



Bedienungsanleitung

boreCONTROL SDK 1.5

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

94496 Ortenburg / Deutschland

Tel. +49/8542/168-0
Fax +49/8542/168-90
e-mail info@micro-epsilon.de
www.micro-epsilon.de

Inhalt

1.	Einleitung.....	5
1.1	Installation und Vorbereitung für den Betrieb.....	6
1.2	Grundlegende Begriffsdefinitionen.....	6
1.3	Messprinzip von boreCONTROL.....	7
2.	Beschreibung einzelner Abläufe.....	8
2.1	Verbinden mit den boreCONTROL-Komponenten.....	8
2.2	Einstellungen der Parameter für boreCONTROL.....	8
2.3	Sensorlanzen mit optischer Temperaturkompensation und Betriebsmodi.....	9
2.4	Dunkelabgleich von boreCONTROL.....	10
2.5	Aufnahme und Auswertung von Messdaten.....	11
2.6	Winkel-Kalibrierung von boreCONTROL.....	11
2.7	Offset-Kalibrierung von boreCONTROL.....	12
2.8	Qualitätskontrolle mittels Videodatenerfassung.....	13
2.9	Filterung von Daten.....	14
2.9.1	Sektor-Filterung.....	14
2.9.2	Median-Filterung.....	15
2.9.3	Mittelwert-Filterung.....	15
3.	Kurzbeschreibung der einzelnen Funktionen.....	16
3.1	Funktionen zur Verbindungsherstellung.....	16
3.2	Funktionen zur Ansteuerung der Rotationseinheit.....	16
3.3	Funktionen zur Interpretation der Z-Achsen-Information.....	17
3.4	Funktionen zum Setzen der Scan-Parameter.....	17
3.5	Funktionen zur Videodatenerfassung.....	17
3.6	Funktionen zur Kalibrierung.....	18
3.7	Funktionen zur Offset-Kalibrierung.....	18
3.8	Funktionen zur Datenakquisition.....	18
3.9	Funktionen zum Lesen und Schreiben von Daten.....	19
3.10	Funktionen zur Unterstützung spezieller Datentypen.....	19
3.11	Funktionen zur Datenfilterung.....	19
3.12	Funktionen zur Auswertung.....	19
3.13	Funktionen zum Datenaustausch.....	20
3.14	Funktionen zur Referenz-Kalibrierung.....	20
3.15	Funktionen zur Bearbeitung von Referenz-Kalibrierungen.....	21
3.16	Funktionen zur Bearbeitung von Kalibrierstapeln.....	21
4.	Beschreibung der Datenstrukturen.....	22
4.1	Verbindungsaufbau – bore_connect_t und Aufzählungstypen.....	22
4.2	Akquisitionsmodi – bore_refcalib_mode_t.....	23
4.3	Sensordaten - bore_data_t und bore_meas_t.....	23
4.4	Auswertungsinformation – bore_eval_t.....	24
4.5	Offset-Kalibrier-Information – bore_offsetcal_t.....	26
4.6	Kalibrierstack-Information – bore_calstack_info_t und bore_calstack_eval_t.....	27
5.	Beschreibung der Fehlercodes.....	29
5.1	Allgemeine Fehler.....	29
5.2	Fehler Rotationseinheit.....	29
5.3	Fehler Sensor-Controller.....	29
5.4	Fehler bei der Auswertung.....	29
5.5	Fehler bei der Offset-Kalibrierung.....	29
5.6	Fehler bei der Referenz-Kalibrierung.....	29
5.7	Fehler beim Datenexport und Datenimport.....	30
5.8	Fehler beim Verarbeiten von Kalibrierstapeln.....	30
6.	SDK und Beispielapplikationen.....	31
6.1	boreCONTROLDemo.....	31

Einleitung

6.2	boreCONTROLExtDemo.....	31
6.2.1	Beispielaufufe boreCONTROLExtDemo.....	32
6.3	boreCONTROLOffline.....	33
6.4	boreCONTROLRecalib.....	35
7.	Hinweise zum Umgang mit boreCONTROL.....	37

1. Einleitung

Dieses Handbuch beschreibt die Verwendung des boreCONTROL SDK (Software Development Kit) und der darin enthaltenen boreCONTROL.dll (Dynamic Link Library) beim Einsatz in kundenseitig erstellten Applikationen.

Die boreCONTROL.dll erlaubt die Ansteuerung des Messsystems boreCONTROL, welches aus Sensorlanze und Controller IFC24x1 besteht und dessen Komponenten für Anwendungen im industriellen Umfeld ausgelegt sind. Mithilfe der DLL lassen sich Sensorparameter einstellen, Sensordaten einlesen sowie grundlegende Messwerte berechnen.

boreCONTROL dient zur Erfassung von 2D-Scandaten, die typischerweise in Bohrlöchern generiert werden. Die Profildaten der Oberfläche werden von der Sensorlanze unter Nutzung des konfokalen Messprinzips erfasst. Die angeschlossene Rotationseinheit mit Controller BCC2410 rotiert die Sensorlanze und sorgt so für die ringförmige Abtastung des zu vermessenden Objekts. Durch Verfahrnung der Rotationseinheit in Richtung der Lanze werden so 3D-Scans im Messobjekt möglich.

Der Controller IFC24x1 des Messsystems boreCONTROL sammelt auf Anforderung Positions-, Intensitäts- und Entfernungsdaten und gibt diese über eine Ethernet-Schnittstelle an den steuernden und auswertenden PC weiter. Die boreCONTROL.dll übernimmt die Ansteuerung der Komponenten von boreCONTROL und stellt die Daten für die kundenseitige Applikation zur Verfügung. Einfache Auswertungen sind ebenfalls im Funktionsumfang enthalten.

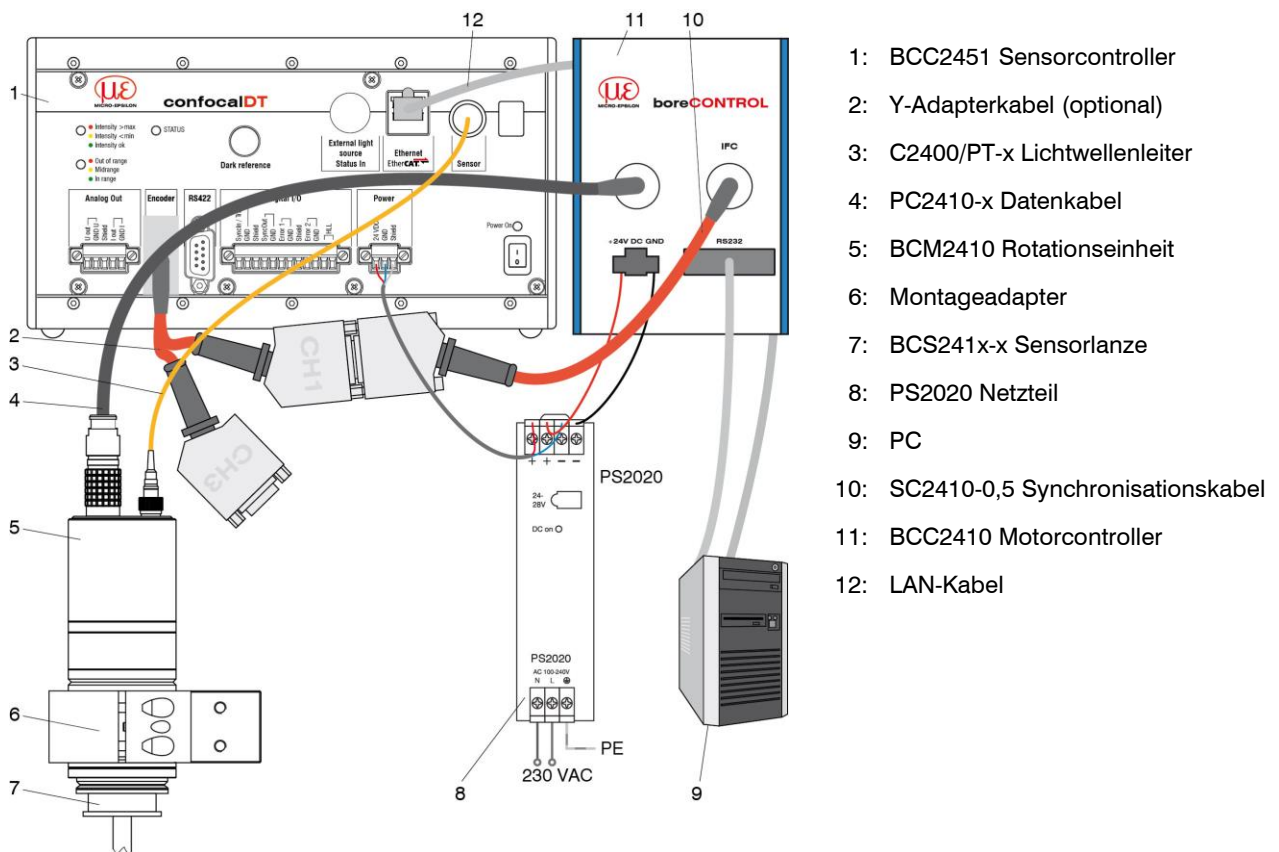


Abbildung 1: Hardwarekomponenten boreCONTROL mit Anschluss an den kundenseitigen PC

Zur Realisierung unterschiedlicher Messbereiche und Auflösungen stehen verschiedene Sensortypen zur Verfügung. Diese Sensorlanzen können bei Bedarf vom Benutzer ausgetauscht werden.

Um die zusammen mit boreCONTROL gelieferte DLL mit möglichst vielen verschiedenen Entwicklungsumgebungen und Compilern nutzen zu können, wurde die DLL-Schnittstelle mit reinen C-Funktionen der „stdcall“ Aufrufkonvention realisiert. Dadurch kann die DLL auch unter C++, Delphi, Visual Basic, Python oder anderen Programmiersprachen genutzt werden (Bedingung dafür ist die Kompatibilität der verwendeten Datentypen).

In dieser Dokumentation wird nur die Einbindung der DLL in C beschrieben, die Einbindung in C++ oder in andere Programmiersprachen kann aus dieser Dokumentation abgeleitet werden.

1.1 Installation und Vorbereitung für den Betrieb

Informationen zum Aufbau der Hardwarekomponenten von boreCONTROL entnehmen Sie bitte der „Bedienungsanleitung boreCONTROL“. Führen Sie den Aufbau der Hardwarekomponenten gemäß der Anleitung durch.

Die Kommunikation mit dem Controller findet über Ethernet statt. Die IP-Adresse der Controller ist auf 169.254.168.150 vorkonfiguriert, kann aber bei Bedarf über die Weboberfläche geändert werden. Diese Adresse ist beim Öffnen der Verbindung anzugeben. Die Installation von Gerätetreibern ist nicht erforderlich.

Zur Ausführung der mitgelieferten Software ist ggf. die Installation der vcredist_x86.exe von Microsoft notwendig. Dieses finden Sie auf der CD unter „Support\vcredist_x86“.

