Wert	Min	Max	Peak-to-peak	Mastering	Einheit	Nachkommastellen
CU40002		21,249	21,244	21,255	0,011	<input type="checkbox"/>	mm	3

Hier muss zunächst unter `Mastering` das Feld aktiviert und der gewünschte Masterwert eingegeben werden.



Im Anschluss können Sie links im Menüband `Signalverarbeitung` auf den Button `Jetzt mastern` klicken.

Die Aktion war erfolgreich, wenn `Master: Enabled` angezeigt wird.



Mit dem Button `Zurücksetzen` wird der hinterlegte Masterwert entfernt.

6.3 Messung aufzeichnen und speichern

In der Datenaufnahme werden ausschließlich die Messdaten ausgegeben, aber nicht automatisch gespeichert. Sie können im seitlichen Menü unter `CSV Ausgabe` die Datenübertragung in eine `*.CSV` Datei starten oder nur den aktuellen sichtbaren Bereich aus dem Zeitdiagramm speichern. In beiden Fällen werden alle vom `sensorTOOL` erhaltenen Datenpunkte des gewählten Intervalls in die angegebene Datei geschrieben.

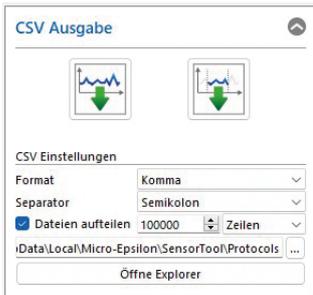


Abb. 6.3: Ansicht sensorTOOL CSV Ausgabe



Die Messdatenaufzeichnung wird in einer *CSV Datei gestartet, wenn Sie diese Schaltfläche betätigen. Um die Aufzeichnung zu beenden und abzuspeichern, klicken Sie erneut auf das Symbol.



Der aktuell sichtbare Bereich im Diagramm wird gespeichert, wenn Sie diese Schaltfläche betätigen.

Tab. 6.7: Messung aufzeichnen und speichern

Unter CSV-Einstellungen können Sie weitere Einstellungen vornehmen:

CSV Ausgabe	CSV Einstellungen	<i>Format</i>	<i>Punkt / Komma</i>	
		<i>Separator</i>	<i>Komma / Semikolon / Tabulator</i>	
		<i>Dateien aufteilen</i>	<i>Wert</i>	<i>Zeilen / MB / Minuten / Stündlich / Zeitpunkt / DAQ-Start</i>

Unter `Öffne Explorer` öffnet sich der vorher ausgewählte Pfad im Explorer, indem man die aufgezeichneten Messergebnisse einsehen kann.

7 Einzelwert im sensorTOOL

Hier werden die jeweiligen Werte einzeln ausgegeben.

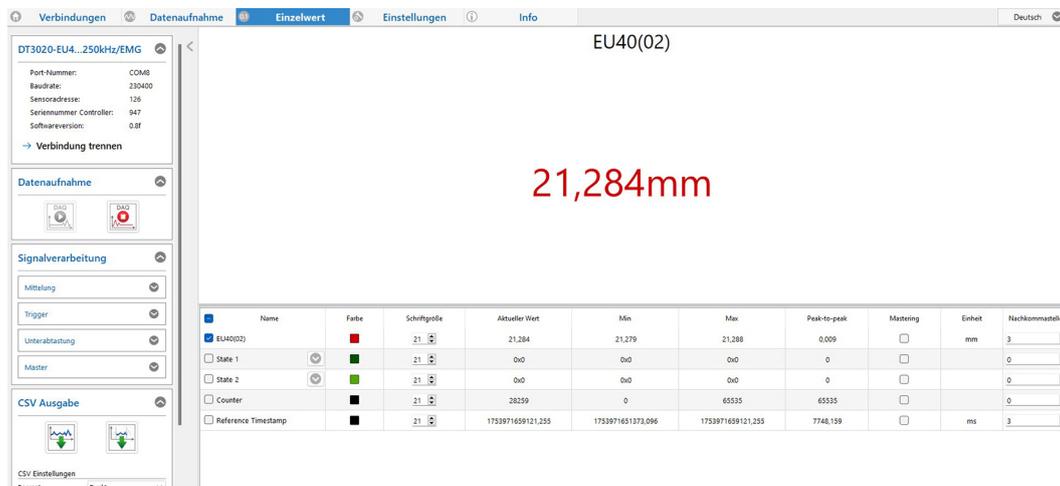


Abb. 7.1: Einzelwert sensorTOOL

Sie haben folgende Einstellmöglichkeiten:

- Farbe** Einstellung der Farbe, in der der jeweilige Wert angezeigt werden soll.
- Schriftgröße** Einstellung der Schriftgröße für den jeweiligen Wert.
- Nachkommastellen** Anzahl der angezeigten und auch in der CSV-Datei gespeicherten Nachkommastellen des Messwerts
- Mastering** Vorgabewert für die Masterfunktion unter Signalverarbeitung.

Signalverarbeitung, Messung aufzeichnen und speichern, [siehe Kap. 6.2](#) sowie, [siehe Kap. 6.3](#)

Durch einen Klick auf das Pfeilsymbol neben dem Signal *State 1* können die einzelnen Stati angezeigt werden, [siehe Kap. 10.5](#)

8 Einstellungen am Controller

- i Die Einstellungen in diesem Bereich werden direkt auf dem Controller gespeichert. Dieser liefert Messwerte unter Berücksichtigung der hier getroffenen Einstellungen. Diese wirken sich somit nicht nur auf die Datenverarbeitung per Software, sondern auch auf die Verarbeitung über Schnittstellen und den Analogausgang aus.

Einstellungen unter `Datenaufnahme` und `Einzelwert` wirken sich dagegen lediglich auf die Verarbeitung per Software aus. Sie werden nicht auf dem Controller gespeichert.

8.1 Allgemeine Informationen zu den Controller-Einstellungen

Abb. 8.1: sensorTOOL Einstellungen

Hier können Sie Einstellungen zu `Feldlinearisierung`, `Schaltausgang`, `Status Bits` sowie `Filter` und `Messraten` vornehmen.

8.2 Feldlinearisierung

8.2.1 Skalierung des Messbereichs

Sie können den Messbereich des eddyNCDT 3020 mit den Feldern `Messbereichsbeginn`, `Messbereichsende` und `Messbereich` und/oder über den Analogen Stromausgang sowie die manuelle Skalierung des Strombereichs zwischen 0 und 20 mA einstellen.

Skalierung des Messbereichs

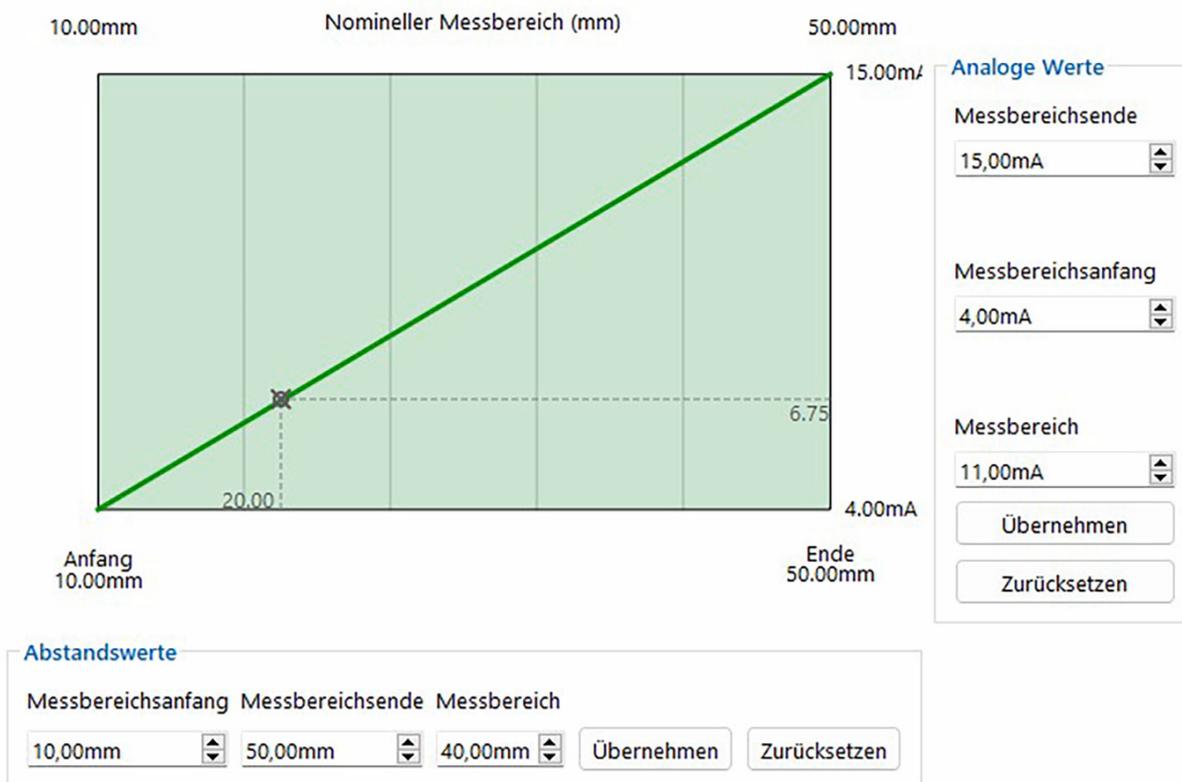


Abb. 8.2: sensorTOOL Skalierung des Messbereichs

Dadurch stellen Sie den Sensor optimal auf die zu messenden Werte ein. Ein angepasster Messbereich bewirkt eine höhere Auflösung. Dabei werden folgende Parameter verwendet:

- **Messbereichsanfang (MBA):** Untere Grenze des darzustellenden Messwertbereichs
- **Messbereichsende (MBE):** Obere Grenze des darzustellenden Messwertbereichs
- **Messbereich (MB):** Tatsächlicher Messbereich, welcher zwischen Messbereichsanfang und Messbereichsende liegt.

Wird der MBA verändert, bleibt das MBE gleich, der MB verschiebt sich entsprechend. Wird MBE verändert, bleibt der MBA gleich, der MB verschiebt sich entsprechend. Wird der MB geändert, bleibt der MBA fest, das MBE wird neu berechnet. Messbereich = Differenz aus Messbereichsende und Messbereichsanfang.

Haben Sie die Werte nach Ihren Anforderungen geändert, bestätigen Sie mit **Übernehmen**. Möchten Sie die Änderung rückgängig machen, drücken Sie **Zurücksetzen**.

- i Ist die Skalierung des Bereichs kleiner als der tatsächlich verwendete Bereich, kann dies dazu führen, dass Messwerte außerhalb des definierten Bereichs nicht dargestellt oder falsch interpretiert werden.

8.2.2 Feldlinearisierung

Jeder eddyNCDT Sensor wird im Werk unter standardisierten Einbaubedingungen (Targetmaterial, Messbereich) kalibriert. Zu den Einbaubedingungen zählen die Befestigung, Position der Mutter und umgebende Werkstoffe. Abweichungen von der Standard-Einbausituation können zu Einbußen in Bezug auf die Linearität führen. Durch Feldlinearisierungen oder werksseitige Sonderabstimmungen kann dem entgegengewirkt werden.

- i Lassen Sie das Messsystem vor einer Linearisierung circa 30 bis 60 Minuten warmlaufen.

Feldlinearisierung

Status Feldlinearisierung: Aktueller Abstandswert

Gespeicherte Feldlinearisierung: **21,241 mm**

Punkt 1 (mm)

Abb. 8.3: sensorTOOL Feldlinearisierung

- i Ist unter Status Feldlinearisierung Inaktiv ausgewählt, so misst der Sensor nach den ab Werk vorgegebenen Werten. Wählen Sie Aktiv aus, so misst der Sensor anhand der zuletzt korrekt durchgeführten Feldlinearisierung.

Funktion	Korrektur	Einsatzbereich
1-Punkt	Nullpunktverschiebung	Schnelle Nullpunktkorrektur
2-Punkt	Nullpunktverschiebung + Steigung	Lineare Abweichungen
3-Punkt	Nullpunktverschiebung + Steigung + Linearität	Nichtlineares Sensorverhalten

Tab. 8.1: Linearisierungsfunktionen

Stellen Sie die jeweiligen Abstände zwischen Sensor und Target her. Geben Sie unter Punkt 1/2/3 den jeweils physisch hergestellten Abstandswert ein. Legen Sie im Idealfall:

Feldlinearisierung	Punkt 1	Punkt 2	Punkt 3
1-Punkt Feldlinearisierung	Nahe an den Messbereichsanfang	-	-
2-Punkt Feldlinearisierung	Nahe an den Messbereichsanfang	Nahe an das Messbereichsende	-
3-Punkt Feldlinearisierung	Nahe an den Messbereichsanfang	Nahe an die Messbereichsmitte	Nahe an das Messbereichsende

8.2.2.1 1-Punkt (Nullpunktverschiebung)

Feldlinearisierung

Status Feldlinearisierung: Aktueller Abstandswert

Gespeicherte Feldlinearisierung: **9,997 mm**

Punkt 1 (mm)

Erfolgreich!

Abb. 8.4: sensorTOOL 1-Punkt (Nullpunktverschiebung)

Das System ist werksseitig linearisiert, der mechanische Nullpunkt im eingebauten Zustand soll neu definiert werden.

Ein gemessener Sensorwert wird mit einem bekannten Referenzabstand (dieser kann frei innerhalb des Sensormessbereichs definiert werden) abgeglichen. Die Differenz wird als fester Korrekturwert (Nullpunktverschiebung) gespeichert und für alle Messwerte verrechnet.

Beispiel: Der Sensor zeigt statt 2,00 mm Abstand 2,10 mm an → ein Offset von 0,10 mm wird abgezogen.

Diese Funktion wird angewendet, wenn die Sensorcharakteristik bereits linear ist und nur eine Verschiebung z.B. durch Montageeinflüsse vorliegt.

So gehen Sie vor

- ▶ Wählen Sie Editieren. Die Felder können nun bearbeitet werden.
- ▶ Stellen Sie 1-Punkt ein.
- ▶ Geben Sie den korrekten Referenzwert bei Punkt 1 (mm) ein.

- Klicken Sie auf `Wert setzen` und im Anschluss auf `Berechnen` und `Aktivieren`.

Anhand der grünen Haken erkennen Sie, ob die Aktion erfolgreich war.

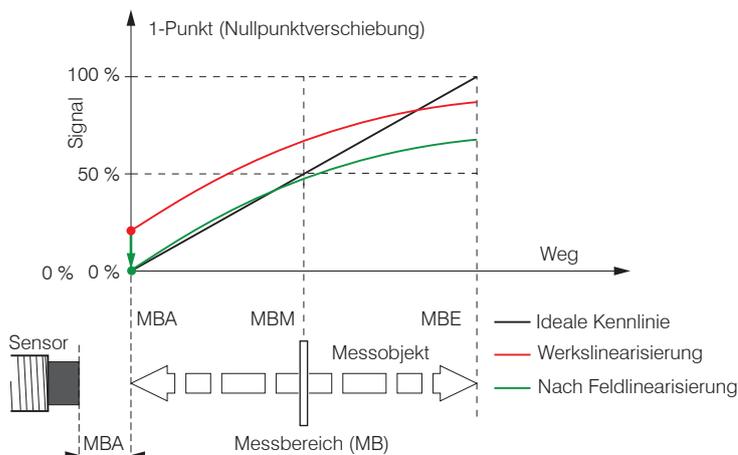


Abb. 8.5: 1-Punkt (Nullpunktverschiebung) Skizze

i

Über dieses Symbol  werden lediglich die voreingestellten Standardwerte in der Anzeige wiederhergestellt. Die auf dem Sensor hinterlegten Werte einer vorherigen Feldlinearisierung werden nicht überschrieben. Um diese Werte neu zu setzen, müssen Sie eine neue Feldlinearisierung durchführen.

8.2.2.2 2-Punkt (Nullpunktverschiebung, Empfindlichkeit)

Feldlinearisierung

Status Feldlinearisierung:	Aktiv	Aktueller Abstandswert	<input type="text" value="45,007 mm"/>
Gespeicherte Feldlinearisierung:	2-Punkt (Offset, Steigung)		
Punkt 1 (mm)	<input type="text" value="20,000"/>	Wert Setzen	✓
Punkt 2 (mm)	<input type="text" value="45,000"/>	Wert Setzen	✓
<input type="button" value="Editieren"/>		<input type="button" value="Berechnen und Aktivieren"/>	✓ Erfolgreich!

Abb. 8.6: sensorTOOL 2-Punkt (Nullpunktverschiebung, Empfindlichkeit)

Das System ist linearisiert und soll dem Messumfeld in der Maschine angepasst werden.

Zwei gemessene Punkte werden mit bekannten Referenzabständen verglichen. Daraus berechnet die Software eine lineare Funktion (Gerade), die Nullpunktverschiebung und Empfindlichkeit anpasst.

Diese Funktion bewirkt eine deutlich präzisere Messung über den gesamten Bereich, solange das Sensorverhalten linear bleibt.

So gehen Sie vor

- Wählen Sie `Editieren`. Die Felder können nun bearbeitet werden.
- Stellen Sie `2-Punkt` ein.
- Geben Sie die korrekten Referenzwerte bei `Punkt 1 (mm)` und `Punkt 2 (mm)` ein.
- Klicken Sie auf `Wert setzen` und im Anschluss auf `Berechnen` und `Aktivieren`.

Anhand der grünen Haken erkennen Sie, ob die Aktion erfolgreich war.

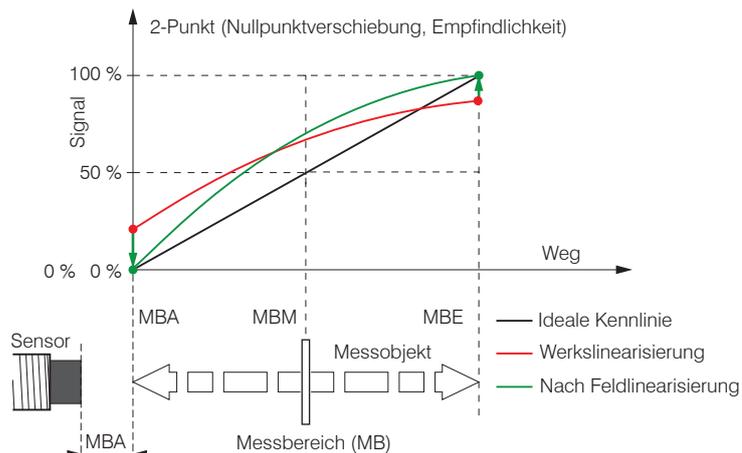


Abb. 8.7: 2-Punkt (Nullpunktverschiebung, Empfindlichkeit) Skizze

8.2.2.3 3-Punkt (Nullpunktverschiebung, Empfindlichkeit, Linearität)

Feldlinearisierung

Status Feldlinearisierung: **Aktiv** Aktueller Abstandswert
44,997 mm

Gespeicherte Feldlinearisierung: 3-Punkt (Offset, Steigung, Linearität)

Punkt 1 (mm)	20,000	Wert Setzen	✓
Punkt 2 (mm)	30,000	Wert Setzen	✓
Punkt 3 (mm)	45,000	Wert Setzen	✓

✓ Erfolgreich!

Abb. 8.8: sensorTOOL 3-Punkt (Nullpunktverschiebung, Empfindlichkeit, Linearität)

Werden vom Anwender der Sensor oder die Messobjektgeometrie gewechselt oder ändern sich z.B. die Materialeigenschaften, so ist vor der Messung eine Feldlinearisierung durchzuführen. Verwenden Sie dabei nach Möglichkeit:

- die originale Sensormontage, die in der Messung verwendet werden soll.
- das originale Messobjekt, das in der Messung verwendet werden soll.

Diese Funktion erreicht die höchste Genauigkeit über den gesamten Messbereich – auch bei komplexen Zielmaterialien oder ungünstiger Einbaulage.

So gehen Sie vor

- ▶ Wählen Sie **Editieren**. Die Felder können nun bearbeitet werden.
- ▶ Stellen Sie die **3-Punkt** ein.
- ▶ Geben Sie die korrekten Referenzwerte bei **Punkt 1 (mm)**, **Punkt 2 (mm)** und **Punkt 3 (mm)** ein.
- ▶ Klicken Sie auf **Wert setzen** und im Anschluss auf **Berechnen und Aktivieren**.

Anhand der grünen Haken erkennen Sie, ob die Aktion erfolgreich war.

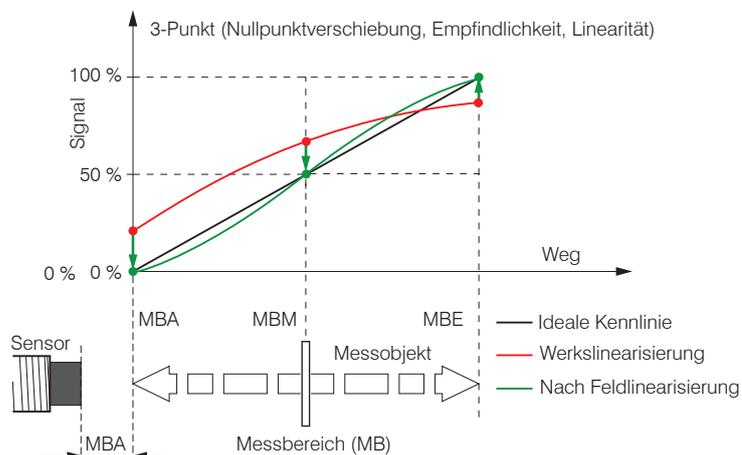


Abb. 8.9: 3-Punkt (Nullpunktverschiebung, Empfindlichkeit, Linearität) Skizze

8.3 Schaltausgang: Grenzwertüberwachung

Durch die Konfiguration der Grenzwertüberwachung, kann abhängig vom gemessenen Abstand der Schaltausgang aktiviert oder deaktiviert werden.

Die Reihenfolge der Signalverarbeitung ist in der folgenden Abbildung dargestellt:

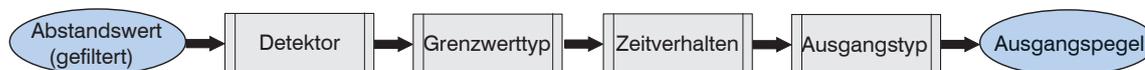


Abb. 8.10: Signalverarbeitung der Grenzwertüberwachung

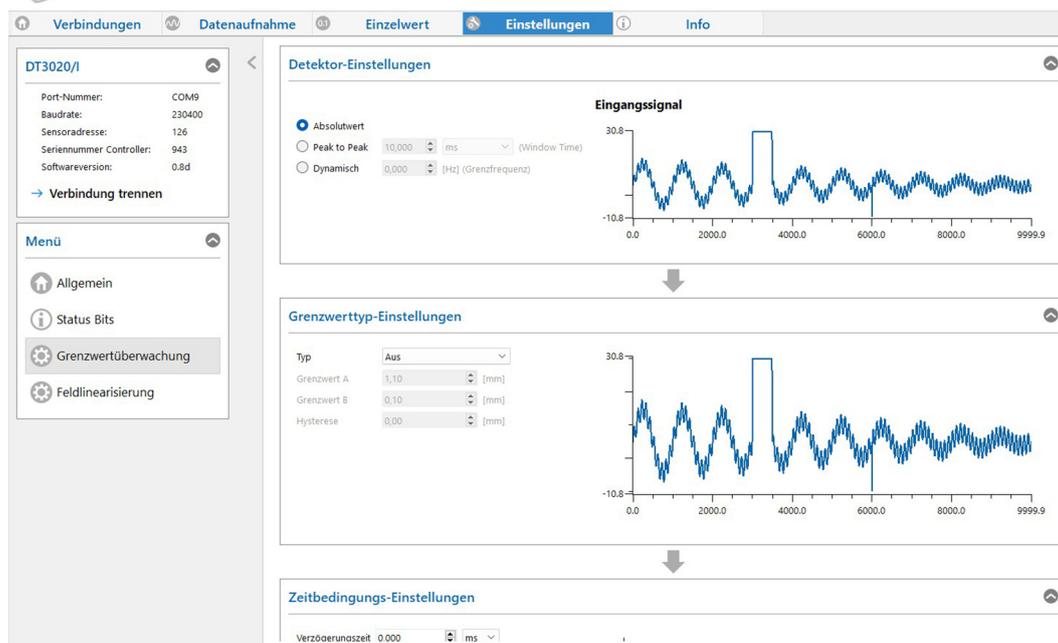


Abb. 8.11: sensorTOOL Grenzwertüberwachung

8.3.1 Detektor-Einstellungen

8.3.1.1 Absolutwert

Bei Absolutwert findet keine Veränderung des Abstandswertes im Detektor statt.

Der gemessene Abstandswert am Detektoreingang entspricht dem Abstandswert am Ausgang.

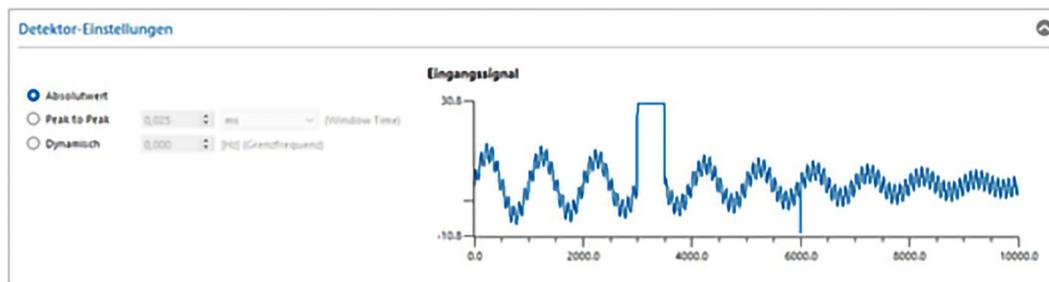


Abb. 8.12: sensorTOOL Detektor Absolutwert

8.3.1.2 Peak to peak

Bei Auswahl von Peak to Peak wird im Detektor die Differenz aus maximalem Abstandswert und minimalem Abstandswert gebildet, welche innerhalb der Fensterzeit t_{pkpk} aufgenommen wurden. Es wird daher nur in Zeitabständen von Fensterzeit t_{pkpk} ein neuer Wert gebildet. Das Signal liegt dadurch erst nach einer Zeitspanne von $1 \times t_{pkpk}$ am Ausgang an.



Abb. 8.13: Beispiel für den Peak-Peak Detektor

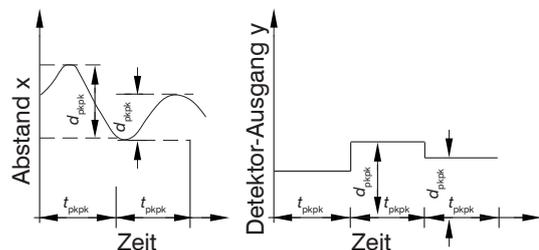


Abb. 8.14:

8.3.1.3 Dynamisch

Der Detektor „Dynamisch“ ist ein Hochpass 1. Ordnung, dessen -3 dB Grenzfrequenz eingegeben wird.

Die nachfolgende Abbildung zeigt ein Beispiel, wie sich der Filter des Detektors auswirkt. Das Eingangssignal besteht aus zwei Sinussignalen mit 0,1 Hz (Periodendauer 10 s) und 4 Hz (Periodendauer 0,25 s) sowie einem Gleichanteil. Für den Detektor ist eine Grenzfrequenz von 1 Hz eingestellt. Der Gleichanteil wird vollständig unterdrückt. Das niederfrequente Sinussignal ist nur wenig am Ausgang zu sehen, während das hochfrequente Sinussignal fast unverändert den Filter passiert.



Abb. 8.15: Beispiel für den Detektor Dynamisch ($f_g \approx 1$ Hz)

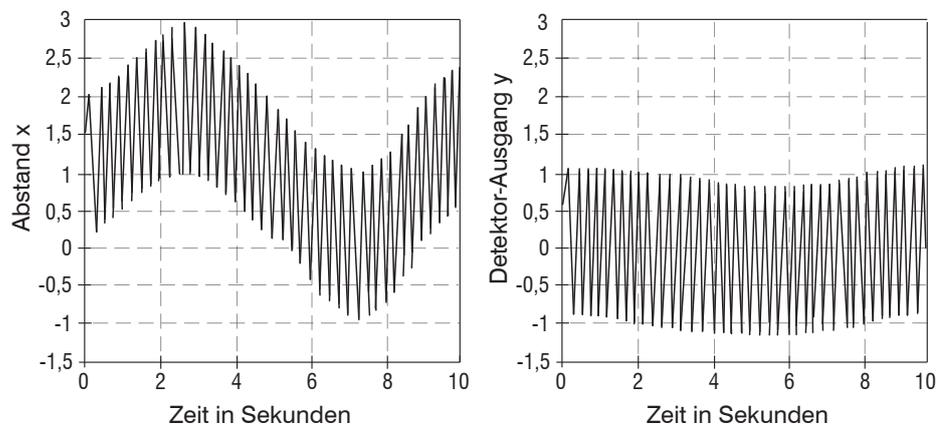


Abb. 8.16: Beispiel für Frequenzunterdrückung mit Hilfe der Detektor-Einstellung: Dynamisch ($f_g = 1$ Hz)

Im Folgenden ist ein 0,1 Hz Sinussignal mit einem kurzen Puls von 0,1 s überlagert. Die Grenzfrequenz ist wieder auf 1 Hz eingestellt. Das langsamere Sinussignal wird unterdrückt. Der kurze Puls am Ausgang tritt dadurch besser hervor.

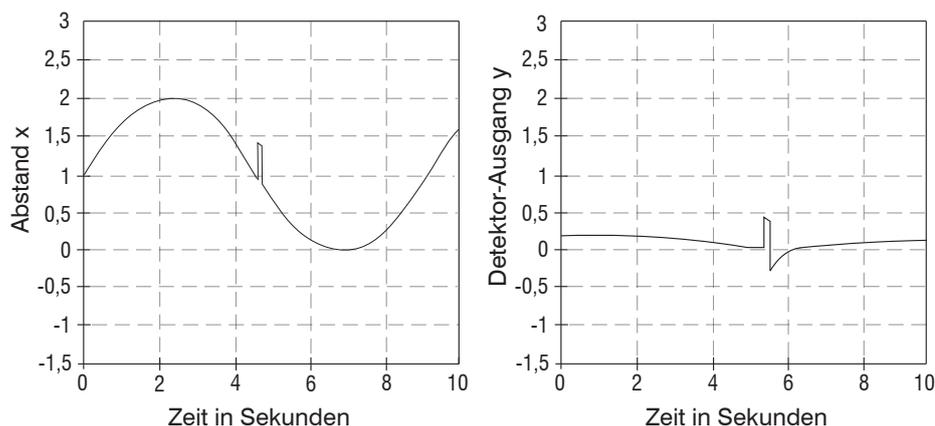


Abb. 8.17: Beispiel für Pulserkennung mit Hilfe der Detektor-Einstellung: Dynamisch ($f_g = 1$ Hz)

8.3.2 Grenzwerttyp-Einstellungen

Bei der Grenzwertüberprüfung gibt es zwei Grundtypen *Grenze* und *Fenster*. Bei *Grenze* wird auf einen Grenzwert A überprüft. Hier wird unterschieden, ob der Schwellwert über- bzw. unterschritten wird, dadurch kann das Ergebnis invertiert werden.

Bei *Fenster* wird überprüft, ob der Schwellwert innerhalb oder außerhalb eines definierten Fensters liegt. Für diese Überprüfung werden zwei Grenzwerte A und B vorgegeben, welche dieses Fenster definieren. Auch hier sind die Ergebnisse bei der Auswahl innerhalb bzw. außerhalb genau invertiert zueinander.

Bei jedem Grenzwert kann zusätzlich eine Hysterese angegeben werden. Diese Hysterese muss für den Wechsel von 1 auf 0 zusätzlich zum Schwellwert über- oder unterschritten werden.

Bei Grenzwerttyp = Aus wird keine Grenzwertüberprüfung durchgeführt, d. h. auch das Status-Bit 6, siehe Kap. 10.5 wird nicht gesetzt.

- 0 = OK
- 1 = NOK

8.3.2.1 Grenzwertüberwachung

Das Abstandssignal wird je nach Einstellung auf über- oder unterschreiten eines Grenzwertes überprüft. Die zu überprüfende Schwelle des Abstandwertes ist vorgegeben.

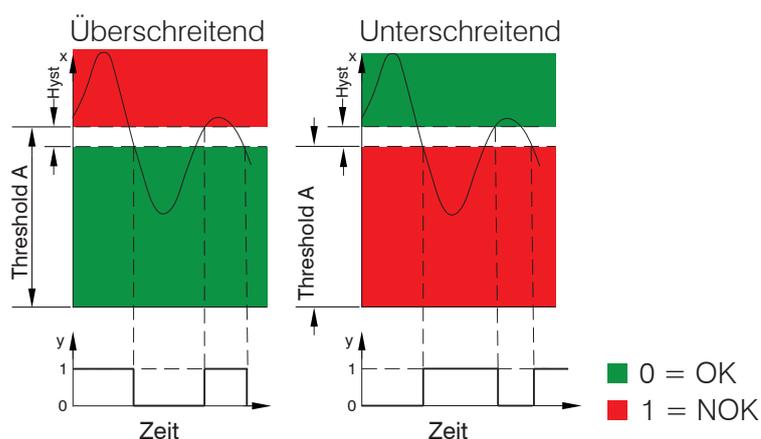


Abb. 8.18: Beispiel für Grenzwerttyp Schaltschwelle über- und unterschreitend

8.3.2.2 Fenster

Es wird überprüft, ob der Wert des Abstandssignals innerhalb oder außerhalb eines Abstandsbereichs liegt. Der Bereich wird durch die beiden Werte Grenzwert A und Grenzwert B definiert. Welcher der beiden Werte größer ist, spielt dabei keine Rolle.