Damit ist die Hardware für die Ansteuerung via boreCONTROL.dll bereit.

1.2 Grundlegende Begriffsdefinitionen

Die Entfernungsbestimmung erfolgt durch Spektralanalyse des vom Messobjekt reflektierten Lichts. Nachfolgend ist ein typisches Spektrum dargestellt:

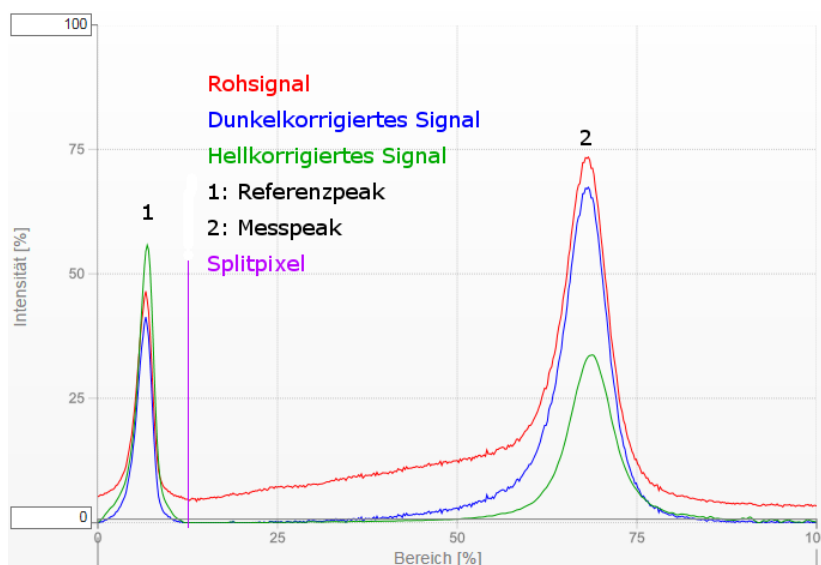


Abbildung 2: Vom Controller erfasstes Videosignal, Beschreibung siehe unten

Die folgenden Begriffe werden im Handbuch verwendet:

- **Rohsignal:** Das vom Controller erfasste Spektrum einschließlich der Einflüsse der optischen Komponenten
- **Dunkelkorrigiertes Signal:** Das vom Controller erfasste Spektrum, bereinigt um die Einflüsse der optischen Komponenten
- **Hellkorrigiertes Signal:** Das vom Controller erfasste Spektrum, korrigiert um das Spektrum der integrierten Lichtquelle
- **Splitpixel:** Für jede Sensorlanze festgelegte Position, die den Referenzpeak (links) vom Messpeak (rechts) trennt.
- **Messpeak:** Vom Messobjekt reflektiertes Licht, Aus Dem Messpeak werden die Helligkeit und der Messabstand in Sensoreinheiten generiert. Der Messpeak befindet sich rechts vom Splitpixel.
- **Referenzpeak:** Im Controller abgebildetes Sensor-Signal, das aus einer auf der Sensorlanze aufgetragenen Reflexionsschicht stammt, damit immer sichtbar ist und zur Temperaturkompensation herangezogen wird. Der Referenzpeak wird wie der Messpeak ausgewertet und zusammen mit diesem über eine Kalibriertabelle zu einem Messabstand in μm verrechnet. Der Referenzpeak befindet sich links vom Splitpixel.
- **Intensität:** Prozentualer Helligkeitswert, zwischen 0% (dunkel) und 100% (hell).
- **Datenpunkt/Messpunkt:** Ein einzelnes Tupel aus Datenwerten (u.a. Winkelkoordinate, Intensität, Abstand, Encoderzähler) zu einem einzelnen Zeitpunkt.
- **Profil:** Ein Profil besteht aus mehreren Messpunkten. Jeder dieser Punkte ist durch seinen Winkel-, Abstands- und Intensitätswert definiert. Ein Profil entspricht einer Umdrehung des Bohrlochensors.

- **Messwert:** Aus einem einzelnen Profil wird durch Kreispassung neue Informationen über das vermessene Objekt, insbesondere der Passkreisradius und das Passkreiszentrum, generiert.
- **Punktrate:** Bezeichnet die Anzahl der Datenpunkte, die pro Sekunde erfasst werden. Aus der Punktrate und der Umdrehungsgeschwindigkeit des Bohrlochensors ergibt sich die Anzahl der pro Profil erfassten Datenpunkte. Die Punktrate wird auch als Scanrate bezeichnet.
- **Offset:** Zum Messabstand addierter Zahlenwert, um beim Wechsel der Sensorlanze entstehende minimale Abweichungen des Messwerts auszugleichen. Der Offset muss nach einem Wechsel der Sensorlanze mittels Kalibrierung mit einem Kalibriertarget neu eingestellt werden.

1.3 Messprinzip von boreCONTROL

Mithilfe der Rotationseinheit wird der Bohrlochsensor im zu vermessenden Objekt mit der vorgegebenen Umdrehungsgeschwindigkeit rotiert. Der Sensor erfasst Abstands- und Intensitätsdaten und kombiniert diese mit der aktuellen Winkelposition und, falls vorgesehen, der Eindringtiefe in das Messobjekt.

Eventuelle Temperaturschwankungen während der Messung werden über eine zusätzliche Reflexionsschicht (Referenzpeak) am Strahlaustritt der Sensorlanze erfasst und von der Software unter Verwendung von mitgelieferten Kalibriertabellen kompensiert.

Bei Inbetriebnahme und nach jedem Lanzenwechsel ist einmalig eine Offset-Kalibrierung mittels eines Kalibrierrings vorzunehmen. Aus der gemessenen Entfernung, zuzüglich des mittels Kalibrierung berechneten Offsets, ergibt sich der absolute Entfernungswert des Datenpunkts.

Die vom Sensor gelieferten Datenpunkte können aufgesammelt werden, bis eine volle Umdrehung komplettiert ist und anschließend einer Kreispassung zugeführt werden.

2. Beschreibung einzelner Abläufe

In diesem Kapitel werden die elementaren Schritte zur Ansteuerung der boreCONTROL Komponenten beschrieben und die Parametereinstellung für den Scanvorgang erläutert.

Zur korrekten Funktion des Systems sind unterschiedliche Kalibrierschritte erforderlich. Die als Datei verfügbaren Kalibriertabellen werden durch MICRO-EPSILON bereitgestellt und sorgen für die Linearisierung der gemessenen Distanzwerte für die gelieferten Sensorlanzen.

Bei der Erstinbetriebnahme und nach jedem Wechsel der Sensorlanze sollte ein Dunkelabgleich durchgeführt, die zur Linearisierung verwendete, zugehörige Kalibriertabelle ausgewählt und außerdem eine „Offset-Kalibrierung“ durchgeführt werden.

Die dazu notwendigen Schritte werden in diesem Kapitel erläutert. Dabei wird davon ausgegangen, dass der Aufbau und Anschluss der boreCONTROL-Komponenten bereits durchgeführt wurde.

Für detailliertere Information wird auf die nachfolgenden Kapitel mit Funktionsübersicht, Datenstruktur-Beschreibung, Aufstellung der Fehlercodes und Erläuterung der mitgelieferten Demonstrationsprogramme verwiesen. Außerdem existiert ein Anzeigeprogramm für die Dokumentationsübersicht, das im Lieferumfang als Documentation.exe enthalten ist und unter anderem auch dieses Dokument und den Einstieg in die automatisch generierte SDK Schnittstellenreferenz enthält.

2.1 Verbinden mit den boreCONTROL-Komponenten

Um die Funktionalität des boreCONTROL SDK einsetzen zu können, ist es erforderlich, eine Verbindung zu den Komponenten aufzubauen. Dazu wird die Funktion „boreConnect“ aufgerufen, die die Datenstruktur bore_t initialisiert:

```
bore_t bore;  
bore_result_t result;  
  
result = boreConnect(&bore, NULL);
```

Bei vom Auslieferungsstand abweichender Konfiguration (z.B. geänderte IP-Adresse des Controllers oder COM-Port Nummer der Rotationseinheit) muss der zweite Parameter manuell befüllt werden. Für den Auslieferungszustand ist dies:

```
bore_connect_t param;  
  
memset(&param, 0, sizeof(param));  
param.controller = BORE_CONTROLLER_IFC2461;  
strcpy(param.tcpip_address, "169.254.168.150"); /* Voreinstellung */  
param.tcpip_port = 23; /* Voreinstellung */  
param.rotation_unit = BORE_ROTATION_BCC2410;  
param.rotation_com_port = 1; /* Voreinstellung */
```

Der Aufruf von „boreConnect“ ändert sich dann wie folgt:

```
result = boreConnect(&bore, &param);
```

Die Komponenten sind erfolgreich initialisiert, falls der Rückgabewert BORE_OK ist.

Um einen gleichmäßigen Temperaturverlauf in der Rotationseinheit zu gewährleisten, ist es empfehlenswert, diese entweder mit konstanter Geschwindigkeit ständig rotieren zu lassen oder immer nur kurzzeitig zur Messung mittels „boreRotationStart“ und „boreRotationStop“ ein- und wieder auszuschalten, um eine Erwärmung der Komponenten zu verhindern.

```
boreRotationSetSpeed(&bore, 2.0); /* Setzen der Rotationsgeschwindigkeit auf 2 Hz */  
boreRotationStart(&bore);
```

Um die Verbindung zu den Komponenten zu trennen wird die Funktion „boreDisconnect“ aufgerufen:

```
boreDisconnect(&bore);
```

2.2 Einstellungen der Parameter für boreCONTROL

Benutzen Sie die folgenden Parameter, um den Scanprozess geeignet einzustellen:

- **Punktrate [Hz]:** Übergeben Sie der Funktion „boreSetScanrate“ die gewünschte Punktrate für die Datenerfassung. Beachten Sie, dass die Anzahl der Datenpunkte für einen kompletten Kreisscan

(Profil) zusätzlich von der Umdrehungsgeschwindigkeit des Bohrlochensors abhängt. Mit der Punktrate wird zugleich die maximale Belichtungszeit festgelegt, die $1000000 / \text{Punktrate } \mu\text{s}$ beträgt. Eine Erhöhung der Punktrate führt zu einer kürzeren maximalen Belichtungszeit. Die bei älteren Controllern / Firmwareständen vor 007.119.148.02c verfügbaren Punktraten sind beschränkt auf die diskreten Werte 100, 200, 300, 1000, 2500, 5000 und 10000 Hz. Neuere IFC2461 Controller erlauben eine freiere Wahl der Punktrate zwischen 100 Hz und 25000 Hz in Schritten von 100 Hz.