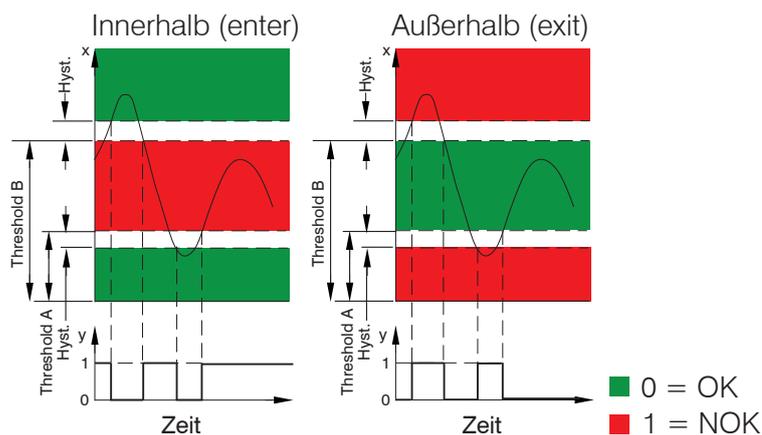


Abb. 8.19: Beispiel für Grenzwerttyp Fenster enter und exit

8.3.2.3 Hysterese

Bei jedem Grenzwert kann zusätzlich eine Hysterese angegeben werden. Diese Hysterese muss für den Wechsel von 1 auf 0 zusätzlich zum Schwellwert über- oder unterschritten werden. Die nachfolgende Grafik zeigt das Ergebnis bei der Auswertung eines Signals mit gleichem Schwellwert, aber unterschiedlich großer Hysterese.

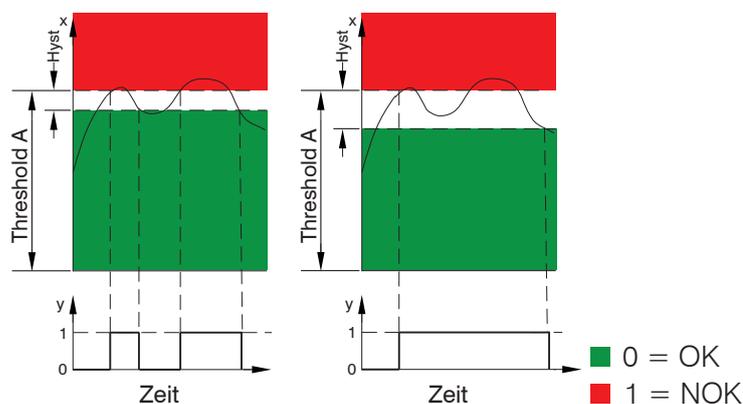


Abb. 8.20: Beispiel für die Wirkung der Hysterese anhand eines Beispielsignals

8.3.3 Zeitbedingungs-Einstellungen

Bei der Grenzwertprüfung kann die Ausgabe über eine Zeitangabe (Delay) verzögert werden. Dies bedeutet, dass z.B. eine Überschreitung eines Grenzwertes mindestens so lange dauern muss, wie die angegebene Verzögerungszeit, damit die Überschreitung zu einer Änderung des Ausgangs führt.

Zudem kann mit einer Haltezeit (Hold) angegeben werden, wie lange der Ausgang nach dem Schalten mindestens aktiv bleiben soll.

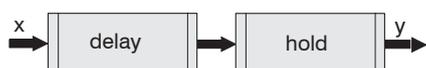


Abb. 8.21: Reihenfolge der Anwendung von delay und hold auf die Grenzwertüberwachung

Die Verzögerung wird zunächst auf das Eingangssignal angewendet. Eine Überschreitung eines Grenzwertes muss demnach mindestens so lange dauern, wie die Verzögerungszeit, damit die Überschreitung zu einer Änderung des Ausgangs führt und das Signal für die angegebene Haltezeit auf 1 (Schalter im aktiven Zustand) bleibt.

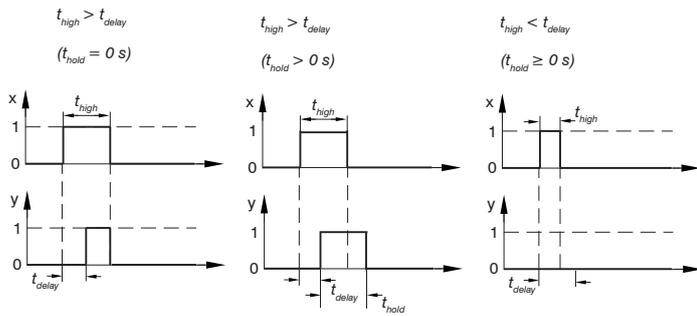


Abb. 8.22: Beispiel für Delayzeit > 0

Die Haltezeit gibt an, wie lange das Signal mindestens auf 1 bleibt. Wird der Grenzwert z.B. für eine längere Zeit überschritten, als dies die Haltezeit vorgibt, so bleibt der Ausgang genau für diese Zeit aktiv. Die Haltezeit wirkt sich damit nur aus, wenn die Grenzwertüberschreitung kürzer ist als die vorgegebene Haltezeit.

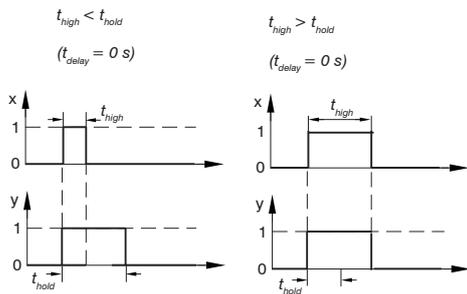


Abb. 8.23: Beispiel für Haltezeit > 0 (Delayzeit = 0)

8.3.4 Ausgangstyp-Einstellungen

Mit der Auswahl des Ausgangstyps wird das Schaltverhalten des Schaltausgang festgelegt, abhängig vom Ergebnis der Grenzwertprüfung. Geschlossen bedeutet, dass der „Schalter GND“ den Schaltausgang mit GND verbindet bzw. beim *Schalter* V_+ der Schaltausgang mit der Betriebsspannung des DT3020 verbunden wird.

Wenn der Ausgangstyp *Aus* gewählt wird, so ist der Ausgang unabhängig von den übrigen Einstellungen und dem Ergebnis der Grenzwertprüfung immer hochohmig. Die Grenzwertüberprüfung findet jedoch immer noch statt und das Status-Bit 6 wechselt je nach Ergebnis den Zustand.

Ausgangstyp-Einstellungen

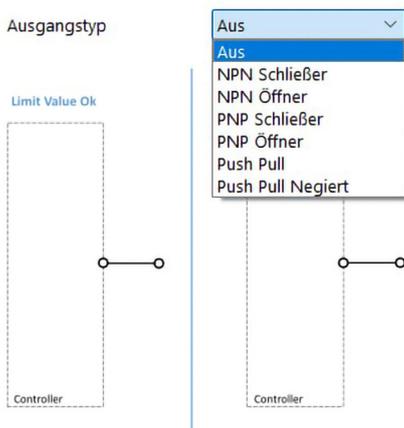


Abb. 8.24: sensorTOOL Ausgangstyp-Einstellungen

Schematische Darstellung der Ausgangskonfiguration

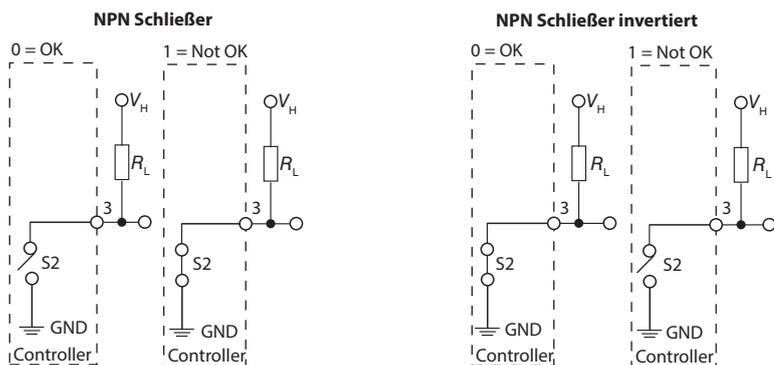


Abb. 8.25: Ausgangstyp NPN

Ein Widerstand R_L muss kundenseitig an eine Spannung (V_H) angeschlossen werden.

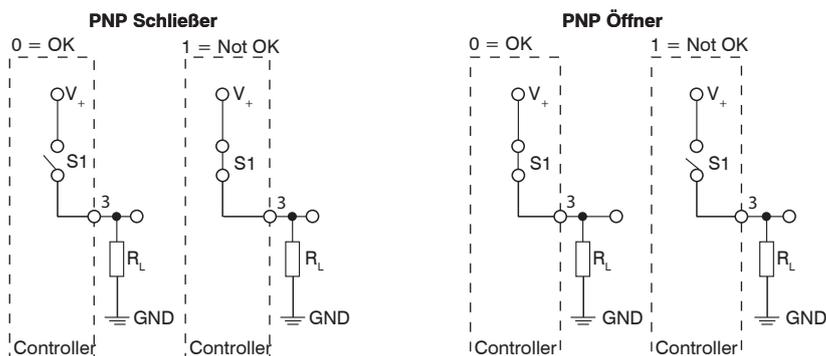


Abb. 8.26: Ausgangstyp PNP

Ein Widerstand R_L muss kundenseitig an GND angeschlossen werden.

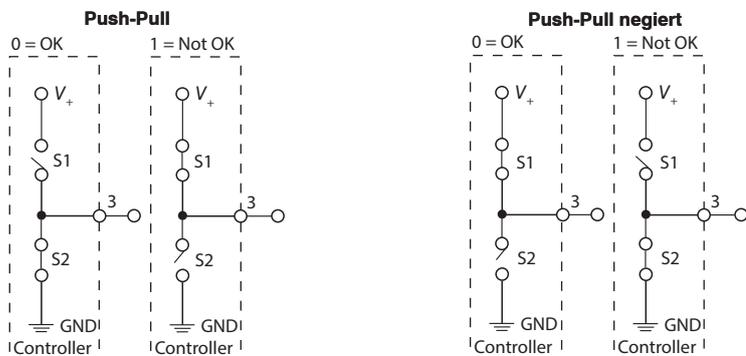


Abb. 8.27: Ausgangstyp Push-Pull

Logischer Ausgangswert der Grenzwertprüfung	0 = Grenzwert OK		1 = Grenzwert NICHT OK	
	Schalter GND S2	Schalter V_+ S1	Schalter GND S2	Schalter V_+ S1
Grenzwerttyp	Schalter GND S2	Schalter V_+ S1	Schalter GND S2	Schalter V_+ S1
Schaltausgang deaktiviert, "Aus"	offen	offen	offen	offen
NPN Schließer	offen	offen	geschlossen	offen
NPN Öffner	geschlossen	offen	offen	offen
PNP Schließer	offen	offen	offen	geschlossen
PNP Öffner	offen	geschlossen	offen	offen
Push-Pull	geschlossen	offen	offen	geschlossen
Push Pull negiert	offen	geschlossen	geschlossen	offen

Tab. 8.2: Wahrheitstabelle für den digitalen Ausgang, abhängig vom gewählten Ausgangstyp

8.4 Status Bits: Temperatur Warningschwellen

Die Einstellung der Temperatur Warningschwellen ist über das Menü auf der linken Seite unter dem Punkt Status Bits möglich. Dort wird die Sensor- und Elektroniktemperatur überwacht. Die Ausgabe von Über-/Unterschreitung einer Schwelle erfolgt durch den Status über die digitale Schnittstelle und als Warnhinweis im sensorTOOL. Überschreitungen der Temperatur-Warningschwellen können nicht am Schaltausgang dargestellt werden.

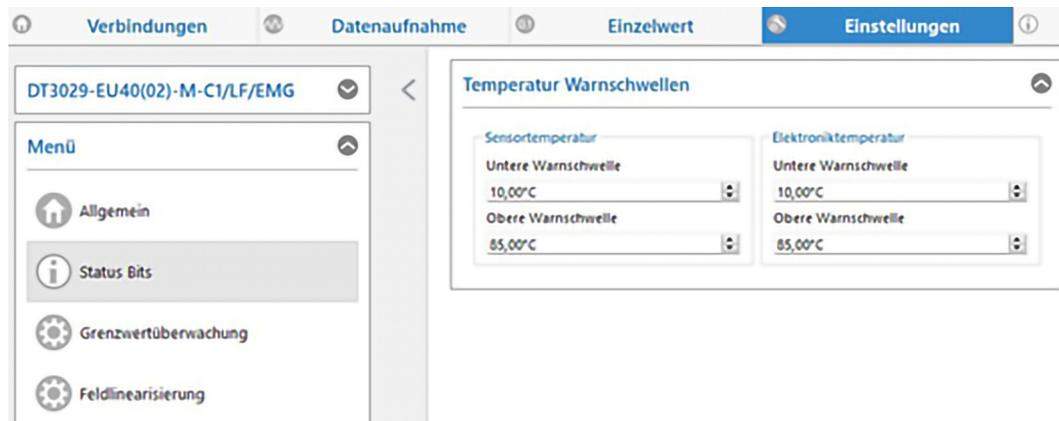


Abb. 8.28: Einstellung der Temperatur Warningschwellen

Der Status der Warningschwelle kann im sensorTOOL unter Einzelwert dargestellt werden, siehe Kap. 7

8.5 Filter und Messraten

Signalverarbeitung

Grenzfrequenz (-3dB) des Filters: 5 kHz

Block-basierte Mittelung über 10 Messwerte

Datenübertragung RS485

Übertragungsart	Abstandswerte pro Paket	Temperaturwerte	Status Bits	Zähler	Auflösung Abstandswert	Effektive Samplerate	Maximale Übertragungsrate USB*
<input checked="" type="radio"/> (1) Einzel, hochauflösend	1	<input checked="" type="checkbox"/>	✓	-	20 Bit	8000,00 Hz	728,93 Hz
<input type="radio"/> (2) Einzel, datenreduziert	1	<input type="checkbox"/>	✓	✓	16 Bit	1000,00 Hz	778,17 Hz
<input type="radio"/> (3) Paket, datenreduziert	15	<input type="checkbox"/>	✓	✓	16 Bit	1000,00 Hz	6214,93 Hz

*bei aktueller Baudrate: 230400 Bd

Datenübertragungsdiagnose

Die Diagnose kann nur bei Übertragungsart (2) oder (3) durchgeführt werden.

Datenübertragung verlustfrei: Starte Diagnose

Verlorene Messwertframes pro Sekunde: ...

Messrate am Eingang: 80000,00 Hz

↓

Messrate am Analogausgang: 8000,00 Hz

↓

Messrate am Digitalausgang: 8000,00 Hz

↓

Maximale Datenrate sensorTOOL (via USB): 728,93 Hz

- Wählen Sie auf der linken Seite den Menüpunkt Filter und Messraten.

8.5.1 Filter und Mittelung im DT3020

Grenzfrequenz

Mit der Auswahl der Grenzfrequenz (-3 dB) des Filters kann die -3 dB Grenzfrequenz eines Tiefpassfilters gewählt werden. Dieser begrenzt die Bandbreite des Abstandssignals, welches am analogen Ausgang, den digitalen Messwerten und der Grenzwertprüfung ausgegeben wird.

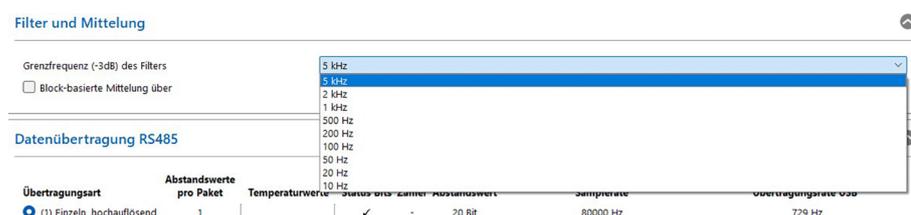


Abb. 8.29: Grenzfrequenz des Filters

Für die Grenzfrequenz von -3 dB lassen sich Werte im Bereich von 10 Hz bis 5 kHz wählen. Diese Einstellung beeinflusst die analog und digital ausgegebenen Messwerte und die Messwerte, welche für die Grenzwertüberprüfung verwendet werden. Die Filtereinstellung wirkt sich nicht auf die interne Datenrate der Signalverarbeitung (analoge Ausgabe und Grenzwertprüfung) aus. Es kann jedoch für jede Filtereinstellung eine individuelle Reduzierung der Datenrate für die Ausgabe eingestellt werden. Neben der Bandbreite, beeinflusst die Filtereinstellung auch die Verzögerung der Messwertausgabe und der Grenzwertüberprüfung.

Filtereinstellung -3 dB Grenzfrequenz	t_{delay} Verzögerung / Delay	t_{rise} Anstiegszeit / Rise Time
5 kHz	0,13 ms	0,07 ms
2 kHz	0,32 ms	0,19 ms
1 kHz	0,54 ms	0,38 ms
500 Hz	0,98 ms	0,76 ms
200 Hz	2,3 ms	1,9 ms
100 Hz	4,6 ms	3,8 ms
50 Hz	9,0 ms	7,6 ms
20 Hz	22,3 ms	19 ms
10 Hz	44,6 ms	38 ms

Tab. 8.3: Typische Werte für die Verzögerung und Anstiegszeit des Signals abhängig von der gewählten Bandbreite

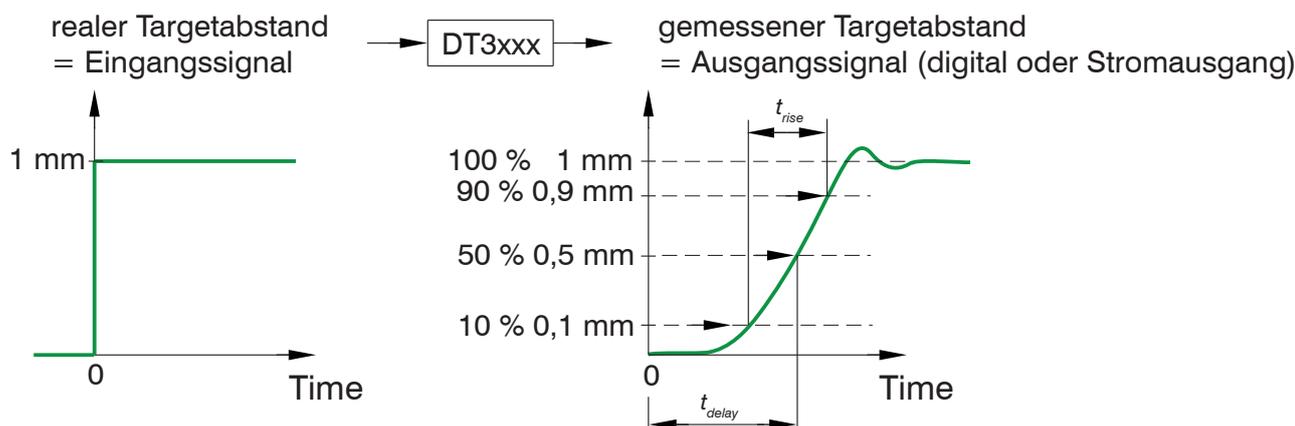


Abb. 8.30: Filterwirkung Skizze

Die Prozentangaben in der obigen Skizze beziehen sich auf den Signalsprung.