- **Belichtungszeit [μs]:** Mithilfe der Funktion „boreSetShutterTime“ kann die Intensität des Signals eingestellt werden. Je nach Reflexionsvermögen des Messobjekts ist der Wert so zu wählen, dass die vom Controller erfasste Helligkeit der Datenpunkte zwischen 10% und 90% liegt. Gemessene Intensitätswerte unter 10% weisen auf ein schwach reflektierendes Messobjekt hin. Reduzieren Sie dann gegebenenfalls die Punktrate, um längere Belichtungszeiten verwenden zu können und damit auch höhere Intensitätswerte zu erzielen. Einzelne Messpunkte mit Intensitäten außerhalb des empfohlenen Intervalls stellen kein Problem dar.
Zur Messung auf Oberflächen mit stark variierender Intensität steht ein Akquisitionsmodus mit vom Controller automatisch eingestellter Belichtungszeit zur Verfügung.
- **Rotationsgeschwindigkeit [Hz]:** Mit der Funktion „boreRotationSetSpeed“ kann die Rotationsgeschwindigkeit des Sensors im Intervall von 0,1 Hz bis 10 Hz eingestellt werden. Höhere Rotationsgeschwindigkeiten führen bei gleicher Punktrate zur Mittelung des Abstands und der Intensität über eine größere Strecke hinweg. Um eine optimale Genauigkeit der Messung zu erzielen, sollte eine besonders schnelle oder langsame Rotation vermieden werden, sofern dies die Nebenbedingungen bei der Messung zulassen.
- **Auswertungs-Schwellwerte [%]:** Mit den Funktionen „boreEvalSetIntensityThres“ und „boreEvalSetIntensityUpperThres“ werden die Helligkeitsschwellwerte für gültige Datenpunkte gesetzt, da für sehr dunkle und helle Messpunkte der Messabstand nur ungenau bestimmt werden kann. Datenpunkte mit einer Intensität unter bzw. über dem angegebenen Schwellwert (Richtwerte 10% und 90%) werden für die Kreispassung sowie die Berechnung der Messwerte ignoriert. Messpunkte mit Intensität kleiner als 5% sollten generell nicht verwendet werden.

2.3 Sensorlanzen mit optischer Temperaturkompensation und Betriebsmodi

Aktuelle Sensorlanzen verfügen über eine optische Kompensation von temperaturbedingten Messfehlern. Dabei wird durch eine Beschichtung der Sensorlanze ein zusätzlicher, als Referenzpeak bezeichneter Peak außerhalb des Messbereichs generiert, dessen Position nur durch die Temperatur verändert wird. Dieser Peak wird intern von der Software zur Korrektur des Messwerts herangezogen.

Die DII führt die Linearisierung und Temperaturkompensation anhand einer zuvor zur Verfügung gestellten Kalibriertabelle durch. Diese wird durch den Aufruf von „boreRefCalibLoad“ geladen.

Für die Temperaturkompensation stehen vier verschiedene Modi zur Verfügung, die über die Funktion „boreRefCalibSetMode“ ausgewählt werden können.

- Der normale Messbetrieb wird im Modus BORE_REFCALIB_ONREQUEST durchgeführt. Dabei muss vor der eigentlichen Messung eine Erfassung des Referenzpeaks durch Aufruf der Funktion „boreRefCalibSingle“ durchgeführt werden. Dieser Referenzpeak wird zur Temperaturkompensation der nachfolgend akquirierten Messwerte verwendet. Die Erfassung des Referenzpeaks sollte in regelmäßigen Zeitabständen, je nach erwarteter Temperaturänderung innerhalb einer Minute oder auch in signifikant längeren Zeitintervallen, wiederholt werden.
Um eine optimale Helligkeit für den Referenzpeak zu erhalten, können Punktrate und Belichtungszeit für die Referenzpeakerafassung separat über die Funktionen „boreRefCalibSetScanrate“ und „boreRefCalibSetShutterTime“ eingestellt werden. Die für das Messobjekt eingestellte Belichtungszeit beeinflusst damit die Temperaturkompensation nicht, daher können so auch sehr gut oder sehr schwach reflektierende Objekte vermessen werden. Die empfohlenen Werte zur Erfassung des Referenzpeaks von $400 \mu\text{s}$ Belichtungszeit bei 2500 Hz Punktrate entsprechen der Konfiguration bei der Werkskalibrierung.
- Für Messobjekte mit stark schwankender Reflektivität, beispielsweise Bohrlöcher in zusammengesetzten Platten aus CFK und Metall, lässt sich keine Einstellung für die Belichtungszeit finden, die sowohl im gut als auch im schlecht reflektierenden Teil befriedigende Ergebnisse liefert. Zu diesem Zweck existieren sowohl ein Dual-Shutter-Modus (BORE_REFCALIB_ONREQ2AUTO), der zwischen zwei vorgegebenen Belichtungszeiten automatisch hin und her wechselt, als auch ein Auto-Shutter-Modus (BORE_REFCALIB_ONREQAUTOSH), der die Belichtungszeit im Rahmen der vorgegebenen Punktrate automatisch nachregelt. Diese beiden Automatik-Modi funktionieren

analog zum oben beschriebenen Modus BORE_REFCALIB_ONREQUEST: Die Information zur Temperaturkompensation wird also vor der Messung separat erfasst.

Während die Belichtungszeit im Automatik-Modus nicht notwendigerweise gesetzt werden muss, sind für den Dual-Shutter-Modus zwei deutlich unterschiedliche Belichtungszeiten mittels der Funktion „boreSetShutterTimes“ zu setzen. Die für den Messpunkt verwendete Belichtungszeit kann auf der Datenstruktur bore_meas_t im Feld „shutter“ abgegriffen werden. Für Rückschlüsse auf die Reflektivität des Messobjekts ist es empfehlenswert, den Intensitätswert durch die vom Controller eingestellte Belichtungszeit zu teilen.

- Für normal reflektierende Messobjekte existiert der Modus BORE_REFCALIB_ALWAYS, bei dem der Referenzpeak bei jedem einzelnen Messpunkt erfasst wird. Eine separate Erfassung vor der Messung entfällt damit. Allerdings ist dabei sicherzustellen, dass die Referenzpeak-Intensität bei der eingestellten Belichtungszeit in einem verwendbaren Bereich von etwa 10% bis 90% Helligkeit liegt. Wegen dieser Einschränkung findet der Modus nur in Ausnahmefällen Verwendung.

2.4 Dunkelabgleich von boreCONTROL

Bei der Erstinbetriebnahme und nach jedem Wechsel der Sensorlanze sollte ein Dunkelabgleich durchgeführt werden. Damit wird der Signalpegel erfasst, der bereits ohne Messobjekt vorhanden ist.

Bei optisch temperaturkompensierten Lanzen ist der Referenzpeak immer sichtbar. Der Bereich um den Referenzpeak muss daher beim Dunkelabgleich ignoriert werden. Die Information über die Lage des Referenzpeaks ist eine Sensoreigenschaft und wird über die zugehörige Kalibrierdatei bereitgestellt. Sobald diese geladen wurde, kann zum Dunkelabgleich auch der Bedienknopf des Controllers verwendet werden. In der nachfolgenden Abbildung ist eine typische Dunkelwerttabelle, bestehend aus den immer vorhandenen Spektrumsanteilen der optischen Komponenten und ausgeblendetem Referenzpeak links vom Splitpixel, dargestellt.

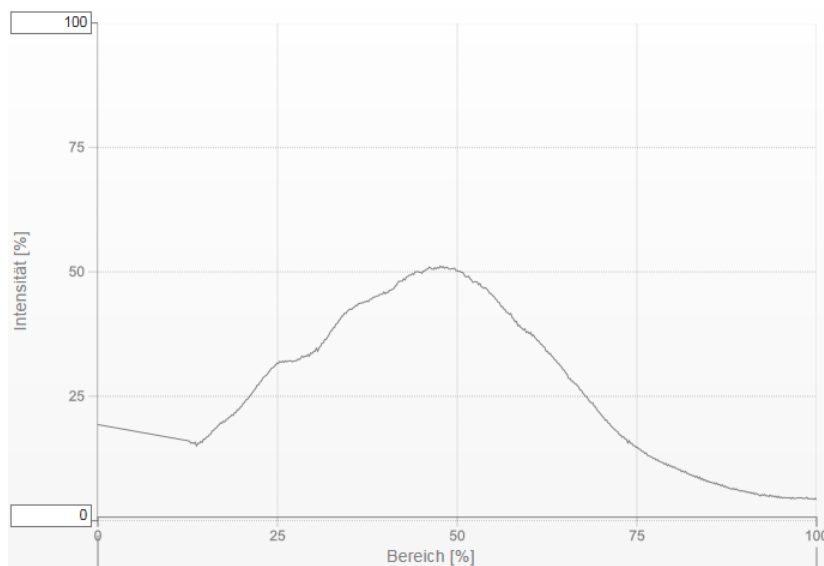


Abbildung 3: Typische Dunkelwerttabelle mit linearem Anteil links vom Splitpixel

Der ohnehin immer vorhandene Anteil des Signalpegels wird anschließend bereits im Controller vom Messsignal abgezogen, noch bevor die Bestimmung der Position von Mess- und Referenzpeak durchgeführt wird. Das Rohsignal ist für den Benutzer somit nicht sichtbar.

Um die Dunkelwertkalibrierung durchzuführen, darf sich kein Messobjekt im Messbereich befinden. Die Punktrate wird für die Dunkelwertkalibrierung mittels „boreSetScanrate“ auf 300 Hz gesetzt, danach die Belichtungszeit per „boreSetShutterTime“ auf 3000 μ s. Anschließend wird die Funktion „boreCalibDark“ aufgerufen. Die Funktion erhält die Position des Trennpixels zwischen Referenzpeak und Messpeak (diese ist in der Kalibrierdatei im Feld SplitPixel nachzuschlagen und trennt den Bereich des Spektrums für die Temperaturkompensation vom Bereich zur Abstandsbestimmung), sowie die Anzahl der durchzuführenden Mittelungen als Parameter. Ausreichende Zeit vorausgesetzt, wird empfohlen, eine Mittelung durchzuführen, die einem ganzzahligen Vielfachen der Datenpunkte pro Umdrehung entspricht, während die Sensorlanze rotiert:

Beschreibung einzelner Abläufe

```
boreSetScanrate(&bore, 300);  
boreSetShutterTime(&bore, 3000);  
boreCalibDark(&bore, 65, 300);
```

Hinweis: Ältere Sensorlanzen, die mit dem IFC2451 Controller ausgeliefert wurden, verwenden 110 statt 65 für die SplitPixel-Position und eine Belichtungszeit von 10000 μ s bei einer Punktrate von 100 Hz.