Beispiel für Signalsprung von 0,5 auf 0,8 mm:

- 0 % = 0,5 mm
- 10 % = 0,53 mm
- 50 % = 0,65 mm
- 90 % = 0,77 mm
- 100 % = 0,8 mm

Block-basierte Mittelung über x Messwerte

Es kann zusätzlich die Berechnung eines blockweisen arithmetischen Mittelwertes auf die Abstandsmesswerte aktiviert werden. Die Mittelung begrenzt die Bandbreite und damit das Rauschen des Messsignals und reduziert auch die Datenrate der Messwerte. Die blockweise Mittelung wirkt sich ebenfalls auf den analogen Ausgang, die digital ausgegebenen Messwerte und die Grenzwertprüfung aus.

Bei einer Einstellung von 10 wird ein Mittelwert über 10 Messwerte berechnet. Die Messrate wird dadurch von 80 kSa/s auf 8 kSa/s reduziert.

Filter und Mittelung

Grenzfrequenz (-3dB) des Filters

Block-basierte Mittelung über Messwerte

Abb. 8.31: Block-basierte Mittelung

Im DT3020 kann maximal über 4096 Werte gemittelt werden.

Erfolgt die Mittelwertberechnung über N Werte, so ergibt sich erst nach N Werten ein neuer Mittelwert. Dies bedeutet, dass sich die Datenrate um den Faktor N reduziert.

$$M(j) = \frac{1}{N} \sum_{k=0}^{N-1} d(j * N + k)$$

$d(k)$ = k-ter Abstandsmesswert
 N = Anzahl an Werten der Mittelung [1 ... 4096]
 $M(j)$ = j-ter Mittelwert

Abb. 8.32: Formel arithmetischer Mittelwert

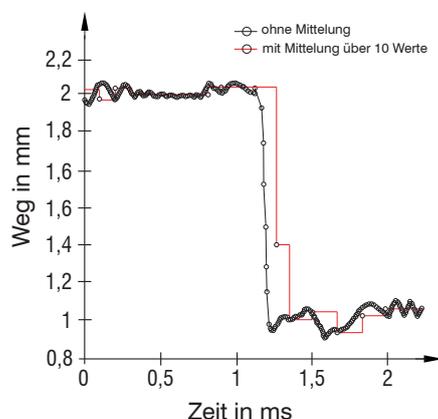


Abb. 8.33: Signal mit Mittelung über 10 Werte

8.5.2 Datenübertragung RS485

Mit dem Drop-down Menü **Datenübertragung RS485** wird ausgewählt, welche Messdaten übertragen und im sensor-TOOL angezeigt werden sollen und in welchem Format diese übertragen werden.

Datenübertragung RS485

Übertragungsart	Abstandswerte pro Paket	Temperaturwerte	Status Bits	Zähler	Auflösung Abstandswert	Effektive Samplerate	Theoretische Übertragungsrate USB*
<input checked="" type="radio"/> (1) Einzel, hochauflösend	1	<input checked="" type="checkbox"/>	✓	-	20 Bit	4000 Hz	729 Hz
<input type="radio"/> (2) Einzel, datenreduziert	1	-	✓	✓	16 Bit	200 Hz	778 Hz
<input type="radio"/> (3) Paket, datenreduziert	115	-	✓	✓	16 Bit	200 Hz	10366 Hz

*bei aktueller Baudrate: 230400 Bd. Reale Übertragungsrate abhängig von PC-Auslastung.

Abb. 8.34: sensorTOOL Datenübertragung

Übertragungsart	Abstandswerte pro Paket	Temperaturwerte	Status Bits	Zähler	Auflösung Abstandswert
Einzel, hochauflösend	1	x (optional)	x	-	20
Einzel, datenreduziert	1	x (optional)	x	x	16
Paket, datenreduziert	115	-	x	x	16

Tab. 8.4: Werkseinstellung für die Unterabtastung abhängig von der Bandbreite

Die Effektive Samplerate ist abhängig von den gewählten Einstellungen. Die Theoretische Übertragungsrate USB ist abhängig von der aktuellen Baudrate, die reale Übertragungsrate hängt von der PC-Auslastung ab.

Die höchste Datenrate wird mit der Einstellung **Paket, datenreduziert** erzielt. Um die höchste Übertragungsrate zu erzielen, wird die maximale Blockgröße von 115 Abstandswerten pro Paket eingestellt. Diese Einstellung ist über den Config Parameter 50067 möglich, siehe Kap. 10

8.5.3 Datenübertragungsdiagnose sensorTOOL

Zum Ausführen dieser Funktion drücken Sie den Button `Starte Diagnose`.

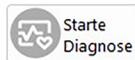
Die Datenübertragungsdiagnose bezieht sich nur auf die Übertragungsarten (2) `Einzeln`, `datenreduziert` und (3) `Paket`, `datenreduziert`. Mit dieser Funktion können Sie überprüfen, ob bei der Datenausgabe Messwert-Frames verlorengehen. Verlustfreie Übertragung wird mit einem grünen Punkt angezeigt, moderater Datenverlust (<1% der Messwerte) mit einem gelben Punkt und kritischer Datenverlust mit einem roten Punkt (die Qualität der Messung am `sensorTOOL` ist nicht sichergestellt). Sie erhalten eine Information darüber, wieviele Frames verlorengehen.

Bei gelbem oder rotem Punkt nehmen Sie gegebenenfalls weitere Einstellungen über, [siehe Kap. 8.5.1](#) vor.

Datenübertragungsdiagnose sensorTOOL

Die Diagnose kann nur bei Übertragungsart (2) oder (3) durchgeführt werden.

Datenübertragung verlustfrei:



Verlorene Messwertframes pro Sekunde: 0/200

Abb. 8.35: sensorTOOL Datenübertragungsdiagnose

9 Info im sensorTOOL

In dieser Ansicht erhalten Sie weiterführende Informationen zu dem verbundenen System wie Controller-, Sensor-, Diagnose- und Kennlinienwerte.

Außerdem können die Informationen in die Zwischenablage kopiert, das System auf Werkseinstellungen zurückgesetzt und Wartezeiten für den ME-Bus Import eingestellt werden.



Indem Sie die Schaltfläche `In Zwischenablage kopieren` betätigen, können Sie die Informationen und Einstellungen zum gewählten Controller in die Zwischenablage speichern.



Indem Sie die Schaltfläche `Werkseinstellungen` betätigen, können Sie den Zustand der Werkseinstellungen wiederherstellen. Alle deaktivierten Kanäle werden wieder aktiviert, die speziell getroffenen kanalbezogenen Einstellungen werden zurückgesetzt. Es öffnet sich ein Fenster mit einer tabellarischen Darstellung aller Parameter, in der die zurückgesetzten Parameter hervorgehoben sowie die alten und neuen Werte angezeigt werden.

10 Config Parameter

Im `sensorTOOL` kann auf die Config Parameter zugegriffen werden.

Auf die Config Parameter kann auch über das Schnittstellenmodul IF2035 und über die MEDAQLib-Software und Treiberbibliothek (SDK), [siehe Kap. 11](#) zugegriffen werden. Relevant sind die Config Parameter für den Betrieb des DT3020, da darüber Einstellungen im Gerät vorgenommen werden können.

Mit der Tastenkombination `STRG+SHIFT+F9` wird im `sensorTOOL` der zusätzliche Menüpunkt `ME-Bus Parameter` aktiviert.



Parameter Name	Parameter ID	Aktueller Wert	Neuer Wert	Min	Max	Typ	Einheit
FitDigitalSel	50063	1		1	9	uint8	
OutBufLength	50067	15		1	115	uint8	
IO1StartAct	50001	4		0	20	float	mA
IO1EndAct	50002	20		0	20	float	mA
TeWarnThresLo	50020	10		10	180	float	°C
TeWarnThresHi	50021	180		10	180	float	°C
TeWarnThresLo	50022	10		-25	150	float	°C
TeWarnThresHi	50023	105		-25	150	float	°C
ProductKey	0	0 0 0 0 0 0 0 0		0	255	uint8	
FitDigitalName	50064	10 Hz		0	126	char	
NoOfSet1	2009	1		1	1	uint8	
OutBufDecim	50072	8		1	1	uint32	
FitDigitalLoad	50070	9		1	9	uint8	
ElivMeasCfgBl	754	114		112	116	uint8	
ElivFilterCtrl	757	0x02 0x00		0	2	bit	
ElivFilterVal	759	10		1	4096	uint32	
Lim1DefFunc	50043	3		1	3	uint8	
Lim1CutoffFreq	50044	0.0004		4e-05	800	float	Hz
Lim1WindowTime	50045	0.01		0.00025	536870.9	float	s
Lim1Type	50046	0		0	4	uint8	
Lim1ThresholdA	50047	100		-105.5	105.5	float	%
Lim1ThresholdB	50048	0		-105.5	105.5	float	%
Lim1Hysteresis	50049	0		0	100	float	%
Lim1DelayTime	50050	0		0	536870.9	float	s
Lim1HoldTime	50051	0		0	536870.9	float	s
IO3SelFct	50010	0		0	6	uint8	
LoadSet1	2004	1		0	1	uint8	
SaveSet1	2005	0		0	1	uint8	
RecetSet1	2008	0		0	1	uint8	
SetTempSet1	2007	1		1	1	uint8	
SetSet1	2006	1		1	1	uint8	
MemProtectUser	50025	0		0	255	uint8	
CalibNameUser	50034	-		0	126	char	
MemProtectMFR	50024	1		0	255	uint8	

Abb. 10.1: ME-Bus Parameter

Rechts oben können Sie über die Schaltfläche `Lese alle Parameter` die Sensorparameter auslesen.

Über das Feld `Neuer Wert` können Sie für die jeweiligen Parameter Werte vergeben und per Befehl `Send` an den Sensor schicken.

10.1 Feldlinearisierung mit Config Parametern durchführen

Beim Controller DT3020 ist die Durchführung einer Feldlinearisierung mit 1, 2 oder 3 Punkten möglich.

Ablauf der Feldlinearisierung

Schreiben Sie die Anzahl n für die jeweils durchzuführende Feldlinearisierung in den Parameter ID 50035

n	=	Anzahl der Punkte, die für eine Feldlinearisierung herangezogen werden (1 für 1-Punkt, 2 für 2-Punkt und 3 für 3-Punkt Feldlinearisierung).
x	=	Auf welchem Messpunkt (1 ... n) bezieht sich der Abstand d .
d	=	Abstand zwischen Sensor und Messobjekt bei Messpunkt x . Es müssen sowohl ein Sollwert also auch ein Istwert eingetragen werden, s. nachfolgende Vorgangsbeschreibung. Diese Werte müssen sich innerhalb des Messbereichs befinden.

- ▶ Stellen Sie den gewünschten Abstand d zwischen Sensor und Messobjekt für Punkt x ein.
- ▶ Schreiben Sie den soeben eingestellten Targetabstand d in mm in den Parameter ID 50040.
- ▶ Schreiben Sie die Nummer des Punktes x in den Parameter ID 50038.
- ▶ Wiederholen Sie die Schritte 1 bis 3 für alle Punkte $x = 1 \dots n$.
- ▶ Starten Sie die Berechnung der n -Punkt Feldlinearisierung, wenn alle Punkte gesetzt wurden. Setzen Sie hierfür den Wert 1 im Parameter ID 50039.

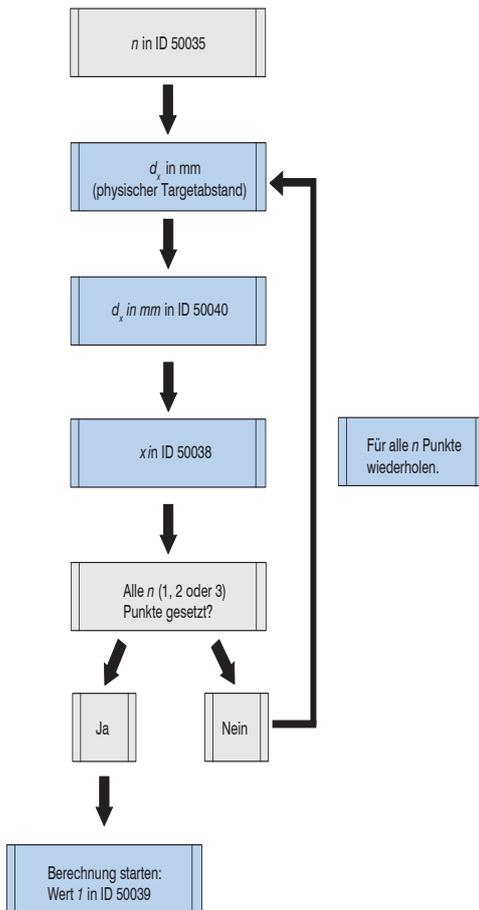


Abb. 10.2: Feldlinearisierung über Config Parametern durchführen

Der Status der Berechnung kann anschließend aus dem Parameter ID 50041 gelesen werden.

Wird 0 gelesen, so ist die Feldkalibrierung erfolgreich durchgeführt und gespeichert worden. Andernfalls ist der gelesene Wert als Bitfeld zu interpretieren, wobei die Bedeutung der einzelnen Bits der nachfolgenden Tabelle zu entnehmen ist.

Bit Nr.	Bedeutung 0	Bedeutung 1	Lösung
0	Kein Fehler	Eingestellte Abstandswerte identisch	Stellen Sie für jeden Punkt (1/2/3) unterschiedliche Abstände zwischen Sensor und Target ein.
1	Kein Fehler	Abstandswerte nicht streng monoton	Wiederholen Sie die Feldlinearisierung. Achten Sie darauf, dass der reale, physikalische Abstand zwischen Sensorfront und Target den eingetragenen Werten d in ParamID 50040 entspricht. Halten Sie diesen Abstand während der Wert x in den ParamID 50038 geschrieben wird.
2	Kein Fehler	Abstandswerte in Begrenzung	Werte liegen außerhalb des Sensormessbereichs. Stellen Sie Werte ein, die sich innerhalb des Messbereichs des jeweiligen Sensors befinden.
3	Kein Fehler	Ungültige Zahl (NaN) enthalten	
4	Kein Fehler	Ungültiger Typ in ParamID 50035	Schreiben Sie gültige Werte: 1, 2 oder 3
5	Kein Fehler	Feldkalibrierung konnte nicht gespeichert werden, da der Schreibschutz in ParamID 50025 gesetzt ist.	Deaktivieren Sie den Schreibschutz in ParamID 50025.

Tab. 10.1: Tabelle für die Interpretation des Wertes ID 50041

- i Die Feldkalibrierung kann mit Parameter ID 50068 = 0 aktiviert und mit = 1 deaktiviert werden. Eingeschaltet werden kann die Kennlinie nur, wenn eine gültige Feldkalibrierung hinterlegt ist. Eine gültige Feldkalibrierung bleibt gespeichert, auch wenn sie deaktiviert wird.

Zur Information wird im Parameter ID 50036 der Typ der aktuell in der aktuellen Kennlinie gespeicherten Feldlinearisierung angezeigt:

0	=	Keine Feldlinearisierung gespeichert.
1	=	1-Punkt Feldlinearisierung
2	=	2-Punkt Feldlinearisierung
3	=	3-Punkt Feldlinearisierung

10.2 Signalverarbeitung: Unterabtastung

Diese Einstellung ist über den Config Parameter ID50072 möglich. Sie reduziert die Datenrate, mit der die digitale Schnittstelle die Daten ausgibt. Die Einstellung hat keine Auswirkung auf den analogen Ausgang oder die Grenzwertüberwachung.

Dies betrifft nur die Datenraten der Messwerte in den Ausgabeformaten `Einzeln`, `datenreduziert` und `Paket`, `datenreduziert`. Die Datenrate für `Einzeln`, `hochauflösend` wird durch diese Einstellung nicht beeinflusst.

Die Messrate des DT3020 beträgt 80 kSa/s. Bei einer eingestellten Unterabtastung von 8, wird nur jeder 8. Wert vom DT3020 ausgegeben. Die über die digitale Schnittstelle gelesenen Werte haben daher eine Datenrate von 10 kSa/s.

Filtertyp	Unterabtastung	Ausgabedatenrate datenreduziert
5 kHz	8	10 kSa/s
2 kHz	20	4 kSa/s
1 kHz	40	2 kSa/s
500 Hz	80	1 kSa/s
200 Hz	80	1 kSa/s
100 Hz	80	1 kSa/s
50 Hz	160	500 Sa/s
20 Hz	400	200 Sa/s
10 Hz	800	100 Sa/s

Tab. 10.2: Werkseinstellung für die Unterabtastung abhängig von der Bandbreite

Die Einstellung der Unterabtastung gilt nur für die Datenformate:

- `Einzeln`, `datenreduziert`
- `Paket`, `datenreduziert`

Die Einstellung wird für jeden Filtertyp individuell im DT3020 gespeichert. Es kann maximal ein Wert von 1 bis 65535 eingestellt werden.

10.3 Liste verfügbarer Config Parameter

Param ID	Datentyp	Name DE	Einheit	Beschreibung DE	Zugriff	min	max
Grenzfrequenz Abstandsmessung							
50063	uint8	Auswahl Filter		Auswählen des digitalen Filters. Die Auswahl wird sofort aktiviert und gespeichert.	rw	1	9
50070	uint8	Digitalfilter Name laden		Lädt den Namen des Digitalfilters mit dieser Nummer in den Parameter 50064	rw	1	9
50064	char[32]	Filter Name		Enthält den Namen des mit 50070 geladen Filters (nicht des aktuell aktiven Filters)	ro	0	126
Feldlinearisierung							

Param ID	Datentyp	Name DE	Einheit	Beschreibung DE	Zugriff	min	max
50035	uint8	Setzen Feldlinearisierung Typ		Setzen des Typs der durchzuführenden Feldlinearisierung 1 = 1-Punkt / Offset 2 = 2-Punkt 3 = 3-Punkt	rw	1	3
50036	uint8	Lesen Feldlinearisierung Typ		Typ der gespeicherten Feldlinearisierung 0 = keine Feldlinearisierung vorhanden 1 = 1-Punkt / Offset 2 = 2-Punkt 3 = 3-Punkt	ro	0	3
50037	uint8	Lesen Abstandswert Stützpunkt		Laden Abstandswertes des Punktes 1 bis 3 in Parameter 50040	rw	1	3
50038	uint8	Setzen Abstandswert Stützpunkt		Beim Schreiben der Zahl wird der Punkt der Feldkalibrierung auf das Wertepaar aktueller Abstandsmesswert und FieldLinValue gesetzt.	rw	1	3
50039	uint8	Speichern der Feldlinearisierung		0 = keine Aktion 1 = Berechnung der Feldkalibrierung wird gestartet. Ist diese erfolgreich, wird die Kalibrierung gespeichert und auch aktiviert.	ro	0	1
50040	float	Abstand Stützpunkt	mm	Lesen des mit 50038 geladenen Wertes. Schreiben des Wertes mit 50038 als Stützstelle der Feldkalibrierung.	rw	-3,40E+38	3,40E+38
50041	uint16	Fehlercodes der Feldlinearisierung		Ergebnis der Berechnung der Feldkalibrierung nach Schreiben von 1 nach 50039. 0 = Feldkalibrierung erfolgreich durchgeführt. Einzelne Bits stehen für Fehler bei der Kalibrierung. Bit 0 = 0x0001 eingestellte Abstandswerte identisch Bit 1 = 0x0002 interne Abstandswerte nicht streng-monoton Bit 2 = 0x0004 interne Abstandswerte in Begrenzung Bit 3 = 0x0008 ungültige Zahl (NaN) enthalten Bit 4 = 0x0010 ungültiger Typ in FieldLinTypeWr (ParamID 50035) Bit 5 = 0x0020 konnte nicht gespeichert werden, da der Schreibschutz in ParamID 50025 gesetzt ist.	ro	0	65535
50068	uint8	Aktivieren Feldlinearisierung		0 = Feldlinearisierung deaktivieren 1 = Feldlinearisierung aktivieren	rw	0	1
Grenzwertschalter							
50010	uint8	Typ des Schaltausgangs		Auswahl der Funktion des Schaltausgangs 0 = Aus 1 = NPNSchliesser 2 = NPNOeffner 3 = PNPSchliesser 4 = PNPOeffner 5 = PushPull 6 = PushPullNegiert	rw	0	6
50051	float	Haltezeit	s	Der Grenzwertschalter bleibt nach dem ansprechen für mindestens diese Zeit aktiv.	rw	0	3,40E+38

Param ID	Datentyp	Name DE	Einheit	Beschreibung DE	Zugriff	min	max
50043	uint8	Detektor		1 = Absolutwert 2 = PeakToPeak blockweise über in 50045 eingestellten Zeitraum 3 = Dynamisch gefiltert Hochpass, mit in 50044 eingestellter Grenzfrequenz	rw	1	3
50044	float	Hochpass Grenzfrequenz	Hz	Grenzfrequenz des Hochpassfilters 1. Ordnung für den Detektor "Dynamisch"	rw	0	3,40E+38
50045	float	Betrachtungszeit	s	Betrachtungszeit für die Bestimmung des Spitze-Spitze-Wertes für den Detektor "PeakToPeak"	rw	0	3,40E+38
50046	uint8	Grenzwerttyp		0 = Grenzwertüberprüfung ausgeschaltet 1 = Grenzwert überschreitend 2 = Grenzwert unterschreitend 3 = Fenster exit 4 = Fenster enter	rw	0	4
50047	float	Grenzwert A	%	Grenzwert auf welchen der Abstandsmesswert überprüft wird.	rw	-105,5	105,5
50048	float	Grenzwert B	%	Zweiter Wert, welcher in Kombination mit Grenzwert A 50047 das Fensters für die Grenzwerttypen "Fenster exit" und "Fenster enter" bestimmt.	rw	-105,5	105,5
50049	float	Hysterese	%	Hysterese der Grenzwertprüfung	rw	0	100
50050	float	Ansprechverzögerung	s	Der Grenzwertschalter spricht um diese Zeit verzögert an.	rw	0	3,40E+38
2000	uint16	Config Parameter Version		Version der allgemeinen Parameter = 2	ro	1	65535
2002	uint16	Parameter Gruppe Identifikation		Identifikationsnummer für Produktgruppe Wirbelstrom = 2	ro	1	65535
2003	uint16	Parameter Gruppe Version		Version der gruppenspezifischen Parameter = 1	ro	1	65535
Kennlinienverwaltung							
2004	uint16	Kennlinie laden		Laden der Informationen der Kennlinie mit der angegebenen Nummer in den Speicher. Beim DT3020 gibt es nur einen Kennlinienspeicherplatz.	rw	0	1
2006	uint8	Kennlinie auswahl permanent		Auswahl der aktuell aktiven Kennlinie mit der angegebenen Nummer. Diese Auswahl wird permanent gespeichert und die Kennlinie ist damit auch beim nächsten Start aktiv.	rw	1	1
2007	uint8	Kennlinie auswahl temporär		Temporäre Auswahl der aktuell aktiven Kennlinie. Nach einem Neustart geht die Auswahl verloren und es ist wieder die mit ParamID 2006 ausgewählte Kennlinie aktiv.	rw	1	1
2005	uint8	Kennlinie speichern		Speichern der Änderungen in der Kennlinie mit der angegebenen Nummer. Betrifft die beiden Parameter 50025 (Benutzer Schreibschutz) und 50034 (Benutzer Kennlinienbezeichnung).	rw	0	1
2008	uint8	Kennlinie Reset		0 = keine Aktion 1 = Kennlinie auf Werkseinstellung zurücksetzen (Feldlinearisierung wird gelöscht, Messbereiche werden auf Nominalwerte zurückgesetzt, ...)	rw	0	1
2009	uint8	Anzahl Kennlinienspeicherplätze		Anzahl der im Gerät verfügbaren Kennlinienspeicherplätze.	ro	1	1
50024	uint8	Schreibschutz Werkskennlinie		0 = Kennlinie ist nicht schreibgeschützt 1 = Kennlinie ist durch Hersteller schreibgeschützt (Werkskennlinie)	ro	0	255

Param ID	Datentyp	Name DE	Einheit	Beschreibung DE	Zugriff	min	max
50025	uint8	Schreibschutz Benutzer		0 = Vollzugriff 1 = Löschschutz 2 = Schreibschutz Wenn ParamID 50024 = 1 ist, dann sind 0 und 1 gleichbedeutend, da eine Werkskennlinie nicht gelöscht werden kann.	rw	0	255
Informationen über Gerät und Sensor							
166	int32	Sensor Artikelnummer		Artikelnummer des Wirbelstromsensors	ro	0	2,15E+09
310	float	Nominaler Messbereich Sensortemperatur	°C	Werkseinstellung für den Messbereich der Sensortemperatur.	ro	-3,40E+38	3,40E+38
342	float	Nominaler Offset Sensortemperatur	°C	Werkseinstellung für den Offset der Sensortemperatur.	ro	-3,40E+38	3,40E+38
311	float	Nominaler Messbereich Elektroniktemperatur	°C	Werkseinstellung für den Messbereich der Elektroniktemperatur.	ro	-3,40E+38	3,40E+38
343	float	Nominaler Offset Elektroniktemperatur	°C	Werkseinstellung für den Offset der Elektroniktemperatur	ro	-3,40E+38	3,40E+38
646	int32	1. Sensorkabel Artikelnummer		Artikelnummer des zu verwendenden	ro	0,00E+00	2,15E+09
647	int32	2. Sensorkabel Artikelnummer		Sensorkabels EC-	ro	0,00E+00	2,15E+09
648	int32	3. Sensorkabel Artikelnummer		Verlängerungskabels ECE-	ro	0,00E+00	2,15E+09
649	int32	4. Sensorkabel Artikelnummer		oder Adapterkabels ECA-	ro	0,00E+00	2,15E+09
650	int32	5. Sensorkabel Artikelnummer		-1 bedeutet, dass das Kabel nicht verwendet wird.	ro	0,00E+00	2,15E+09
651	int32	6. Sensorkabel Artikelnummer			ro	0,00E+00	2,15E+09
198	int32	Sensor Optionsnummer		Optionsnummer des Wirbelstromsensors	ro	0,00E+00	2,15E+09
662	int32	1. Sensorkabel Optionsnummer		Optionsnummer des zu verwendenden	ro	0,00E+00	2,15E+09
663	int32	2. Sensorkabel Optionsnummer		Sensorkabels EC-	ro	0,00E+00	2,15E+09
664	int32	3. Sensorkabel Optionsnummer		Verlängerungskabels ECE-	ro	0,00E+00	2,15E+09
665	int32	4. Sensorkabel Optionsnummer		oder Adapterkabels ECA-	ro	0,00E+00	2,15E+09
666	int32	5. Sensorkabel Optionsnummer		-1 bedeutet, dass das Kabel nicht verwendet wird.		0,00E+00	2,15E+09
667	int32	6. Sensorkabel Optionsnummer			ro	0,00E+00	2,15E+09
262	int32	Sensor Seriennummer		Seriennummer des Wirbelstromsensors	ro	0,00E+00	2,15E+09
694	int32	1. Sensorkabel Seriennummer		Seriennummer des zu verwendenden	ro	0,00E+00	2,15E+09
695	int32	2. Sensorkabel Seriennummer		Sensorkabels EC-	ro	0,00E+00	2,15E+09
696	int32	3. Sensorkabel Seriennummer		Verlängerungskabels ECE-	ro	0,00E+00	2,15E+09

Param ID	Datentyp	Name DE	Einheit	Beschreibung DE	Zugriff	min	max
697	int32	4. Sensorkabel Seriennummer		oder Adapterkabels ECA-	ro	0,00E+00	2,15E+09
698	int32	5. Sensorkabel Seriennummer		-1 bedeutet, dass das Kabel nicht verwendet wird.		0,00E+00	2,15E+09
699	int32	6. Sensorkabel Seriennummer			ro		2,15E+09
50026	uint8	Kalibrierdatum Tag		Datum der Werkskalibrierung - Tag	ro	1	31
50027	uint8	Kalibrierdatum Monat		Datum der Werkskalibrierung - Monat	ro	1	12
50028	uint8	Kalibrierdatum Jahr		Datum der Werkskalibrierung - Jahr	ro	0	255
50029	int32	Abgleich Artikelnummer		Artikelnummer der Werkskalibrierung	ro	0	2,15E+09
50030	int32	Abgleich Optionsnummer		Optionsnummer der Werkskalibrierung	ro	0,00E+00	2,15E+09
50031	int32	Abgleich Seriennummer		Eindeutige, fortlaufende Seriennummer der Werkskalibrierung	ro	0,00E+00	2,15E+09
50032	int32	Abgleich Identifikationsnummer		Eindeutige Identifikationsnummer des Kalibrierungstyps (beinhaltet Target, Messbereich, Temperaturkompensationsbereich, ...)	ro	0,00E+00	2,15E+09
50033	uint8	Kennlinie gültig		0 = kein gültiger Abgleich gespeichert 1 = gültiger Abgleich gespeichert	ro	0,00E+00	1
454	char[32]	Sensor Name		Bezeichnung des verwendeten Wirbelstromsensors	ro	0	126
50034	char[32]	Kennlinie Benutzer Bezeichnung		Durch den Benutzer beschreibbarer Text mit maximal 32 Zeichen	rw	0	126
50069	char[32]	Bezeichnung der Feldkalibrierung		Durch den Benutzer beschreibbarer Text mit maximal 32 Zeichen	ro	0,00E+00	126
50071	float	Gesamtlänge des Sensorkabels	mm	Summe der Länge aller für den Abgleich verwendeten Kabel (integriertes Sensorkabel + Sensorkabel EC- + Verlängerungskabel ECE- + Adapterkabel ECA-)	ro	1,18E-38	3,40E+38
422	char[32]	Targetmaterial		Bezeichnung des Targetmaterials auf welches der Sensor abgeglichen ist.	ro	0,00E+00	1,26E+02
294	float	Nominaler Messbereich Abstand	mm	Werkseinstellung für den Messbereich des Abstandes	ro	-3,40E+38	3,40E+38
358	float	Aktueller Messbereich Abstand	mm	Aktuell verwendeter Messbereich des Abstandes	ro	-3,40E+38	3,40E+38
326	float	Nominaler Offset Abstand	mm	Werkseinstellung für den Offset des Abstandes	ro	-3,40E+38	3,40E+38
390	float	Aktueller Offset Abstand	mm	Aktuell verwendeter Offset des Abstandes	ro	-3,40E+38	3,40E+38
Blockweise Mittelung							
757	bit[2]	Mittelung Status		0 = Mittelung ausgeschaltet 2 = Mittelung eingeschaltet	rw	0	2
759	uint32	Mittelung Anzahl Werte		Anzahl Messwerte [1 ... 4096] über welche ein blockweises arithmetisches Mittel gebildet wird.	rw	1	4096
Messbereich skalieren							
50001	float	Aktueller Start des Ausgabebereichs	mA	aktueller Ausgangsstrom des Stromausgangs bei MBA	rw	4	20
50002	float	Aktuelles Ende des Ausgabebereichs	mA	aktueller Ausgangsstrom des Stromausgangs bei MBE	rw	4	20
50061	float	Benutzer Messbereich	mm	Durch den Benutzer änderbarer Messbereich der Abstandsmessung	rw	0	2

Param ID	Datentyp	Name DE	Einheit	Beschreibung DE	Zugriff	min	max
500062	float	Benutzer Offset	mm	Durch den Benutzer änderbarer Offset der Abstandsmessung	rw	0,2	2,2
50003	float	Werkseinstellung Start des Ausgabebereichs	mA	Werkseinstellung des Ausgangsstroms am Stromausgang bei MBA	ro	4	20
50004	float	Werkseinstellung Ende des Ausgabebereichs	mA	Werkseinstellung des Ausgangsstroms am Stromausgang bei MBE	ro	4	20
Digitale Messwertausgabe							
754	uint8	Ausgabedatenformat, siehe Kap. 10.4.1		0x70 Multiple distances from FIFO, state, counter 0x71 Single high resolution distance, sensor temp., electronic temp., state 0x72 Single high resolution distance, state 0x73 Single distance from FIFO, state, counter 0x74 Single distance from FIFO, sensor temp., electronic temp, state, counter	rw	112	116
2016	float	Ausgabedatenrate	Hz	Ausgabedatenrate auf der digitalen Schnittstelle	ro	0	4,29E+09
50067	uint8	Ausgabepuffer Länge		Anzahl an Messwerten, welche im FIFO Ausgabepuffer maximal gespeichert werden. Diese Einstellung wird nur verwendet wenn das Ausgabeformat 754 auf 0x70 gesetzt ist und die Messwerte Blockweise ausgegeben werden.	rw	1	115
50072	uint32	Ausgabepuffer Datenreduzierung		Es wird nur jeder x-te Abstandsmesswert in den Ausgabepuffer übernommen. Es gilt Ausgabedatenrate = Messdatenrate/x. Diese Einstellung wird nur verwendet wenn das Ausgabeformat 754 auf 0x70, 0x73 oder 0x74 gesetzt ist.	rw	1	65535
50042	uint16	Status des Systems		Bit 0 = 0x0001 Messsignal außerhalb des Messbereichs Bit 1 = 0x0002 kein Sensor vorhanden / Bruch des Sensorkabels Bit 2 = 0x0004 Sensortemperatur außerhalb der Grenzen (Vorgabe ab Werk) Bit 3 = 0x0008 Elektroniktemperatur außerhalb der Grenzen (Vorgabe ab Werk) Bit 4 = 0x0010 Sensortemperatur außerhalb der Benutzergrenzen (50020, 50021) Bit 5 = 0x0020 Elektroniktemperatur außerhalb der Benutzergrenzen (50022, 50023) Bit 6 = 0x0040 Messwert außerhalb der Grenzen der Grenzwertprüfung	ro	0	65535
0	uint8[8]				rw	0	255
Warnlevel für die Sensor- und Elektroniktemperatur							

Param ID	Datentyp	Name DE	Einheit	Beschreibung DE	Zugriff	min	max
50020	float	Sensortemperatur unter Warnschwelle	°C	Durch Benutzer einstellbare untere Warnschwellen für die Sensortemperatur. Liegt die Sensortemperatur kleiner wird das entsprechende Status-Bit in 50042 gesetzt.	rw	10	180
50021	float	Sensortemperatur obere Warnschwelle	°C	Durch Benutzer einstellbare obere Warnschwellen für die Sensortemperatur. Liegt die Sensortemperatur größer wird das entsprechende Status-Bit in 50042 gesetzt.	rw	10	180
50022	float	Elektroniktemperatur untere Warnschwelle	°C	Durch Benutzer einstellbare untere Warnschwellen für die Elektroniktemperatur. Ist Elektroniktemperatur kleiner wird das entsprechende Status-Bit in 50042 gesetzt.	rw	-25	150
50023	float	Elektroniktemperatur obere Warnschwelle	°C	Durch Benutzer einstellbare obere Warnschwellen für die Elektroniktemperatur. Ist die Elektroniktemperatur größer, wird das entsprechende Status-Bit in 50042 gesetzt.	rw	-25	150

10.4 Digitalwerte

10.4.1 Ausgabedatenformate

Übersicht über die im Config Parameter 754 einstellbaren Datenformate.