2.5 Aufnahme und Auswertung von Messdaten

Um einen Datenpuffer mit Messdaten zu befüllen, muss dieser zuvor mit der benötigten Größe angelegt worden sein. Sofern noch nicht geschehen, wird spätestens jetzt die Rotation mittels „boreRotationStart“ gestartet. Anschließend wird die Datenakquisition gestartet, der Puffer befüllt und ausgewertet und die Datenakquisition wieder gestoppt:

```
bore_data_t data;  
bore_eval_t eval;  
  
boreDataAlloc(&data, 10000);  
boreDataStart(&bore);  
boreDataGetRevolution(&bore, &data);  
boreEvaluate(&bore, &data, &eval);  
boreDataStop(&bore);  
boreDataFree(&data);
```

Auswertungsergebnisse wie Passkreiszentrum und Durchmesser, aber auch diverse weitere Informationen stehen danach zur weiteren Bearbeitung auf der so befüllten Datenstruktur bore_eval_t zur Verfügung.

Um die Sensorlanze im Messobjekt zu zentrieren, können das Füllen des Puffers und die Auswertung des Passkreises beliebig oft wiederholt werden.

2.6 Winkel-Kalibrierung von boreCONTROL

Nach dem Einschalten stimmen die Winkelpositionen im Rotationscontroller und im konfokalen Controller nicht notwendigerweise überein. Die korrekte Position wird im Rotationscontroller durch Rotation auf die Nullposition via „boreRotationHome“ gesetzt. Dazu dreht die Rotationseinheit langsam in die Null-Position. Spätestens nach einer Vollumdrehung kennt auch der konfokale Controller diese Null-Position. Die vom konfokalen Controller gezählten Winkel-Ticks werden dort bei jeder Umdrehung wieder auf Null zurückgesetzt und sind damit synchron zur Rotationseinheit.

Rotationseinheit und Sensorlanzen weisen eine Markierung auf, die beim Einsetzen der Sensorlanze in Übereinstimmung gebracht werden muss. Damit zeigt der Lichtauslass von verschiedenen Lanzen relativ zur Rotationseinheit in dieselbe Richtung. Allerdings ist die Winkel-Null-Position der Rotationseinheit von außen nicht direkt erkennbar.

Zur Zentrierung der Sensorlanze im Messobjekt ist es hilfreich, das Koordinatensystem definiert auszurichten, so dass sich die Bewegungsrichtung beim Zentrieren mit der Änderung der Zentrumskoordinaten bei der Auswertung deckt.

Um die Lage des Koordinatensystems wie gewünscht festzulegen, ist pro Sensorlanze einmalig zur Inbetriebnahme eine Bestimmung des Winkel-Offsets vom Sensor-Koordinatensystem zum Welt-Koordinatensystem notwendig. Dies kann beispielsweise durch ein justiertes Target durchgeführt werden, das die Richtung des Koordinatensystems durch eine entsprechende Markierung definiert, z.B. eine dunkel gelaserte Linie.

Nach Akquisition einer Vollumdrehung und Auswertung der Markierung kennt man den Tick-Zähler „tick_counter_of_mark“ für diese Position. Die gewünschte Richtung für den Nullwinkel kann durch entsprechende Korrektur des Winkel-Tick-Offsets durch Aufruf von

```
boreRotationSetTickOffset(&bore, - tick_counter_of_mark);
```

oder alternativ durch

```
boreRotationSetAngleZero(&bore, angle_position);
```

verändert werden. Dieser Schritt muss dann bei jedem Neustart des Controllers, allerdings mit dann schon bekanntem Parameterwert, erneut durchgeführt werden.

Die Ausrichtung kann auch mechanisch durchgeführt werden. Dazu stoppt man die Rotation („boreRotationStop“), dreht die Lanze auf 0° Position („boreRotationMoveTo“) und richtet die Rotationseinheit durch manuelles Drehen einmalig wie gewünscht aus. Zur nachfolgenden Datenakquisition muss die Rotationseinheit dann wieder mit „boreRotationStart“ gestartet werden.

2.7 Offset-Kalibrierung von boreCONTROL

Mittels Offset-Kalibrierung wird der Abstand des Austrittspunkts des Messstrahls aus der Sensorlanze von der Rotationsachse ausgeglichen und damit kleinere Abweichungen, die beim Einsetzen der Sensorlanze auftreten, kompensiert.

Um eine Offset-Kalibrierung durchführen zu können, muss die Sensorlanze in dem mitgelieferten Kalibrierring oder einem ähnlich präzise gefertigten Kalibriertarget mit bekanntem Durchmesser platziert, zentriert und anschließend die Belichtungszeit geeignet eingestellt werden.

Folgende Arbeitsschritte sind nach dem Öffnen der Verbindung zu boreCONTROL für eine Offset-Kalibrierung durchzuführen:

```
bore_data_t data;
bore_eval_t eval;

boreRefCalibLoad(&bore, " calibration_file.txt ");
boreRefCalibSetMode(&bore, BORE_REFCALIB_ONREQUEST );
boreRotationSetSpeed(&bore, 2.0);
boreRotationStart(&bore);
boreSetScanrate(&bore, 1000);
boreSetShutterTime(&bore, 200);
boreRefCalibSingle(&bore);

boreDataAlloc(&data, 10000);
boreDataStart(&bore);
do
{
    boreDataGetRawRevolution(&bore, &data);
    boreEvaluate(&bore, &data, &eval);
}
while(sqrt(eval.center_x * eval.center_x + eval.center_y * eval.center_y) > 100.0);
boreDataStop(&bore);

boreCalibrateOffset(&bore, 4000.0 – eval.radius);
boreDataFree(&data);
```

Dies bedeutet im Detail:

- Setzen Sie die für die verwendete Sensorlanze vorgesehene Kalibriertabelle. Rufen Sie dazu die Funktion „boreRefCalibLoad“ auf und aktivieren Sie die Verwendung der Kalibriertabelle durch Aufruf von „boreRefCalibSetMode“ mit Parameter BORE_REFCALIB_ONREQUEST.
- Setzen Sie die Rotationsgeschwindigkeit („boreRotationSetSpeed“) und starten Sie die Rotation („boreRotationStart“).
- Setzen sie die Punktrate mittels „boreSetScanrate“ auf den gewünschten Wert, in Keramik-Kalibrierringen normalerweise 1000 Hz. Anschließend stellen Sie die Belichtungszeit und damit die Intensität des Signals mit der Funktion „boreSetShutterTime“ geeignet ein, so dass Helligkeitswerte im Bereich von 10% bis 90% produziert werden. Das Setzen der Belichtungszeit kann entfallen, wenn statt des Modus BORE_REFCALIB_ONREQUEST der Modus BORE_REFCALIB_ONREQAUTOSH verwendet wird.
- Initialisieren Sie die Temperaturkompensation mittels „boreRefCalibSingle“. Ohne diesen Schritt können die intern erfassten Messwerte in Onrequest-Modi nicht in μm konvertiert werden.
- Stellen Sie einen Datenpuffer für die aquirierten Daten mittels „boreDataAlloc“ in ausreichender Größe bereit.
- Starten Sie die Datenakquisition („boreDataStart“).

- Zur Justierung des Zentrums akquirieren Sie einen Vollkreis aus Daten („boreDataGetRawRevolution“, identisch zu „boreDataGetRevolution“, sofern als Offset 0.0 gesetzt ist) und werten Sie die Daten aus („boreEvaluate“). Wiederholen Sie die Zentrierung und Auswertung, bis die Beträge der Zentrumskoordinaten kleiner als 50 μm , besser 10 μm sind.
- Stoppen Sie die Datenakquisition mittels „boreDataStop“.
- Der Unterschied zwischen Kalibrierring-Radius und dem zuletzt ausgewertetem Radius wird als neuer Offset-Wert gesetzt („boreCalibrateOffset“).
- Falls nicht mehr benötigt, wird der Datenpuffer mittels „boreDataFree“ freigegeben.

Durch Mittelung der Auswertungen an unterschiedlichen Stellen im Kalibrierring kann der Offset-Wert noch genauer eingestellt werden.

Das System ist nun für Absolutmessungen kalibriert. Offsetwerte von mehr als 30 μm sind im Regelfall nicht zu erwarten.

Außerdem haben Sie die Möglichkeit, mehrere Ringe für eine Korrektur der Kalibriertabelle zu verwenden. Akquirieren Sie dazu, ebenenweise oder als Spiralscan, durch alle Ringe mindestens 50 Umdrehungen pro Ring in einen hinlänglich großen Datenpuffer. Alternativ können Sie den Datenpuffer auch aus Teilscans einzelner Ringe zusammensetzen. Bei vormontierten Kalibrierring-Sätzen wird eine Datei mitgeliefert, die die Ringstapel beschreibt:

```
# BCCSI 1 4
3996 3000
6000 7000
7999 10000
9998 10000
```

Die erste Zeile legt das Dateiformat (BCCSI, boreCONTROL Calibration Stack Info), die Datei-Format-Version (1) und die Anzahl der enthaltenen Ringe (4) fest. Die weiteren Zeilen enthalten den Ring-Nenndurchmesser und die Ring-Höhen, jeweils in μm .

Damit lässt sich eine modifizierte Kalibrierdatei mittels Daten in bore_data_t data wie folgt erstellen:

```
bore_recalibrate_eval_t refcal_eval;
bore_calstack_info_t calstack;
bore_refcalib_t refcal;

boreCalStackReadInfo(&calstack, stackinfo_name);
boreCalStackProcess(&data, &calstack);
boreRefCalibRead(&refcal, calfile_name);
boreRefCalibRecalibrateStack(&refcal, &caleval, calstack.count, &refcal_eval);
boreRefCalibWrite(&refcal, outfile_name);
boreRefCalibFree(&refcal);
```

Auf die Abfrage der Rückgabewerte wurde dabei aus Darstellungsgründen verzichtet.

2.8 Qualitätskontrolle mittels Videodatenerfassung

Im normalen Messbetrieb werden Referenzpeak- und Messpeak-Position als Schwerpunkte eines auf einer Videozeile abgebildeten Spektrums vom konfokalen Controller berechnet und bereitgestellt. Um Verschmutzungen im optischen Weg zu prüfen reicht dies nicht aus. Hierzu benötigt man die Informationen aus der Sensorzeile des Controllers. Zu diesem Zweck steht eine Schnittstelle zum Auslesen der Video-Information zur Verfügung. Man geht dazu wie folgt vor:

- Zur Kontrolle des optischen Weges ist es erforderlich, dass die Sensorlanze nicht in einem Messobjekt platziert ist und kein signifikantes Restlicht aufnimmt.
- Die Belichtungszeit wird, wie beim Dunkelabgleich, bei niedriger Scanrate sehr hoch eingestellt, typischerweise 3000 μs bei 300 Hz.
- Nachdem die normale Messdatenakquisition gestoppt ist, wird die Übertragung der Video-Daten mittels Aufruf von „boreVideoStreamStart“ aktiviert. Der zusätzlich übergebene Parameter ist eine Bitkombination aus in bore_videotype_t definierten Bitmasken und selektiert ein oder mehrere gewünschten Kanäle. Zur Prüfung des optischen Weges wird BORE_VIDEOTYPE_RAW verwendet.