Parameter ID ^[14]	Block Nr. ^[15]	Format
0x70	0x80	Multiple distances from FIFO, state, counter
0x71	0x81	Single high resolution distance, sensor temp., electronic temp., state
0x72	0x82	Single high resolution distance, state
0x73	0x83	Single distance from FIFO, state, counter
0x74	0x84	Single distance from FIFO, sensor temp., electronic temp, state, counter

Datenformat Block 0x80 (Config Parameter ID754 = 0x70)

Data type	Name	Description
uint8	system state	State of the DT3020, siehe Kap. 10.5
uint8	filler	always zero
uint16	measurement counter	Measurement Counter value of the first (oldest) measurement in the data package.
uint8	number values	Number of measurement values <i>n</i> received
uint8	filler	always zero
int16	distance[0]	
...		
....		
int16	distance[n-1]	

Datenformat Block 0x81 (Config Parameter ID754 = 0x71)

Data type	Name	Description
uint8	system state	State of the DT3020, siehe Kap. 10.5
uint8	filler	always zero
uint8	filler	always zero

[14] Über die Config Parameter ID kann der Ausgabeblock für den zyklischen Datenaustausch geändert werden.

[15] Lesen mit MEDAQLib Befehl `Get_AlternateMeasure` oder ME-Bus Befehl `Block lesen`.

uint8	filler	always zero
int32	distance	high resolution distance
uint16	sensor temperature	
uint16	electronic temperature	

Datenformat Block 0x82 (Config Parameter ID754 = 0x72)

Data type	Name	Description
uint8	system state	State of the DT3020, siehe Kap. 10.5
uint8	filler	always zero
uint8	filler	always zero
uint8	filler	always zero
int32	distance	high resolution distance

Datenformat Block 0x83 (Config Parameter ID754 = 0x73)

Data type	Name	Description
uint8	system state	State of the DT3020, siehe Kap. 10.5
uint8	filler	always zero
uint16	measurement counter	The internal measurement counter is incremented with every sample.
int16	distance	distance

Datenformat Block 0x84 (Config Parameter ID754 = 0x74)

Data type	Name	Description
uint8	system state	State of the DT3020, siehe Kap. 10.5
uint8	filler	always zero
uint16	measurement counter	The internal measurement counter is incremented with every sample.
int16	distance	distance
uint16	sensor temperature	
uint16	electronic temperature	

Berechnung der Abstands- und Temperaturwerte aus den empfangenen Messwerten.

10.4.2 Berechnung des Abstandswertes

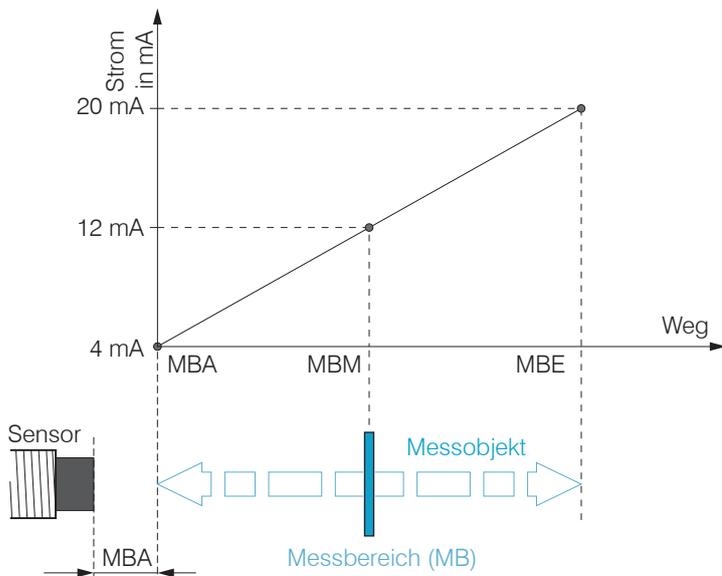
Der Messbereich des Sensors kann auch aus den Config Parametern ausgelesen werden.

Begriffsdefinition:

- MB: Messbereich
- MBA: Messbereichsanfang
- MBM: Messbereichsmittle
- MBE: Messbereichsende

Abstände mit MBA entsprechen dem Abstand zwischen Sensorfront und Target.

Abstände ohne MBA berücksichtigen nicht den sogenannten Offset am Anfang des Messbereichs.



MB in mm = Parameter-ID 358

MBA in mm = Parameter-ID 390

Entfernung mit MBA: $MBA = MBA$; $MBM = MBA + MB/2$; $MBE = MBA + MB$

Entfernung ohne MBA: $MBA = 0$; $MBM = MB/2$; $MBE = MB$

Berechnung digitaler Distanzwerte für einen U3 Sensor mit einem Messbereich von 3 mm. (MBA = 0.3 mm, MBE = 3.3 mm).

10.4.2.1 int16 distance (Config Parameter 754 = 0x70, 0x73, 0x74)

	Entfernung in digit	Entfernung in mm	
		mit MBA	ohne MBA
MBA	-29696	0,3	0
MBM	0	1,8	1,5
MBE	29696	3,3	3

Umrechnung der Digitalwerte

Abstand mit MBA: $d = MBA + \frac{MB}{2} * (1 + \frac{x}{29696})$

Abstand ohne MBA: $d = \frac{MB}{2} * (1 + \frac{x}{29696})$

10.4.2.2 int32 distance (Config Parameter 754 = 0x71, 0x72)

	Entfernung in digit	Entfernung in mm	
		mit MBA	ohne MBA
MBA	-475136	0,3	0
MBM	0	1,8	1,5
MBE	475136	3,3	3

Umrechnung der Digitalwerte

Abstand mit MBA: $d = MBA + \frac{MB}{2} * (1 + \frac{x}{475136})$

Abstand ohne MBA: $d = \frac{MB}{2} * (1 + \frac{x}{475136})$

10.4.3 Berechnung der Sensor- und Elektroniktemperatur

Umrechnung der Digitalwerte

$$\text{Temperatur: } T = \frac{(x_s - 12800)}{128} \text{ } ^\circ\text{C} = \left(\frac{x_s}{128} - 100\right) \text{ } ^\circ\text{C}$$

$$x_s = [0 \dots 64512]$$

$$T = [-100 \text{ } ^\circ\text{C} \dots + 404 \text{ } ^\circ\text{C}]$$

Sensortemperatur

MB in $^\circ\text{C}$ = Parameter-ID 310 ist 504 $^\circ\text{C}$

MBA in $^\circ\text{C}$ = Parameter-ID 342 ist -100 $^\circ\text{C}$

$$\text{MBE} = \text{MBA} + \text{MB} = 404 \text{ } ^\circ\text{C}$$

Elektroniktemperatur

MB in $^\circ\text{C}$ = Parameter-ID 311 ist 504 $^\circ\text{C}$

MBA in $^\circ\text{C}$ = Parameter-ID 343 ist -100 $^\circ\text{C}$

$$\text{MBE} = \text{MBA} + \text{MB} = 404 \text{ } ^\circ\text{C}$$

10.5 Bedeutung des Status

Die Werte in State 1 geben Fehlerzustände und den Zustand der Grenzwertüberprüfung aus. State 2 wird aktuell nicht genutzt und ist immer 0.

Der Wert des Statusbytes State 1 ist als Bitfeld zu interpretieren, d.h.:

- ▶ Angezeigten Wert in Binärdarstellung umrechnen
- ▶ Status des jeweiligen Statusbits aus der Tabelle für die Interpretation von *State 1* entnehmen.

Tabelle für die Interpretation von State 1

Bit Nr.	Bedeutung 0	Bedeutung 1
0	Messsignal im Messbereich	Messsignal außerhalb des Messbereichs
1	Sensor vorhanden	Bruch des Sensorkabels / kein Sensor vorhanden
2	Sensortemperatur innerhalb der ab Werk vorgegebenen Grenzen	Sensortemperatur außerhalb der ab Werk vorgegebenen Grenzen
3	Elektroniktemperatur innerhalb der ab Werk vorgegebenen Grenzen	Elektroniktemperatur außerhalb der ab Werk vorgegebenen Grenzen
4	Sensortemperatur innerhalb der ab werk vorgegebenen Grenzen	Sensortemperatur außerhalb der ab Werk vorgegebenen Grenzen
5	Elektroniktemperatur innerhalb der benutzerdefinierten Warn Grenzen	Elektroniktemperatur außerhalb der benutzerdefinierten Warn Grenzen
6	Grenzwert OK (Grenzwert wird eingehalten)	Grenzwert NICHT OK (Grenzwert wird nicht eingehalten)
7	Status des Schaltausgangs: OK	Status des Schaltausgangs: NICHT OK (Überlast oder Übertemperatur)

Beispiel für die Interpretation von State 1

- Angezeigter Wert: State 1 = 5
- Wert 5 in Binärdarstellung: $5 \triangleq 00000101$
- Bit Nr. 0 und Bit Nr. 2 sind gesetzt.
- Aus Tabelle: Messsignal außerhalb des Messbereichs und Sensortemperatur kritisch.

Das Statusbyte *State 2* hat beim DT3020 immer den Wert 0 und ist daher nicht relevant.

11 Softwareunterstützung mittels MEDAQLib

Die MEDAQLib-Software und Treiberbibliothek (SDK) verbindet Micro-Epsilon Sensoren schnell und einfach mit Ihrer Software. Diese können Sie über die Micro-Epsilon Website herunterladen <https://www.micro-epsilon.de/fileadmin/download/software/MEDAQLib.zip>.

Die MEDAQLib wird als Dynamic Link Library (DLL) ausgeliefert. Dank ihrer einheitlichen C-Schnittstelle ist die Integration und Nutzung in verschiedenen Programmiersprachen möglich, z. B. C#, C++, VisualBasic, LabView, aber auch MATLAB und Python.

Softwarebeispiele für gängige Programmiersprachen finden Sie im Anhang , [siehe Kap. 17.5](#)

Unterstützte ME-Bus Sensorbefehle

Befehl	Unterstützt	Bemerkung
Logout	Nein	
Login	Nein	
Get_UserLevel	Nein	
Set_Password	Nein	
Set_Samplerate	Nein	Die Samplerate kann nicht direkt verändert werden. Eine Änderung ist nur über die Konfigurationsparameter, siehe Kap. 10.3 der Schnittstelle möglich.
Get_Samplerate	Ja	
Set_Trigger	Nein	
Get_Trigger	Nein	
Set_Averaging	Ja	
Get_Averaging	Ja	Mittelung Typ = (0;2), Mittelung Wert = (0; 4096)
Get_Measure	Ja	
Get_AlternateMeasure	Ja	
Set_ContinuousMode	Nein	
Get_ContinuousMode	Ja	
Set_Range	Ja	
Test_Baudrate	Ja	
Set_Baudrate	Ja	
Get_Baudrate	Ja	
Set_SensorAddress	Ja	
Get_SensorInfo	Ja	
Get_ChannelInfo	Ja	
Get_ChannelInfos	Ja	
Get_ControllerInfo	Ja	
Get_DiagnosticInfo	Ja	
Get_ConfigDescription	Ja	
Set_ConfigParameter	Ja	Detaillierte Beschreibung, siehe Kap. 10
Get_ConfigParameter	Ja	Detaillierte Beschreibung, siehe Kap. 10
Read_AllBlocks	Ja	

11.1 Config Parameter lesen und schreiben

Kapitel 8 Config Parameter enthält eine Liste mit Konfigurationsparametern. Diese Konfigurationsparameter können mit der MEDAQLib gelesen und geschrieben werden. Zum Lesen dient `Get_ConfigParameter` und zum schreiben `Set_ConfigParameter`. Zur Identifikation des Parameters wird die Config Parameter ID genutzt, welche in der MEDAQLib vor dem Schreiben oder Lesen mit dem Parameter `SP_ParameterID` gesetzt werden muss. Der MEDAQLib Parameter `SP_ParameterName` wird dann nicht verwendet und muss daher nicht gesetzt werden.

Der mit dem Befehl `Get_ConfigParameter` gelesene Wert eines Config Parameters ist als Text in dem MEDAQLib Parameter `SA_Value` enthalten.

Um einen Parameter mit `Set_ConfigParameter` zu schreiben, muss dieser vorher in den MEDAQLib Parameter `SP_Value` als Text geschrieben werden.

11.2 Messwerte lesen

Die mit dem MEDAQLib Befehl `Get_Measure` vom Controller DT3020 gelesenen Messwerte können über das Ausgabeformat mit dem Config Parameter der ID 754 verändert werden, [siehe Kap. 10.4.1](#)

Zudem ist es möglich mit dem MEDAQLib Befehl `Get_AlternateMeasure` wahlweise auch die anderen Messwerte zu lesen ohne das Ausgabeformat umstellen zu müssen. Der Befehl für das Lesen der alternativen Ausgabeblöcke ist länger. Daher sollte für die maximale Datenrate immer der Befehl `Get_Measure` verwendet werden. Der Befehl `Get_AlternateMeasure` kann z. B. dazu genutzt werden, um Temperaturwerte zu lesen, während mit `Get_Measure` blockweise Abstandswerte aus dem FIFO übertragen werden.

Die im Parameter `SP_AlternateBlockIdx` angegebene Blocknummer ist ebenfalls in der Tabelle , [siehe Kap. 10.3](#) angegeben.

11.3 Feldlinearisierung

Eine Feldlinearisierung mit der MEDAQLib erfolgt über die Config Parameter. Der Ablauf dafür ist hier beschrieben , [siehe Kap. 10.1](#).

12 Diagnosemeldungen

Die Fehler- und Warnmeldungen werden im `sensorTOOL` unter `Info` bei `Diagnose Informationen` ausgegeben, bzw. über das blinkende Symbol in der rechten unteren Ecke, das `Status Log`. Bei der Schnittstelle IF2035 Profinet gibt es für die Ausgabe dieser Meldungen den Index 0x2213, aus welchem die Fehlercodes ausgelesen werden können.

12.1 Fehlercodes

Code	Beschreibung
E16	ParamId 759: Anzahl der Samples für die Mittelung ist zu groß.
E17	ParamId 759: Anzahl der Samples für die Mittelung ist kleiner als 1.
E18	ParamId 757: Ungültiger Filtertyp geschrieben.
E19	ParamId 754: Ungültiges Ausgabedatenformat angegeben.
E20	ParamId 50067: Länge des Ausgangspuffers außerhalb des gültigen Bereichs.
E21	ParamId 50063: Einzustellende Filterkonfiguration ist nicht vorhanden.
E22	ParamId 2004, 2007, 2005, 2008: Mehr als ein Befehl der Kennlinienverwaltung wurde gleichzeitig geschrieben.
E23	ParamId 2004: Daten der Kennlinie können nicht geladen werden.
E24	ParamId 2005: Daten der Kennlinie können nicht gespeichert werden, da der Speicherplatz nicht vorhanden ist.
E25	ParamId 2005: Speichern der Kennlinie ist nur möglich, wenn zuvor mit 2004 eine Kennlinie geladen wurde.
E26	ParamId 2005: Speichern der Kennlinie ist nicht möglich. Die aktuell geladene Kennlinie ist ungültig.
E27	ParamId 2005: Speichern der Kennlinie ist auf den angegebenen Speicherplatz nicht möglich, da dieser bereits belegt ist. Die dort hinterlegte Kennlinie muss zuerst gelöscht werden.
E28	ParamId 2008: Angegebene Kennlinie kann nicht zurückgesetzt werden.
E29	ParamId 2008: Angegebene Kennlinie ist gerade aktiv und die Feldkennlinie ist schreibgeschützt.
E30	ParamId 2008: Kennlinie ist schreibgeschützt.
E31	ParamId 50025: Wert für den Schreibschutz ist ungültig.
E32	ParamId 2007: Angegebene Kennlinienspeicherplatz außerhalb des gültigen Bereichs.
E33	ParamId 2007: Angegebene Kennlinie ist nicht vorhanden oder fehlerhaft.
E34	ParamId 2006: Angegebene Kennlinienspeicherplatz außerhalb des gültigen Bereichs.
E35	ParamId 2006: Angegebene Kennlinie ist nicht vorhanden oder fehlerhaft.
E36	ParamId 50039: Der mit ParamId 50035 gesetzte Typ der Feldlinearisierung ist ungültig.
E37	ParamId 50035: Typ der Feldlinearisierung ist ungültig.
E38	ParamId 50068: Ungültigen Wert geschrieben. Zustand der Feldlinearisierung wird nicht geändert.
E39	ParamId 50039: Feldlinearisierung konnte nicht gespeichert werden, da der Schreibschutz in 50025 gesetzt ist.
E40	ParamId 50068: Feldlinearisierung kann nicht aktiviert/deaktiviert werden, da der Schreibschutz in 50025 gesetzt ist.
E41	ParamId 50068: Feldlinearisierung kann nicht aktiviert werden, da in der Kennlinie keine gültige Feldlinearisierung vorhanden ist.
E42	ParamId 50070: Filter ist nicht vorhanden.
E45	ParamId 50072: Dezimierungsfaktor ist außerhalb des gültigen Bereichs.

12.2 Warnungen

Code	Beschreibung/Ursache für User
W1	ParamId 2008: Die Feldkennlinie der aktuell aktiven Kennlinie wurde gelöscht.
W2	ParamId 2005: Benutzer Schreibschutz in ParamId 50025 ist gesetzt. Teile der Kennlinie konnten daher nicht gespeichert werden.
W3	ParamId 50025: Ungültiger Wert für den Benutzer Schreibschutz geschrieben. Aktueller Zustand des Schreibschutz wurde daher nicht verändert.
W4	ParamId 50025, 50034: Lesen von Kennlinieninformationen ohne, dass vorher die Informationen einer gültigen Kennlinie mit ParamId 2004 geladen wurden.
W5	Lesen von Kennlinieninformationen ohne, dass vorher die Informationen einer gültigen Kennlinie mit ParamId 2004 geladen wurden.
W13	ParamId 50037: Keine gültige Feldkalibrierung vorhanden oder die aktuell gültige Feldkalibrierung enthält weniger Punkte.
W14	ParamId 50071: Lesen von Kennlinieninformationen ohne, dass vorher die Informationen einer gültigen Kennlinie mit ParamId 2004 geladen wurden.

13 Fehlerbehebung

Fehler	Grund und Lösung
Ausgangssignal in positiver oder negativer Sättigung, abhängig von der Skalierung des Analogausganges.	<ul style="list-style-type: none"> • Target außerhalb des Messbereichs • Kabel und/oder Sensor sind nicht angeschlossen • Sensor ist defekt (z.B. Unterbrechung, Kurzschluss) • Kabel ist defekt <p>Beachten Sie die Hinweise im sensorTOOL. Ersetzen Sie Kabel und/oder Sensor.</p>
Ausgangssignal oszilliert mit geringer Frequenz bei Mehrkanalbetrieb.	<ul style="list-style-type: none"> • Gegenseitige Beeinflussung durch Interferenzen <p>Beachten Sie die Hinweise zur Sensoranordnung mit LF- und HF-Band, s. Kapitel Messaufbau, Einsatz mehrerer Sensoren</p>
Keine Ausgangssignaländerung.	<ul style="list-style-type: none"> • Überprüfen Sie die Versorgungsspannung. • Überprüfen Sie die Zuordnung von Sensortyp und Kabellänge. • Überprüfen Sie Sensor und Kabel.

14 Haftungsausschluss

Alle Komponenten des Gerätes wurden im Werk auf die Funktionsfähigkeit hin überprüft und getestet. Sollten jedoch trotz sorgfältiger Qualitätskontrolle Fehler auftreten, so sind diese umgehend an Micro-Epsilon oder den Händler zu melden.

Micro-Epsilon übernimmt keinerlei Haftung für Schäden, Verluste oder Kosten, die z.B. durch

- Nichtbeachtung dieser Anleitung / dieses Handbuches,
- Nicht bestimmungsgemäße Verwendung oder durch unsachgemäße Behandlung (insbesondere durch unsachgemäße Montage, - Inbetriebnahme, - Bedienung und - Wartung) des Produktes,
- Reparaturen oder Veränderungen durch Dritte,
- Gewalteinwirkung oder sonstige Handlungen von nicht qualifizierten Personen

am Produkt entstehen, entstanden sind oder in irgendeiner Weise damit zusammenhängen, insbesondere Folgeschäden.

Diese Haftungsbeschränkung gilt auch bei Defekten, die sich aus normaler Abnutzung (z. B. an Verschleißteilen) ergeben, sowie bei Nichteinhaltung der vorgegebenen Wartungsintervalle (sofern zutreffend).

Für Reparaturen ist ausschließlich Micro-Epsilon zuständig. Es ist nicht gestattet, eigenmächtige bauliche und/oder technische Veränderungen oder Umbauten am Produkt vorzunehmen. Im Interesse der Weiterentwicklung behält sich Micro-Epsilon das Recht auf Konstruktionsänderungen vor.

Im Übrigen gelten die Allgemeinen Verkaufsbedingungen der Micro-Epsilon, die unter Impressum | Micro-Epsilon <https://www.micro-epsilon.de/impressum/> abgerufen werden können.

15 Service, Reparatur

Bei einem Defekt am Messsystem:

- Speichern Sie nach Möglichkeit die aktuellen Sensoreinstellungen in einem Parametersatz, um nach der Reparatur die Einstellungen wieder in den Controller laden zu können.
- Senden Sie bitte die betreffenden Teile zur Reparatur oder zum Austausch ein.

Bei Störungen, deren Ursachen nicht eindeutig erkennbar sind, senden Sie bitte immer das gesamte System inklusive Sensor und aller Kabel an:

MICRO-EPSILON MESSTECHNIK
GmbH & Co. KG
Königbacher Str. 15
94496 Ortenburg / Deutschland

Tel: +49 (0) 8542 / 168-0
Fax: +49 (0) 8542 / 168-90
info@micro-epsilon.de
www.micro-epsilon.com/contact/worldwide/
<https://www.micro-epsilon.de>

16 Außerbetriebnahme, Entsorgung

Um zu vermeiden, dass umweltschädliche Stoffe freigesetzt werden und um die Wiederverwendung von wertvollen Rohstoffen sicherzustellen, weisen wir Sie auf folgende Regelungen und Pflichten hin:

- Sämtliche Kabel am Sensor und/oder Controller sind zu entfernen.
- Der Sensor und/oder Controller, dessen Komponenten und das Zubehör sowie die Verpackungsmaterialien sind entsprechend den landesspezifischen Abfallbehandlungs- und Entsorgungsvorschriften des jeweiligen Verwendungsgebietes zu entsorgen.
- Sie sind verpflichtet, alle einschlägigen nationalen Gesetze und Vorgaben zu beachten.

Für Deutschland / die EU gelten insbesondere nachfolgende (Entsorgungs-) Hinweise:

- Altgeräte, die mit einer durchgestrichenen Mülltonne gekennzeichnet sind, dürfen nicht in den normalen Betriebsmüll (z.B. die Restmülltonne oder die gelbe Tonne) und sind getrennt zu entsorgen. Dadurch werden Gefahren für die Umwelt durch falsche Entsorgung vermieden und es wird eine fachgerechte Verwertung der Altgeräte sichergestellt.



- Eine Liste der nationalen Gesetze und Ansprechpartner in den EU-Mitgliedsstaaten finden Sie unter https://ec.europa.eu/environment/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee_en. Hier besteht die Möglichkeit, sich über die jeweiligen nationalen Sammel- und Rücknahmestellen zu informieren.

- Altgeräte können zur Entsorgung auch an Micro-Epsilon an die im Impressum unter <https://www.micro-epsilon.de/impressum> angegebene Anschrift zurückgeschickt werden.

- Wir weisen darauf hin, dass Sie für das Löschen der messspezifischen und personenbezogenen Daten auf den zu entsorgenden Altgeräten selbst verantwortlich sind.

- Unter der Registrierungsnummer WEEE-Reg.-Nr. DE28605721 sind wir bei der Stiftung Elektro-Altgeräte Register, Nordostpark 72, 90411 Nürnberg, als Hersteller von Elektro- und/ oder Elektronikgeräten registriert.

17 Anhang

17.1 Optionales Zubehör

PS2020



Netzteil für Hutschienenmontage
Eingang 230 VAC, Ausgang 24 VDC/2,5 A

IF1032/ETH



Schnittstellenmodul

- Ethernet / EtherCAT
- 1x RS485 (ME-internes Protokoll)
- 2x Analog-In (14 Bit, max. 4 kSps), Spannung
- 1x Analog-in, (14 Bit, max. 4 kSps) Strom
- Eingänge für Versorgungsspannung
- Triggereingang
- EtherCAT-Synchronisations-Ausgang
- Ausgang für Spannungsversorgung des Sensors

IF2035-EtherCAT
IF2035-PROFINET
IF2035-EtherNet/IP



Schnittstellenmodul zur Anbindung an EtherCAT, PROFINET oder EtherNet/IP eines Micro-Epsilon Sensors mit RS485 oder RS422-Schnittstelle; Hutschienengehäuse, inkl. Gerätebeschreibungsfdatei zur Softwareeinbindung in der SPS

MC25D



Digitale Mikrometerkalibriervorrichtung, Einstellbereich 0 - 25 mm, verstellbarer Nullpunkt für alle Sensoren

Vakuumdurchführung
eddy/fB0/fB0/triax



Stecksystem für den Einsatz im Vakuum. Mit allen gängigen eddyNCDT-Produkten kompatibel

- Anwendung als Wanddurchführung
- Steckbare Ausführung
- Montage-Gewinde M9x0,5 / Länge 39 mm
- Wandstärke für die Montage maximal 22 mm
- Maximale Leckagerate $<10^{-8}$ mbar l/s

IF7001



- Einkanal USB/RS485 Konverter
- 2 x 0,14 mm², geschirmt, PVC, \varnothing 4 mm \pm 0,2 mm
- Betriebstemperatur: 0 ... 70 °C
- Schutzart: IP 20

PCx/8-M12 PVC

- Versorgungs- und Signalkabel
- M12 Buchse, gerade, 8-polig, geschirmt
- Standardkabel: 29011159 (3m)
- Optionale Kabel: 29011141 (5m), 29011058 (10 m)
- Nicht schleppkettentauglich
- PVC-Leitung: -25 °C ... +105 °C
- Buchse: -25 °C ... +90 °C

PC10/8-M12 TPE

- Versorgungs- und Signalkabel
- M12 Buchse, gerade, 8-polig, geschirmt
- 29011285 (10m)
- 29011059 (15m)
- Schleppkettentaugliche Leitung TPE
- Beweg: -25 °C ... +80 °C
- Fest: -40 °C ... +80 °C

PC5/8-M12/105 PUR

- Versorgungs- und Signalkabel
- M12 Buchse, gerade, 8-polig, geschirmt
- 29011506 (5m)
- Schleppkettentaugliche Leitung PUR
- Beweg: -20 °C ... +105 °C
- Fest: -40 °C ... +105 °C
- Schleppkette: -20 °C ... +60 °C

17.2 Modellbezeichnung Sensor

Eddy Sensor	Messbereich	S = Stecker C = Kabel integriert	Kabellänge [m]	A = mini B = normal C = groß	Option
ES - S	1 / 2 / 3 / 4 / 6 / 8 mm		2,0 / m	B	0 / □
S = geschirmt U = ungeschirmt		C = zylindrisch F = flach T = Klemmflansch	m = Stecker f = Buchse OE = offene Enden		0 = gerade 90 = ungerade

17.3 Modellbezeichnung Sensorkabel

EC = Eddy Kabel ECE = Eddy Kabel Verlängerung									
EC - 3,0 / m		A	90 / f	B	0 /	D3,9			
Kabellänge, nominal [m]	m = Stecker f = Buchse OE = offene Enden	A = mini B = normal C = groß	0 = gerade 90 = gewinkelt	A = mini B = normal C = groß	0 = gerade 90 = gewinkelt	D = Kabel-durchmesser, nominal [mm]			
				Seite Controller					
				Seite Sensor					

17.4 Stabilität gegenüber Störeinstrahlung

Wirbelstromsensoren von Micro-Epsilon zeichnen sich durch eine hohe Beständigkeit gegenüber Störeinstrahlung nach EN61000-4-6 (leitungsgebunden) und EN61000-4-3 (hochfrequente elektromagnetische Felder) aus. Typische Störquellen wie z. B. Netzanwendungen liegen weit von der Trägerfrequenz eines Oszillators entfernt.

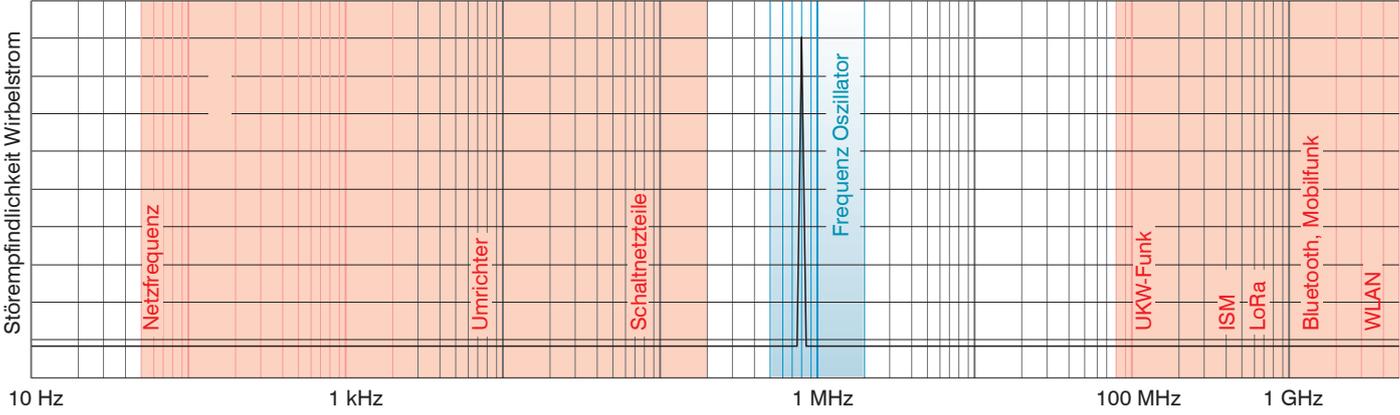


Abb. 17.1: Störimpfindlichkeit Wirbelstromsensoren

17.5 Softwarebeispiele für gängige Programmiersprachen

Die folgenden Beispiele lesen den Namen, die Seriennummer des DT3020 und die Beschreibung der Messwerte aus. Dann werden einige Messwerte aus dem DT3020 ausgelesen.