- Ab dem Start der Videoübertragung liefert der Controller mit hoher Geschwindigkeit Video-Daten, die zur Vermeidung von Puffer-Überläufen auch fortlaufend von der Applikation abgeholt werden müssen. Dies geschieht durch Aufrufe von „boreVideoStreamGet“ zur Ablage in einem internen Zwischenspeicher, aus dem die angeforderten Kanäle mittels „boreVideoStreamCopy“ in den Anwendungspuffer übertragen werden können. Die Größe der Video-Zeile zum Anlegen des Puffers kann mittels „boreVideoStreamGetTableSize“ abgefragt werden und ist 512 für die aktuell verwendeten Controller IFC24x1.
- Nach dem Abholen der benötigten Anzahl von Video-Daten wird die Übertragung mittels Aufruf von „boreVideoStreamStop“ angehalten.
- Für den einfachen Fall der Qualitätsprüfung mit einem einzigen Kanal sind die obigen Schritte mit Start, Abholen der Daten, Stop und Kopieren der Daten in einer einzigen Funktion „boreVideoStreamSingle“ zusammengefasst:

```
double table[512];  
  
boreSetScanrate(&bore, 300.0);  
boreSetShutterTime(&bore, 3000.0);  
boreVideoStreamSingle(&bore, BORE_VIDEOTYPE_RAW, table, 512);
```

Bei der verwendeten hohen Belichtungszeit übersteuert der Referenzpeak. Dieser Bereich des Spektrums kann für die Beurteilung der Signalqualität ignoriert werden. Die maximale Intensität im Bereich des Messpeaks, der mangels Messobjekt nur aus den akkumulierten Effekten des optischen Weges besteht, sollte nicht höher als 90%, besser 70% liegen. Anderenfalls ist normalerweise eine Reinigung notwendig. Ein zu hoher Wert kann aber auch durch die Beschädigung einer Teilkomponente hervorgerufen werden.

Um die zu reinigende Komponente zu identifizieren, wird der Test mit schrittweise entfernten Komponenten (Sensorlanze, Rotationseinheit und optischem Kabel am Anschluss des Controllers) wiederholt, wobei zu berücksichtigen ist, dass sich die akkumulierten Effekte ebenso reduzieren. Fällt die Intensität gegenüber dem vorhergehenden Schritt stark ab, so ist die verschmutzte Koppelstelle identifiziert und kann gereinigt werden.

2.9 Filterung von Daten

2.9.1 Sektor-Filterung

Bei dezentrierter Platzierung der Sensorlanze im Bohrloch ist der Auftreffwinkel des Lichtstrahls auf der Bohrlochoberfläche stellenweise ungünstig und der Messwert damit möglicherweise gestört, während nahe der Stellen mit kleinstem und größtem Abstand der Auftreffwinkel auf die Oberfläche nahezu senkrecht ist.

Um die Bestimmung des Passkreises zu optimieren, kann man die Datenpunkte von der Kreispassung ausschließen, die abseits der Stellen mit minimalem und maximalem Abstand liegen. Dazu löscht man das Bit BORE_FLAG_VALID auf dem Element „flag“ der Datenstruktur bore_meas_t für die Punkte, die nicht verwendet werden sollen.

Dies wird durch die Funktion

```
boreFilterInvalidateTwoAngleSectors(bore_data_t *data, double selang, double selrng)
```

erledigt, die Punkte mit Winkelabstand größer als „selrng“ von dem vorgegebenen Winkel „selang“ und dem gegenüberliegenden Winkel (jeweils in Grad) invalidiert.

Zur Bestimmung des Winkels mit kleinstem Abstand kann die Funktion

```
boreFilterGetMinDistAngle(double offset_x, double offset_y)
```

verwendet werden, die die Zentrumskoordinaten aus einer vorangegangenen Kreispassung zu einer Winkelposition konvertiert.

In Abhängigkeit von der Dezentrierung der Sensorlanze ist es sinnvoll, die Größe der für die Kreispassung verwendeten Sektoren zu variieren. Während bei einer kleinen Dezentrierung alle Punkte verwendet werden können, sollte die Größe der Sektoren mit wachsender Dezentrierung immer kleiner werden, aber nie unter eine gewisse Mindestgröße fallen, um genug Punkte für eine stabile Kreispassung übrig zu behalten. Die Bestimmung der halben Sektorgröße zur Verwendung in „boreFilterInvalidateTwoAngleSectors“ leistet die Funktion

```
boreFilterGetAngleRangeFromRamp(  
    double offset_x, double offset_y, double rampstart, double rampend, double endangle)
```

die das Passkreiszentrum (offset_x, offset_y) und die Parameter einer Rampenfunktion als Argumente erhält. Für Dezentrierung kleiner rampstart wird 90° zurückgegeben, für Werte über rampend das letzte Argument endangle. Zwischen rampstart und rampend wird affin linear interpoliert.

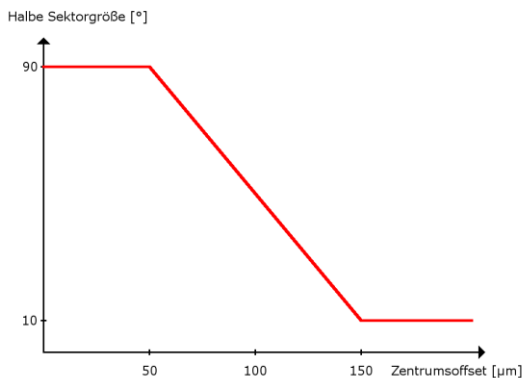


Abbildung 4: Beispiel für die Rampenfunktion

Sinnvolle Werte sind beispielsweise rampstart=50 [µm] und rampend=150 [µm] mit endangle=10 [°]. Insbesondere die Wahl des Winkelbereichs für große Dezentrierungen hängt aber von der gewählten Punktrate und Rotationsgeschwindigkeit ab, die Werte sollten also experimentell optimiert werden.

2.9.2 Median-Filterung

Um beispielsweise bei rauen Oberflächen einzelne Ausreißer zu eliminieren, steht eine Median-Filterung

```
boreFilterMedian(  
    const bore_data_t *data, int halfsize, bore_filter_flags_t flags, bore_data_t *outdata)
```

bzw. für Teilbereiche von Datenpuffern (startidx bis ausschließlich endidx, Zielindex outidx)

```
boreFilterMedianRange(  
    const bore_data_t *data, int startidx, int endidx, int halfsize, bore_filter_flags_t flags,  
    bore_data_t *outdata, int outidx);
```

zur Verfügung, die aus vorgegebenen Messdaten data und der halben Filtergröße halfsize unter Berücksichtigung von flags einen zuvor mit geeigneter Größe angelegten Ausgabepuffer outdata füllt. Die Filter-Optionen flags beinhalten

- BORE_FILTER_NO_WRAPAROUND:
Die Filterung wird nicht am anderen Datenpufferende fortgesetzt, was bei Spiralscans sinnvoll ist, da die Daten an den Enden des Datenpuffers von unterschiedlichen Positionen stammen.
- BORE_FILTER_OMIT_INVALID:
Zuvor ungültige Punkte werden nicht durch gültige Filterwerte ersetzt, sondern bleiben auch in der Ausgabe ungültig. Datenlücken bleiben so erhalten.
- BORE_FILTER_BARY:
Es wird statt des Distanzwerts das Feld für den Messwert-Schwerpunkt gefiltert.

2.9.3 Mittelwert-Filterung

Um eine Glättung der Daten zu erzielen, steht analog zur Median-Filterung eine Mittelwert-Filterung via boreFilterAverage bzw. boreFilterAverageRange zur Verfügung.

3. Kurzbeschreibung der einzelnen Funktionen

Die Funktionen der boreCONTROL.dll gliedern sich in mehrere Funktionsgruppen.

Verbindung	Verbindung mit boreCONTROL aufbauen und trennen
Rotationseinheit	Funktionen zur Ansteuerung der Rotationseinheit
Z-Achse	Funktionen zur Interpretation der Z-Achsen-Positionsdaten
Parameter	Funktionen zum Setzen und Auslesen von Scan-Einstellungen
Videoerfassung	Funktionen zur Videosignal-Erfassung
Kalibrierung	Funktionen zur Linearisierungs-Kalibrierung im Sensor-Controller
Offset-Kalibrierung	Funktionen zur Kalibrierung des Sensor-Offsets
Datenerfassung	Funktionen rund um die Daten-Akquisition
Daten Import/Export	Funktionen zum Lesen und Schreiben von Daten
Datentyp-Unterstützung	Funktionen zur Unterstützung spezieller Datentypen
Filterung	Funktionen zur Datenfilterung
Auswertung	Funktionen zur Auswertung akquirierter Daten
Datenaustausch	Funktionen zum Datenaustausch
Referenzkalibrierung	Funktionen zur optischen Temperaturkompensation
Referenzkalibrierung Bearbeitung	Funktionen zur Bearbeitung von Referenzkalibrierungen
Kalibrierstack-Verarbeitung	Funktionen zur Auswertung von Kalibrierstacks

Die detaillierte Beschreibung der einzelnen Funktionsaufrufe und Datenstrukturen finden Sie in der beiliegenden HTML-SDK-Dokumentation.

3.1 Funktionen zur Verbindungsherstellung

boreConnect	Verbinden mit den Komponenten von boreCONTROL
boreDisconnect	Verbindung mit den boreCONTROL Komponenten trennen
boreGetControllerVersion	Auslesen der Sensor-Controller Versionsinformation
boreGetDllVersion	Auslesen der Versionsinformation der boreCONTROL-Dll
boreGetErrorMessage	Abholen der Beschreibung zu aufgetretenen Fehlern
boreGetControllerTemperature	Abholen der internen Controllertemperatur

3.2 Funktionen zur Ansteuerung der Rotationseinheit

boreRotationSetSpeed	Setzen der Umdrehungsgeschwindigkeit der Rotationseinheit
boreRotationGetSpeed	Auslesen der Umdrehungsgeschwindigkeit der Rotationseinheit
boreRotationStart	Start der Rotationseinheit
boreRotationStop	Stop der Rotationseinheit
boreRotationGetTicks	Auslesen der Winkel-Ticks für eine volle Umdrehung
boreRotationSetTickOffset	Setzen des Offsets für die Winkel-Ticks
boreRotationGetTickOffset	Auslesen des Offsets für die Winkel-Ticks
boreRotationAddTickOffset	Winkeltick-Offset relativ zum eingestellten Wert verändern
boreRotationGetAngle	Aktuelle Winkelposition der Rotationseinheit relativ zur Null-Position

Kurzbeschreibung der einzelnen Funktionen

boreRotationGetAngleRaw	Aktuelle Hardware-Winkelposition der Rotationseinheit
boreRotationSetAngleZero	Setzen der Null-Position der Rotationseinheit
boreRotationGetAngleZero	Aktuelle Null-Position der Rotationseinheit
boreRotationAddAngleZero	Null-Position relativ zum eingestellten Wert verändern
boreRotationHome	Rotationseinheit auf Null-Position fahren
boreRotationMoveTo	Rotationseinheit zu vorgegebenem Winkel-Tick fahren
boreRotationMoveAbsAngle	Absolute Winkelposition anfahren
boreRotationMoveRelAngle	Position relativ zur aktuellen Position anfahren