Python

➤ Kopieren Sie die beiden Dateien `MEDAQLib.dll` und `MEDAQLib.py` aus dem Unterverzeichnis `Snippets/Python` im `MEDAQLib-Installationsverzeichnis` in das gleiche Verzeichnis wie den Python-Quellcode.

```
#
# This is a very simple sample following MEDAQLib.pdf section 4 Using MEDAQLib
#
# Please adjust to your setup (interface card and sensor used)
#

from MEDAQLib import MEDAQLib, ME_SENSOR, ERR_CODE
import time

number_of_reads = 10;

# Tell MEDAQLib about sensor type to be used
MEDAQLib_object = MEDAQLib.CreateSensorInstByName ("MEBus")

# Tell MEDAQLib about interface to be used
MEDAQLib_object.SetParameterString("IP_Interface", "RS232")
MEDAQLib_object.SetParameterString("IP_Port", "COM4")
MEDAQLib_object.SetParameterInt("IP_SensorAddress", 126)
MEDAQLib_object.SetParameterInt("IP_Baudrate", 230400)

# Enable Logfile writing
# MEDAQLib_object.SetParameterInt("IP_EnableLogging", 1)

# Try to open communication to sensor via interface specified
MEDAQLib_object.OpenSensor()
if MEDAQLib_object.GetLastError() != ERR_CODE.ERR_NOERROR:
    raise RuntimeError("OpenSensor: " + MEDAQLib_object.GetError())

MEDAQLib_object.ExecSCmd("Get_ControllerInfo")
if MEDAQLib_object.GetLastError() != ERR_CODE.ERR_NOERROR:
    raise RuntimeError("Get_ControllerInfo: " + MEDAQLib_object.GetError())

controller_name = MEDAQLib_object.GetParameterString("SA_ControllerName")
serial_number = MEDAQLib_object.GetParameterString("SA_SerialNumber")
print(f"Controller Name: {controller_name}")
print(f"Controller Serial Number: {serial_number}")

MEDAQLib_object.ExecSCmd("Get_TransmittedDataInfo")
if MEDAQLib_object.GetLastError() != ERR_CODE.ERR_NOERROR:
    raise RuntimeError("Get_TransmittedDataInfo: " + MEDAQLib_object.GetError())

number_of_channels = MEDAQLib_object.GetParameterInt("IA_ValuesPerFrame")
```

```

if number_of_channels == 0:
    raise RuntimeError("No data channels available")

for i in range(1, number_of_channels+1):
    index = MEDAQLib_object.GetParameterInt("IA_Index"+str(i))
    raw_name = MEDAQLib_object.GetParameterString("IA_Raw_Name"+str(i))
    scaled_name = MEDAQLib_object.GetParameterString("IA_Scaled_Name"+str(i))
    raw_unit = MEDAQLib_object.GetParameterString("IA_Raw_Unit"+str(i))
    scaled_unit = MEDAQLib_object.GetParameterString("IA_Scaled_Unit"+str(i))
    raw_range_min = MEDAQLib_object.GetParameterDouble("IA_Raw_RangeMin"+str(i))
    scaled_range_min = MEDAQLib_object.GetParameterDouble("IA_Scaled_RangeMin"+str(i))
    raw_range_max = MEDAQLib_object.GetParameterDouble("IA_Raw_RangeMax"+str(i))
    scaled_range_max = MEDAQLib_object.GetParameterDouble("IA_Scaled_RangeMax"+str(i))
    print(f"{index}: {raw_name} [{raw_range_min} .. {raw_range_max} {raw_unit}], " \
          f"{scaled_name} in {scaled_unit} [{scaled_range_min} .. {scaled_range_max}]")

print(f"Read {number_of_reads} measurements from {number_of_channels} channels ...")
# If no error then try to acquire data
if MEDAQLib_object.GetLastError() == ERR_CODE.ERR_NOERROR:
    for num_read in range(number_of_reads):
        # Sleep for 10 ms
        time.sleep(0.01)
        # Ask sensor for new data
        MEDAQLib_object.ExecSCmd("Get_Measure")
        # Check whether there is enough data to read in
        currently_available = MEDAQLib_object.DataAvail()
        # Check if DataAvail causes an Error
        if (MEDAQLib_object.GetLastError() != ERR_CODE.ERR_NOERROR):
            print(MEDAQLib_object.GetError())
        # If data is available?
        if currently_available >= number_of_channels:
            # Transfer/Move data from MEDAQLib internal buffer to own buffer
            transfered_data = MEDAQLib_object.TransferData(currently_available)
            # Check if TransferData causes an error
            if MEDAQLib_object.GetLastError() == ERR_CODE.ERR_NOERROR:
                # contains original values form sensor
                raw_data = transfered_data[0]
                # contains scaled data values
                scaled_data = transfered_data[1]
                # get number of data values received,
                # should be equal to currently_available
                nr_values_transfered = transfered_data[2]
                # output raw and scaled value of very first measurement
                for j in range(0,nr_values_transfered,number_of_channels):
                    print(scaled_data[j:j+number_of_channels], sep=', ')

```

```
        # do your computation on data ....
    else:
        # Print TransferData error
        print(MEDAQLib_object.GetError())
else:
    # Print OpenSensor Error
    print(MEDAQLib_object.GetError())

# Closing down by closing interface and releasing sensor instance
MEDAQLib_object.CloseSensor()
MEDAQLib_object.ReleaseSensorInstance()
```

C#

► Kopieren Sie die beiden Dateien `MEDAQLib.dll` und `MEDAQLib.Net.dll` aus dem Unterverzeichnis `Release` im `MEDAQLib-Installationsverzeichnis` in das gleiche Verzeichnis wie den C# Code.

```
using System;
using System.Diagnostics;
using MicroEpsilon; // MEDAQLib

namespace C_Sharp_Example
{
    class Program
    {
        static ERR_CODE Error(string location, ref MEDAQLib sensor)
        {
            string errText = "";
            ERR_CODE err = sensor.GetError(ref errText);
            Console.WriteLine(location + " returned error: " + errText);
            Console.WriteLine("Demo failed, press any key ...");
            Console.ReadKey(true);
            return err;
        }

        static int sValsPerFrame = 0;

        static string StrWithIndex(string name, int index)
        {
            return name + index.ToString();
        }

        static ERR_CODE GetControllerInfo(ref MEDAQLib sensor)
        {
            string controllerName = "", controllerSerialNumber = "";

            if (sensor.ExecSCmd("Get_ControllerInfo") != ERR_CODE.ERR_NOERROR)
                return Error("Get_ControllerInfo", ref sensor);

            sensor.GetParameterString("SA_ControllerName", ref controllerName);
            sensor.GetParameterString("SA_SerialNumber", ref controllerSerialNumber);

            Console.WriteLine("Controller Name: {0}", controllerName);
            Console.WriteLine("Controller Serial Number: {0}", controllerSerialNumber);

            return ERR_CODE.ERR_NOERROR;
        }

        static ERR_CODE GetTransmittedDataInfo(ref MEDAQLib sensor)
        {
            int maxValsPerFrame = 0, maxOutputIndex = 0;

            if (sensor.ExecSCmdGetInt("Get_TransmittedDataInfo", "IA_ValuesPerFrame",
```

```

        ref sValsPerFrame) != ERR_CODE.ERR_NOERROR)
    return Error("Get_TransmittedDataInfo", ref sensor);

sensor.GetParameterInt("IA_MaxValuesPerFrame", ref maxValsPerFrame);
sensor.GetParameterInt("IA_MaxOutputIndex", ref maxOutputIndex);
Console.WriteLine("Sensor transmits {0} of {1} possible values," +
    "maximum output index is {2}",
    sValsPerFrame, maxValsPerFrame, maxOutputIndex);

for (int i = 0; i < sValsPerFrame; i++)
{
    int index = 0;
    double rawRangeMin = 0.0, rawRangeMax = 0.0;
    double scaledRangeMin = 0.0, scaledRangeMax = 0.0;
    string rawName = "", scaledName = "", rawUnit = "", scaledUnit = "";
    sensor.GetParameterString(
        StrWithIndex("IA_Raw_Name", i + 1), ref rawName);
    sensor.GetParameterString(
        StrWithIndex("IA_Scaled_Name", i + 1), ref scaledName);
    sensor.GetParameterString(
        StrWithIndex("IA_Raw_Unit", i + 1), ref rawUnit);
    sensor.GetParameterString(
        StrWithIndex("IA_Scaled_Unit", i + 1), ref scaledUnit);
    sensor.GetParameterInt(
        StrWithIndex("IA_Index", i + 1), ref index);
    sensor.GetParameterDouble(
        StrWithIndex("IA_Raw_RangeMin", i + 1), ref rawRangeMin);
    sensor.GetParameterDouble(
        StrWithIndex("IA_Scaled_RangeMin", i + 1), ref scaledRangeMin);
    sensor.GetParameterDouble(
        StrWithIndex("IA_Raw_RangeMax", i + 1), ref rawRangeMax);
    sensor.GetParameterDouble(
        StrWithIndex("IA_Scaled_RangeMax", i + 1), ref scaledRangeMax);
    Console.WriteLine(
        " {0,2}: {1} [{2} .. {3} {4}], {5} in {8} [" +
        "{6} .. {7}" +
        "]",
        index, rawName, rawRangeMin, rawRangeMax, rawUnit, scaledName,
        scaledRangeMin, scaledRangeMax, scaledUnit
    );
    Console.WriteLine(" {0,2}: {1} [{2} .. {3} {4}], {5} in {8} " +
        "[{6} .. {7}]", index, rawName, rawRangeMin, rawRangeMax, rawUnit,
        scaledName, scaledRangeMin, scaledRangeMax, scaledUnit);
}

return ERR_CODE.ERR_NOERROR;
}

```

```

static ERR_CODE TransferData(ref MEDAQLib sensor)
{
    Console.WriteLine("Transfer data ...");

    while (!Console.KeyAvailable)
    {
        System.Threading.Thread.Sleep(10);
        sensor.ExecSCmd("Get_Measure");

        int avail = 0;
        if (sensor.DataAvail(ref avail) != ERR_CODE.ERR_NOERROR)
            return Error("DataAvail", ref sensor);

        int[] rawData = new int[avail];
        double[] scaledData = new double[avail];
        int read = 0;
        if (sensor.TransferData(rawData, scaledData, avail, ref read)
            != ERR_CODE.ERR_NOERROR)
            return Error("TransferData", ref sensor);

        int num_values = read/sValsPerFrame;
        for (int i = 0; i < num_values; i++)
        {
            Console.Write("{0:F3}", scaledData[i*sValsPerFrame]);
            for (int j = 1; j < sValsPerFrame; j++)
            {
                Console.Write(", {0:F3}", scaledData[i*sValsPerFrame+j]);
            }
            Console.WriteLine("");
        }
    }
    Console.ReadKey(true);
    Console.WriteLine("");

    return ERR_CODE.ERR_NOERROR;
}

static void Main(string[] args)
{
    Console.WriteLine("Start Demo...");

    MEDAQLib sensor = new MEDAQLib("ME-Bus");
    sensor.SetParameterString("IP_Interface", "RS232");
    sensor.SetParameterString("IP_Port", "COM4");
    sensor.SetParameterInt("IP_Baudrate", 230400);
    sensor.SetParameterInt("IP_SensorAddress", 126);
}

```

```
// Enables logging of additional debugging information to TXT file
//sensor.SetParameterInt("IP_EnableLogging", 1);

if (sensor.OpenSensor() != ERR_CODE.ERR_NOERROR)
{
    Error("OpenSensor", ref sensor);
    return;
}

if (GetControllerInfo(ref sensor) != ERR_CODE.ERR_NOERROR)
    return;

if (GetTransmittedDataInfo(ref sensor) != ERR_CODE.ERR_NOERROR)
    return;

if (sValsPerFrame == 0)
{
    Console.WriteLine("No data channels available");
    Console.WriteLine("Demo failed, press any key ...");
    Console.ReadKey(true);
    return;
}

if (TransferData(ref sensor) != ERR_CODE.ERR_NOERROR)
    return;

Console.WriteLine("Demo successfully finished, press any key ...");
Console.ReadKey(true);
}
}
}
```

MATLAB

➡ Kopieren Sie die beiden Dateien `MEDAQLib.h` und `Release-x64\MEDAQLib.dll` aus dem `MEDAQLib-Installationsverzeichnis` in das gleiche Verzeichnis wie das `MATLAB-Skript`.

```
%DT3020_READ Example for reading one measurement value from DT3020
clear;
medaqlib_install_dir = 'C:\Program Files (x86)\MEDAQLib';
max_str_length = 32;
max_err_length = uint32(1024); % Reserved maximum length for error messages

number_of_reads = 10; % Number read requests to the sensor

%% Load MEDAQLib
if ~isfile('MEDAQLib.dll')
    copyfile(fullfile(medaqlib_install_dir, 'Release-x64', 'MEDAQLib.dll'), '.');
end
if ~isfile('MEDAQLib.h')
    copyfile(fullfile(medaqlib_install_dir, 'MEDAQLib.h'), '.');
end
if ~libisloaded('medaqlib')
    [notfound, warnings] = loadlibrary('MEDAQLib', 'MEDAQLib.h', 'alias', 'medaqlib');
end

try

    %% Tell MEDAQLib about sensor type to be used.
    h_sensor = uint32(calllib('medaqlib', 'CreateSensorInstByName', 'MEbus'));

    %% Tell MEDAQLib about interface to be used
    calllib('medaqlib', 'SetParameterString', h_sensor, 'IP_Interface', 'RS232');
    calllib('medaqlib', 'SetParameterString', h_sensor, 'IP_Port', 'COM4');
    calllib('medaqlib', 'SetParameterInt', h_sensor, 'IP_SensorAddress', 126);
    calllib('medaqlib', 'SetParameterInt', h_sensor, 'IP_Baudrate', 230400);

    %% Enable Logfile writing
    calllib('medaqlib', 'SetParameterInt', h_sensor, 'IP_EnableLogging', 1);

    %% Try to open communication to sensor via interface specified
    err = calllib('medaqlib', 'OpenSensor', h_sensor);
    if ~strcmp(err, 'ERR_NOERROR')
        error('Unable to open Sensor %s', err);
    end
catch ME
    unloadlibrary('medaqlib');
    rethrow(ME);
end

try

    %% Read information about connected controller
```

```

err = calllib('medaqlib', 'ExecSCmd', h_sensor, 'Get_ControllerInfo');
assert(strcmp(err, 'ERR_NOERROR'), 'medaqlib:ExecSCmd', 'Get_ControllerInfo');

% Display controller name
[~, param_name, controller_name, ~] = calllib('medaqlib', ...
    'GetParameterString', h_sensor, 'SA_ControllerName', ...
    blanks(max_str_length), libpointer('uint32Ptr', max_str_length));
fprintf('%s = %s\n', param_name, controller_name);

% Display controller serial number
[~, param_name, controller_serial_number, ~] = calllib('medaqlib', ...
    'GetParameterString', h_sensor, 'SA_SerialNumber', ...
    blanks(max_str_length), libpointer('uint32Ptr', max_str_length));
fprintf('%s = %s\n', param_name, controller_serial_number);

%% Read information about the transmitted data
err = calllib('medaqlib', 'ExecSCmd', h_sensor, 'Get_TransmittedDataInfo');
assert(strcmp(err, 'ERR_NOERROR'), 'medaqlib:ExecSCmd', 'Get_TransmittedDataInfo');
[~, ~, channel_count] = calllib('medaqlib', 'GetParameterInt', h_sensor, ...
    'IA_ValuesPerFrame', libpointer('int32Ptr', 0));

disp("Read "+string(number_of_reads)+" measurements from "+ ...
    channel_count+" channels ...");
if channel_count == 0
    error('No data channels available');
end

channel_names = cell(1, channel_count);
for k = 1:channel_count
    [~, ~, index] = calllib('medaqlib', 'GetParameterInt', h_sensor, ...
        sprintf('IA_Index%d', k), libpointer('int32Ptr', 0));
    [~, ~, scaled_name, ~] = calllib('medaqlib', 'GetParameterString', ...
        h_sensor, sprintf('IA_Scaled_Name%d', k), blanks(max_str_length), ...
        libpointer('uint32Ptr', max_str_length));
    [~, ~, scaled_unit, len] = calllib('medaqlib', 'GetParameterBinary', ...
        h_sensor, sprintf('IA_Scaled_Unit%d', k), ...
        zeros(1, max_str_length, 'uint8'), ...
        libpointer('uint32Ptr', max_str_length));
    if len > 0
        scaled_unit = char(scaled_unit(1:len));
    else
        scaled_unit = '';
    end
    [~, ~, scaled_range_min] = calllib('medaqlib', 'GetParameterDouble', ...
        h_sensor, sprintf('IA_Scaled_RangeMin%d', k), libpointer('doublePtr', 0));
    [~, ~, scaled_range_max] = calllib('medaqlib', 'GetParameterDouble', ...
        h_sensor, sprintf('IA_Scaled_RangeMax%d', k), libpointer('doublePtr', 0));
    disp(string(index)+": "+scaled_name+" ["+string(scaled_range_min) ...

```

```

        +" .. "+string(scaled_range_max)+"]");
        channel_names{k} = strip(strrep(scaled_name, '(scaled)', ''), 'both');
    end
    clear str;
    disp(strjoin(channel_names, ', '));

%% Read measurement value from sensor
for i = 1:number_of_reads
    % Ask sensor for new data
    err = calllib('medaqlib', 'ExecSCmd', h_sensor, 'Get_Measure');
    assert(strcmp(err, 'ERR_NOERROR'), 'medaqlib:ExecSCmd', 'Get_Measure');

    % Check whether there is enough data to read in
    [~, currently_available] = calllib('medaqlib', 'DataAvail', h_sensor, ...
        libpointer('int32Ptr', 1));

    % If data is available?
    if currently_available >= channel_count
        % Transfer/Move data from MEDAQLib's internal buffer to own buffer
        raw_data = libpointer('int32Ptr', zeros(1, currently_available));
        scaled_data = libpointer('doublePtr', zeros(1, currently_available));

        [err, raw_data, scaled_data, num_read] = calllib('medaqlib', ...
            'TransferData', h_sensor, raw_data, scaled_data, ...
            currently_available, libpointer('int32Ptr', 1));
        assert(strcmp(err, 'ERR_NOERROR'), 'medaqlib:TransferData', '');

        data_read = reshape(scaled_data, channel_count, []).';
        for k = 1:size(data_read, 1)
            disp(strjoin(string(data_read(k, :)), ', '));
        end
    else
        error('No data available');
    end
end
calllib('medaqlib', 'CloseSensor', h_sensor);
calllib('medaqlib', 'ReleaseSensorInstance', h_sensor);
unloadlibrary('medaqlib');
catch ME
    s = split(ME.identifier, ':');
    if strcmp(s{1}, 'medaqlib')
        [~, error_txt] = calllib('medaqlib', 'GetError', h_sensor, ...
            blanks(max_err_length), max_err_length);
        if isempty(error_txt)
            error('%s: %s', ME.identifier, ME.message);
        else
            error('%s: %s MEDAQLib "%s"', ME.identifier, ME.message, error_txt);
        end
    end
end

```

```
end
calllib('medaqlib', 'CloseSensor', h_sensor);
calllib('medaqlib', 'ReleaseSensorInstance', h_sensor);
unloadlibrary('medaqlib');
rethrow(ME);
end
```



MICRO-EPSILON MESSTECHNIK GmbH & Co. KG
Königbacher Str. 15 94496 Ortenburg / Deutschland
Tel. +49 (0) 8542 / 168-0 Fax +49 (0) 8542 / 168-90
info@micro-epsilon.de <https://www.micro-epsilon.de>
Your local contact: www.micro-epsilon.com/contact/worldwide/

X9750495-A01
© MICRO-EPSILON MESSTECHNIK