3.3 Funktionen zur Interpretation der Z-Achsen-Information

boreZAxisSetScale	Setzen des Z-Achsen-Skalierungsfaktors Ticks zu μm
boreZAxisGetScale	Auslesen des Z-Achsen-Skalierungsfaktors Ticks zu μm
boreZAxisSetOffset	Setzen des Z-Achsen-Offsets in μm
boreZAxisGetOffset	Auslesen des Z-Achsen-Offsets in μm
boreZAxisZero	Nullsetzen der Z-Achsen-Position
boreZAxisSetPosition	Setzen der aktuellen Z-Achsen-Position in μm

3.4 Funktionen zum Setzen der Scan-Parameter

boreSetIntensity	Setzen der LED-Intensität des Controllers (nur IFC2401)
boreGetIntensity	Auslesen der LED-Intensität des Controllers (nur IFC2401)
boreSetShutterTime	Setzen der Belichtungszeit
boreGetShutterTime	Auslesen der Belichtungszeit
boreSetShutterTimes	Setzen der Belichtungszeiten im Dual-Shutter Modus
boreGetShutterTimes	Auslesen der Belichtungszeit im Dual-Shutter Modus
boreSetScanrate	Setzen der Scanrate des Controllers
boreGetScanrate	Auslesen der Scanrate des Controllers

3.5 Funktionen zur Videodatenerfassung

boreVideoStreamStart	Starten der Videodaten-Erfassung
boreVideoStreamStop	Stoppen der Videodaten-Erfassung
boreVideoStreamGet	Abholen von Videodaten
boreVideoStreamCopy	Kopieren der abgeholten Videodaten
boreVideoStreamSingle	Abholen und kopieren einzelner Videodaten
boreVideoStreamGetTableSize	Abholen der Größe der Videodaten-Tabellen

3.6 Funktionen zur Kalibrierung

boreCalibSetTable	Setzen der verwendeten Kalibriertabelle
boreCalibGetTable	Auslesen der verwendeten Kalibriertabelle
boreCalibGetRange	Auslesen des Messbereichs der aktuell eingestellten Kalibriertabelle
boreCalibDarkSignal	Dunkelabgleich für hardware-kompensierte Sensoren (nur IFC2401)
boreCalibDark	Dunkelabgleich für optisch kompensierte Sensoren
boreCalibAngleReference	Justierung der Winkelreferenz (nur IFC2401)
boreCalibGetRanges	Auslesen der Messbereiche für alle Kalibriertabellen

3.7 Funktionen zur Offset-Kalibrierung

boreSetOffset	Setzen des Werts für die Offset-Kalibrierung
boreGetOffset	Auslesen des verwendeten Werts für die Offset-Kalibrierung
boreDataAddOffset	Offset-Korrektur eines Datenpuffers
boreDataApplyFactorOffsetRange	Faktor und Offset auf Teilbereich eines Datenpuffers verrechnen
boreDataApplyFactorOffset	Faktor und Offset auf Datenpuffer verrechnen
boreSetFactor	Setzen des Steigungskorrektur-Faktors für die Kalibrierung
boreGetFactor	Auslesen des Steigungskorrektur-Faktors für die Kalibrierung
boreCalibrateOffset	Offset-Kalibrierung für einen vorgegebenen Ringdurchmesser in μm
boreCalibrateOffsetMulti	Offset-Kalibrierung für mehrere Ringdurchmesser in μm

3.8 Funktionen zur Datenakquisition

boreDataAlloc	Allokieren eines Messdatenpuffers
boreDataFree	Freigeben eines Messdatenpuffers
boreDataReset	Rücksetzen eines Messdatenpuffers
boreDataGetRevCount	Anzahl der Umdrehungen in einem Datenpuffer
boreDataGetRevIndices	Startindizes für Umdrehungen in einem Datenpuffer
boreDataRevalidate	Neusetzen der Valid-Markierung im Messdatenpuffer
boreDataStart	Start von Akquisition und Datentransfer
boreDataStop	Stop von Akquisition und Datentransfer
boreDataRestart	Leeren des internen Datenpuffers bei gestarteter Akquisition
boreDataAvailable	Auslesen der Anzahl der verfügbaren Datenpunkte
boreDataGetRawRevolutionPartial	Aufsammeln weiterer Daten ohne Kalibrieroffset, ohne zu Blockieren
boreDataGetRevolutionPartial	Aufsammeln weiterer Daten mit Kalibrieroffset, ohne zu Blockieren
boreDataGetRawRevolution	Aufsammeln eines Vollkreises an Daten ohne Kalibrieroffset
boreDataGetRevolution	Aufsammeln eines Vollkreises an Daten mit Kalibrieroffset
boreDataGetRaw	Auslesen von Messdaten ohne Kalibrieroffset

Kurzbeschreibung der einzelnen Funktionen

boreDataGet	Auslesen von Messdaten mit Kalibrieroffset
boreDataAppendRange	Anhängen eines Teils eines Datenpuffers an einen anderen Puffer
boreDataAppend	Anhängen eines Datenpuffers an einen anderen Datenpuffer

3.9 Funktionen zum Lesen und Schreiben von Daten

boreDataIOGetSize	Auslesen der Anzahl der Datenwerte in einer Datei
boreDataIOReadAppend	Anhängen der Messdaten einer Datei an einen Datenpuffer
boreDataIORead	Lesen von Messdaten von Datei
boreDataIOReadAlloc	Lesen von Messdaten von Datei, Datenpuffer passend allokkieren
boreDataIOWrite	Schreiben von Messdaten auf Datei
boreDataIOGetFilesetSize	Anzahl Dateien und Messpunkte einer Menge von Dateien
boreDataIOReadFilesetAppend	Anhängen einer Menge von Dateien an einen Datenpuffer
boreDataIOReadFileset	Einlesen einer Menge von Dateien in einen Datenpuffer
boreDataIOReadFilesetAlloc	Einlesen einer Menge von Dateien, Datenpuffer passend allokkieren

3.10 Funktionen zur Unterstützung spezieller Datentypen

boreMatrixAlloc	Allokieren einer Matrix
boreMatrixFree	Freigeben einer Matrix
boreMatrixSet	Matrix mit Konstante beschreiben

3.11 Funktionen zur Datenfilterung

boreFilterInvalidateAngleSectors	Invalidieren von Daten außerhalb vorgegebener Sektoren
boreFilterInvalidateTwoAngleSectors	Invalidieren von Daten außerhalb gegenüberliegender Sektoren
boreFilterGetAngleRangeFromRamp	Berechnung der Sektorgröße mittels Rampenfunktion
boreFilterGetMinDistAngle	Berechnung des Winkels mit minimalem Abstand
boreFilterMedian	Median-Filterung
boreFilterMedianRange	Median-Filterung auf einem Teil eines Datenpuffers
boreFilterAverage	Mittelwert-Filterung
boreFilterAverageRange	Mittelwert-Filterung auf einem Teil eines Datenpuffers
boreFilterResample	Datenpuffer auf äquidistantem Raster neu auswerten
boreFilterResampleRange	Teil eines Datenpuffers auf äquidistantem Raster neu auswerten

3.12 Funktionen zur Auswertung

boreEvalSetIntensityThres	Setzen des unteren Intensitäts-Schwellwerts in % für die Auswertung
boreEvalGetIntensityThres	Lesen des unteren Intensitäts-Schwellwerts in % für die Auswertung
boreEvalSetIntensityUpperThres	Setzen des oberen Intensitäts-Schwellwerts in % für die Auswertung
boreEvalGetIntensityUpperThres	Lesen des oberen Intensitäts-Schwellwerts in % für die Auswertung

Kurzbeschreibung der einzelnen Funktionen

boreEvaluate	Messdatenauswertung mittels Kreispassung
boreEvaluateCal	Messdatenauswertung von Rohdaten zu Kalibrierzwecken
boreEvaluateData	Messdatenauswertung nur mit Datenpuffer
boreEvaluateCalData	Messdatenauswertung Kalibrierung, nur mit Datenpuffer
boreEvaluateDataRange	Messdatenauswertung für Indexbereich
boreEvaluateCalDataRange	Messdatenauswertung Kalibrierung für Indexbereich
boreEvaluateDataRevs	Auswertung mehrerer Umdrehungen mit Index Vektor
boreEvaluateCalDataRevs	Auswertung Kalibrierung, mehrere Umdrehungen mit Index Vektor

3.13 Funktionen zum Datenaustausch

boreMeasGetFlag	Extrahieren des Flag-Elements in ein Datenfeld
boreMeasGetDistance	Extrahieren des Distanz-Elements in ein Datenfeld
boreMeasGetIntensity	Extrahieren des Intensitäts-Elements in ein Datenfeld
boreMeasGetCounter	Extrahieren des Datenzähler-Elements in ein Datenfeld
boreMeasGetBarycenter	Extrahieren des Schwerpunkt-Elements in ein Datenfeld
boreMeasGetAngleRad	Extrahieren des Winkel-Elements in ein Datenfeld
boreMeasGetXPos	Extrahieren des X-Positions-Elements in ein Datenfeld
boreMeasGetYPos	Extrahieren des Y-Positions-Elements in ein Datenfeld
boreMeasGetZPos	Extrahieren des Z-Positions-Elements in ein Datenfeld
boreMeasGetTempCelsius	Extrahieren des Temperatur-Elements in ein Datenfeld
boreMeasGetAngleTicks	Extrahieren des Winkelzähler-Elements in ein Datenfeld
boreMeasGetRevTicks	Extrahieren des Umdrehungszähler-Elements in ein Datenfeld
boreMeasGetZTicks	Extrahieren des Z-Achsen-Zähler-Elements in ein Datenfeld
boreMeasGetRefIntensity	Extrahieren des Referenzpeak-Intensitäts-Elements in ein Datenfeld
boreMeasGetRefBarycenter	Extrahieren des Referenzpeak-Schwerpunkt-Elements in ein Datenfeld
boreMeasGetShutter	Extrahieren des Shutter Elements in ein Datenfeld

3.14 Funktionen zur Referenz-Kalibrierung

boreRefCalibLoad	Laden einer Kalibriertabelle zur optischen Temperaturkompensation
boreRefCalibSingleData	Referenzpeak-Ermittlung aus einem vorhandenen Datenpuffer
boreRefCalibSingle	Durchführung einer Einzelmessung zur Referenzpeak-Ermittlung
boreRefCalibSetMode	Setzen des Temperaturkompensations-Modus
boreRefCalibGetMode	Auslesen des Temperaturkompensations-Modus
boreRefCalibSetSingleMode	Setzen des Modus für Referenzpeak-Erfassung
boreRefCalibGetSingleMode	Auslesen des Modus für Referenzpeak-Erfassung
boreRefCalibSetShutterTime	Setzen der Belichtungszeit zur Referenzpeak-Ermittlung

Kurzbeschreibung der einzelnen Funktionen

boreRefCalibGetShutterTime	Auslesen der Belichtungszeit zur Referenzpeak-Ermittlung
boreRecCalibSetScanrate	Setzen der Scanrate zur Referenzpeak-Ermittlung
boreRefCalibGetScanrate	Auslesen der Scanrate zur Referenzpeak-Ermittlung

3.15 Funktionen zur Bearbeitung von Referenz-Kalibrierungen

boreRefCalibFree	Freigeben einer initialisierten Referenzkalibrierstruktur
boreRefCalibRead	Einlesen einer Referenzkalibrierung
boreRefCalibWrite	Abspeichern einer Referenzkalibrierung
boreRefCalibSet	Verwendung einer zuvor geladenen Referenzkalibrierung
boreRefCalibRecalibrate	Umkalibrierung einer Referenzkalibrierung an neuen Stützstellen
boreRefCalibRecalibrateStack	Umkalibrierung einer Referenzkalibrierung mit Calstack-Auswertung
boreRefCalibGetComment	Auslesen des Benutzer-Kommentars einer Referenzkalibrierung
boreRefCalibSetComment	Setzen des Benutzer-Kommentars einer Referenzkalibrierung
boreRefCalibGetPeakRanges	Auslesen des Referenzpeak- und Messpeak-Wertebereichs
boreRefCalibEvaluate	Umrechnung von Peakpositionen in μm

3.16 Funktionen zur Bearbeitung von Kalibrierstapeln

boreCalStackReadInfo	Einlesen von Informationen zu Kalibrierstapeln
boreCalStackWriteInfo	Schreiben von Informationen zu Kalibrierstapeln
boreCalStackProcessRevs	Verarbeitung von Daten von Kalibrierstapeln
boreCalStackProcess	Vollautomatische Auswertung von Daten von Kalibrierstapeln

4. Beschreibung der Datenstrukturen

4.1 Verbindungsaufbau – bore_connect_t und Aufzählungstypen

Die Datenstruktur bore_connect_t wird nur beim Verbinden mit den boreCONTROL-Komponenten in der Funktion „boreConnect“ verwendet.

```

/// boreCONTROL hardware controller
typedef enum {
    BORE_CONTROLLER_NONE      = 0,    ///< no controller selected
    BORE_CONTROLLER_IFC2401   = 1,    ///< boreCONTROL IFC2401 controller
    BORE_CONTROLLER_IFC2431   = 2,    ///< boreCONTROL IFC2431 controller
    BORE_CONTROLLER_IFC2451   = 3,    ///< boreCONTROL IFC2451 controller
    BORE_CONTROLLER_IFC2471   = 4,    ///< boreCONTROL IFC2471 controller
    BORE_CONTROLLER_IFC2461   = 5,    ///< boreCONTROL IFC2461 controller
} bore_controller_t;

/// boreCONTROL hardware rotation unit
typedef enum {
    BORE_ROTATION_NONE       = 0,    ///< no controller selected
    BORE_ROTATION_BCC2410    = 1,    ///< boreCONTROL BCC2410 controller
} bore_rotation_t;

/// boreCONTROL hardware temperature
typedef enum {
    BORE_TEMPERATURE_NONE    = 0,    ///< no temperature sensor selected
    BORE_TEMPERATURE_NN      = 1,    ///< boreCONTROL temperature sensor
} bore_temperature_t;

/// parameters for boreCONTROL connection
typedef struct bore_connect_ {
    bore_controller_t controller; ///< boreCONTROL controller selection
    int controller_table;        ///< boreCONTROL sensor table
    int usb_instance_number;     ///< usb based controllers, 0 is first instance
    char tcpip_address[32];     ///< eth based controllers, server address
    int tcpip_port;              ///< eth based controllers, command port

    bore_rotation_t rotation_unit; ///< rotation unit hardware
    int rotation_com_port;        ///< serial port of rotation unit, COM1 == 1

    bore_temperature_t temp_unit; ///< boreCONTROL temperature sensor
} bore_connect_t;

```

Die Datenstruktur ist wie folgt zu füllen:

- **controller:** Typ des verwendeten Controllers, normalerweise BORE_CONTROLLER_IFC2461, aber auch BORE_CONTROLLER_IFC2451 oder BORE_CONTROLLER_IFC2471. BORE_CONTROLLER_IFC2401 und BORE_CONTROLLER_IFC2431 werden nicht mehr eingesetzt.
- **controller_table:** Immer 0, da die Kalibriertabelle bei IFC2451, IFC2461 und IFC2471 softwareseitig zur Verfügung gestellt wird
- **usb_instance_number:** 0, nicht verwendet für IFC2451, IFC2461 und IFC2471
- **tcpip_address:** Die IP-Adresse des IFC2451, IFC2461 oder IFC2471, normalerweise „169.254.168.150“, sofern die Adresse kundenseitig nicht geändert wurde
- **tcpip_port:** 0 (oder 23), wird bei Werten <= 0 automatisch auf 23 gesetzt
- **rotation_unit:** BORE_ROTATION_BCC2410
- **rotation_com_port:** Serieller Port für BCC2410, 1 entspricht COM 1 u.s.w.
- **temp_unit:** Nicht verfügbar, daher BORE_TEMPERATURE_NONE

Sollen nur Standardwerte verwendet werden, so kann statt der bore_connect_t Datenstruktur auch ein NULL-Zeiger an die Funktion boreConnect übergeben werden.

4.2 Akquisitionsmodi – bore_refcalib_mode_t

Über bore_refcalib_mode_t wird in „boreRefCalibSetMode“ der Akquisitionsmodus ausgewählt:

```

/// when to do reference calibration
typedef enum {
    BORE_REFCALIB_OFF           = 0,    ///< reference calibration not used
    BORE_REFCALIB_ONREQUEST    = 1,    ///< reference calibration on request
    BORE_REFCALIB_ALWAYS       = 2,    ///< reference calibration done always
    BORE_REFCALIB_ONREQ2AUTO   = 3,    ///< on request, autoselect from 2 shutters
    BORE_REFCALIB_ONREQAUTOSH  = 4,    ///< on request, auto shutter
    BORE_REFCALIB_REFAUTOSH    = 5     ///< reference peak only, auto shutter
} bore_refcalib_mode_t;

```

4.3 Sensordaten - bore_data_t und bore_meas_t

bore_data_t stellt einen Vektor von Messpunkten zur Verfügung. Der Vektor wird mittels „boreDataAlloc“ initialisiert, durch „boreDataFree“ wird der zuvor allokierte Speicher wieder freigegeben.

Zum Zurücksetzen der Datenstruktur ist die Funktion „boreDataReset“ zu verwenden. Sie setzt die Anzahl der Datenpunkte im Puffer und den Offset auf 0, gibt den allokierten Speicher aber nicht frei.

```

/// buffer with measurement points
typedef struct bore_data_ {
    bore_meas_t *data;           ///< data array
    int  maxsize;               ///< allocated size of data array
    int  currsz;                ///< used size of data array
    double offset;              ///< offset added to data
} bore_data_t;

```

Die Datenstruktur besteht aus folgenden Einträgen:

- **data:** Zeiger auf die Daten der Messpunkte, siehe bore_meas_t
- **maxsize:** Die Anzahl der allokierten Datenpunkte in „data“
- **currsz:** Die aktuell verwendete Anzahl der Datenpunkte in „data“
- **offset:** Der Offset in μm aus der Offset-Kalibrierung, der gegebenenfalls zu den Distanzen in den Messpunkten addiert wurde. Für die Akquisition von Daten stehen jeweils zwei verschiedene Varianten zur Verfügung. Bei Abholung der Daten mittels der Raw-Variante (boreDataGetRawRevolution/boreDataGetRaw) ist der Offset 0, zum Addieren des aktuell eingestellten Kalibrieroffsets verwenden Sie anschließend die Funktion „boreDataAddOffset“. Die Variante ohne Namensanteil „Raw“ (boreDataGetRevolution/boreDataGet) addiert bereits den eingestellten Offset aus der Offset-Kalibrierung.

Die Datenpunkte werden durch den Datentyp bore_meas_t realisiert:

```

/// single measurement point
typedef struct bore_meas_ {
    bore_flag_t flag;           ///< valid flag, first of circle, last of circle
    double distance;           ///< measurement peak distance in  $\mu\text{m}$ 
    double intensity;          ///< measurement peak intensity, range [0%-100%]
    int counter;               ///< measurement point counter
    double barycenter;         ///< measurement raw distance in sensor units
    double angle_rad;          ///< angle converted to radian 0..2PI from encl
    double x_pos;              ///< angle and distance converted to x position
    double y_pos;              ///< angle and distance converted to y position
    double z_pos;              ///< z_ticks converted to  $\mu\text{m}$ 
    double temp_celsius;       ///< if temperature sensor connected, else 20°C
    int angle_ticks;           ///< angle tick counter
    int rev_ticks;             ///< revolution tick counter
    int z_ticks;               ///< z-axis tick counter
    double ref_intensity;      ///< reference peak intensity
    double ref_barycenter;     ///< reference peak distance in sensor units
    double shutter;            ///< shutter time in microseconds
} bore_meas_t;

```

Die einzelnen Strukturelemente enthalten folgende Information:

- **flag:** Verschiedene Markierungen für den Messpunkt, definiert im Aufzählungsdatentypen bore_flag_t. Valide Messpunkte (BORE_FLAG_VALID) sind solche mit einem Helligkeitswert über dem eingestellten Schwellwert. Bei der Akquisition von Vollkreisen ist auch der erste (BORE_FLAG_FIRST) und letzte Messpunkt (BORE_FLAG_LAST) gesondert markiert.
- **distance:** Der kalibrierte Abstand des Messpunkts vom Messtarget in μm . Die Umrechnung aus der Sensor-Position erfolgt über die zuvor ausgewählte Kalibriertabelle bereits im Controller.
- **intensity:** Die Helligkeitsinformation des Messpunkts in Prozent
- **counter:** Ein fortlaufender Zähler für akquirierte Messpunkte
- **barycenter:** Messpeak-Koordinaten des Sensors in Pixeln, für Kalibrierzwecke
- **angle_rad:** Winkel in Bogenmaß (rad), berechnet aus dem Zähler angle_ticks
- **x_pos:** x-Koordinate in μm , berechnet aus distance und angle_rad
- **y_pos:** y-Koordinate in μm , berechnet aus distance und angle_rad
- **z_pos:** z-Koordinate in μm , berechnet aus z_ticks
- **temp_celsius:** Temperatur in $^{\circ}\text{C}$, kann bei Bedarf mit der internen Controller-Temperatur befüllt werden.
- **angle_ticks:** Zähler der Winkel-Ticks vom Encoder der Rotationseinheit.
- **rev_ticks:** Zähler der Vollumdrehungs-Ticks vom Encoder der Rotationseinheit, nur bei IFC2401 angeschlossen, sonst frei verwendbar
- **z_ticks:** Zähler der Z-Ticks vom Encoder der Z-Achse, soweit vorgesehen/vorhanden
- **ref_intensity:** Die Helligkeitsinformation des Referenzpeaks in Prozent
- **ref_barycenter:** Die Referenzpeak-Koordinaten des Sensors in Pixeln, zur optischen Kalibrierung
- **shutter:** Verwendete Shuttertime in Mikrosekunden

4.4 Auswertungsinformation – bore_eval_t

bore_eval_t stellt ausgewählte Informationen nach der Kreispassung zur Verfügung und wird entweder durch „boreEvaluate“, „boreEvaluateCal“, „boreEvaluateData“, „boreEvaluateCalData“ oder als Teil von bore_offset_cal_t über „boreCalibrateOffset“ gefüllt.

```

/// evaluation of data
typedef struct bore_eval_ {
    int total_points;           ///< total number of points supplied
    int valid_points;          ///< number of valid data points

    double radius;             ///< fit circle radius in  $\mu\text{m}$ 
    double diameter;          ///< fit circle diameter in  $\mu\text{m}$ 

    double center_x;          ///< signed center x deviation
    double center_y;          ///< signed center y deviation

    double avg_intensity;     ///< average intensity in %
    double min_intensity;     ///< minimum intensity in %
    double max_intensity;     ///< maximum intensity in %

    double deviation_outside;  ///< max deviation away from center, positive
    double deviation_inside;   ///< max deviation towards center, positive

    // extended information, mainly for calibration purposes

    bore_eval_flag_t flags;    ///< evaluation warnings
    double ring_diameter;     ///< filled before recalib: ring diameter in  $\mu\text{m}$ 

    double avg_refbary;       ///< reference peak bary center, if available
    double avg_refintensity;  ///< reference peak intensity, if available
    double min_refintensity;  ///< minimum reference peakintensity in %
    double max_refintensity;  ///< maximum reference peak intensity in %

    double avg_measbary;      ///< measurement peak bary center, pixel
    
```



```
double measbary_center_x;    ///< measurement peak center x position, pixel
double measbary_center_y;    ///< measurement peak center y position, pixel

double measdev_outside;     ///< meas peak max deviation away from center
double measdev_inside;     ///< meas peak max deviation towards center

double avg_shutter;         ///< average shutter time
double min_shutter;        ///< minimum shutter time
double max_shutter;        ///< maximum shutter time
} bore_eval_t;
```

Die Elemente der Datenstruktur enthalten im Einzelnen:

- **total_points**: Anzahl der Datenpunkte im ausgewerteten Datenpuffer
- **valid_points**: Anzahl der vom Controller für eine volle Umdrehung gelieferten Datenpunkte, die einen Helligkeitswert im eingestellten Intervall (siehe „boreEvalSetIntensityThres“ und „boreEvalSetIntensityUpperThres“) aufweisen und damit zur Auswertung und Kreispassung (siehe „boreEvaluate“) herangezogen werden.
- **radius**: Der Radius des Passkreises an die Datenwerte in μm
- **diameter**: Der Durchmesser des Passkreises an die Datenwerte in μm
- **center_x**: Die x-Koordinate des Passkreisentrums in μm . Die Lage des Koordinatensystems wird durch die Definition des 0° -Winkels eingestellt, der die Richtung der positiven x-Achse vorgibt (siehe „boreRotationSetTickOffset“).
- **center_y**: Die y-Koordinate des Passkreisentrums in μm . Die positive y-Achse ist von oben gesehen um 90° in mathematisch positivem Sinn gegenüber der x-Achse gedreht.
- **avg_intensity**: Durchschnittliche Helligkeit über alle validen Datenpunkte in %
- **min_intensity**: Minimal gemessene Helligkeit aller Datenpunkte in %. Diese kleinste Helligkeit kann den eingestellten Schwellwert (siehe „boreEvalSetIntensityThres“) nicht unterschreiten.
- **max_intensity**: Maximal gemessene Helligkeit aller Datenpunkte in %. Diese größte Helligkeit kann den eingestellten Schwellwert (siehe „boreEvalSetIntensityUpperThres“) nicht überschreiten.
- **deviation_outside**: Größte Abweichung eines Datenpunkts vom Passkreis nach außen mit positivem Vorzeichen, in μm
- **deviation_inside**: Größte Abweichung eines Datenpunkts zum Passkreis in Richtung zum Kreiszentrum mit positivem Vorzeichen, in μm

Die Messwerte „min_intensity“ und „max_intensity“ zeigen die Sättigung an. Diese wird neben dem Material und den Reflexionseigenschaften des Messobjekts auch maßgeblich durch die eingestellte Punktrate und die Belichtungszeit beeinflusst und hat Auswirkungen auf die Berechnung der Profildaten. Die Sättigung sollte bei einer Messung zwischen 10 % und 90 % betragen.

Der Rest der Datenstruktur enthält zusätzliche Angaben, die vor allem bei der Rekalibrierung Verwendung finden:

- **flags**: Zusätzliche Markierungen zur Spezifikation von Fehlerzuständen bei der Auswertung, siehe unten
- **ring_diameter**: Beim Auswertungsaufwurf „boreEvaluateCal“ oder „boreEvaluateCalData“ spezifizierter Durchmesser des vermessenen Objekts in μm , typischerweise der Durchmesser eines Kalibrierrings
- **avg_refbary**: Durchschnittliche Position des Referenzpeaks in Pixel
- **avg_refintensity**: Durchschnittliche Intensität des Referenzpeaks in %
- **min_refintensity**: Kleinste Intensität aller Referenzpeaks in %
- **max_refintensity**: Größte Intensität aller Referenzpeaks in %
- **avg_measbary**: Durchschnittlicher Messpeak anhand Kreispassung in Pixel
- **measbary_center_x**: x-Koordinate des Passkreisentrums für den Messpeak in Pixel
- **measbary_center_y**: y-Koordinate des Passkreisentrums für den Messpeak in Pixel

- **measdev_outside:** Größte Abweichung eines Messpeaks vom Passkreis nach außen mit positivem Vorzeichen in Pixel
- **measdev_inside:** Größte Abweichung eines Messpeaks zum Passkreis in Richtung Kreiszentrum mit positivem Vorzeichen in μm
- **avg_shutter:** Durchschnittlicher Wert für Belichtungszeit in μs
- **min_shutter:** Kleinster Wert für Belichtungszeit in μs
- **max_shutter:** Größter Wert für Belichtungszeit in μs

In `bore_eval_flag_t` werden Markierungen zu verdächtigen Daten bei der Auswertung zusammengefasst, die als Warnungen zu interpretieren sind. Die dabei verwendeten Schwellwerte beziehen sich ausdrücklich auf Kalibrierringe und spielen für reale Messobjekte keine Rolle.

```

/// extended evaluation bits, primarily for use with gauge rings
typedef enum {
    /// general evaluation flags, set by evaluation
    BORE_EVAL_NONE = 0x0000,    ///< unmarked
    BORE_EVAL_LOW_MIN_INTENSITY = 0x0001,    ///< minimum intensity too low
    BORE_EVAL_HIGH_MAX_INTENSITY = 0x0002,    ///< maximum intensity too high
    BORE_EVAL_LOW_REF_INTENSITY = 0x0004,    ///< min ref intensity too low
    BORE_EVAL_HIGH_REF_INTENSITY = 0x0008,    ///< max ref intensity too high
    BORE_EVAL_LOW_INTENSITY = 0x0010,    ///< overall intensity too low
    BORE_EVAL_HIGH_INTENSITY = 0x0020,    ///< overall intensity too high
    BORE_EVAL_WARN_OFF_CENTER = 0x0040,    ///< center deviation too large
    BORE_EVAL_WARN_HIGH_DEVIATION = 0x0080,    ///< suspicious deviation to fit
    BORE_EVAL_WARN_REF_POSITION = 0x0100,    ///< suspicious ref peak position
    BORE_EVAL_WARN_MEAS_POSITION = 0x0200,    ///< suspicious meas peak pos
    BORE_EVAL_WARN_FEW_VALID = 0x0400,    ///< low percentage of valid points
    BORE_EVAL_WARN_REF_INTENSITY = 0x0800,    ///< high ref peak intensity deviation

    /// set by recalibration
    BORE_EVAL_HIGH_TABLE_OFFSET = 0x010000,    ///< high offset to original table
    BORE_EVAL_RING_IGNORED_AVG = 0x020000,    ///< ring ignored for averaging

    /// user defined content
    BORE_EVAL_USER_MASK = 0xFF000000    ///< bitmask for user defined bits
} bore_eval_flag_t;

```

4.5 Offset-Kalibrier-Information – `bore_offsetcal_t`

`bore_offsetcal_t` besteht aus den Ergebnissen der Offset-Kalibrierung sowie den Auswertinformationen der zur Offset-Kalibrierung ausgeführten Kreispassung. Die Datenstruktur wird durch die Funktion „`boreCalibrateOffset`“ gefüllt.

```

/// offset calibration result info
typedef struct bore_offsetcal_ {
    int successful;    ///< bool: overall offset calibration status
    double offset;    ///< generated calibration offset
    double factor;    ///< generated calibration slope factor
    double expected_diameter;    ///< expected diameter in  $\mu\text{m}$ , calibration target

    int intensity_ok;    ///< bool: whether intensity was ok
    int center_ok;    ///< bool: whether centering was ok
    int deviation_ok;    ///< bool: whether deviation from fit circle was ok

    bore_eval_t eval;    ///< evaluation results for detailed analysis
} bore_offsetcal_t;

```

Die Einträge in der Datenstruktur haben folgende Bedeutung:

- **successful:** Wird gesetzt, wenn die Offset-Kalibrierung insgesamt erfolgreich war. Dabei müssen sowohl `intensity_ok`, als auch `center_ok` und `deviation_ok` gesetzt worden sein.
- **offset:** Der berechnete Offset aus der Offset-Kalibrierung für die Daten
- **factor:** Der berechnete Korrekturfaktor zur Steigungskorrektur
- **expected_diameter:** Der Kalibrierring-Durchmesser, der bei der Offset-Kalibrierung gesetzt war

- **intensity_ok:** Die Intensität lag innerhalb des voreingestellten sinnvollen Bereichs. Für eine genauere Analyse siehe min_intensity und max_intensity unter Abschnitt 4.4.
- **center_ok:** Das Passkreiszentrum lag innerhalb des voreingestellten Bereichs
- **deviation_ok:** Die Punktabweichungen vom Passkreis lagen innerhalb des für Kalibrierringe üblichen Bereichs
- **eval:** Detailinformationen zur Kreispassung, siehe Abschnitt 4.4.

4.6 Kalibrierstack-Information – bore_calstack_info_t und bore_calstack_eval_t

Die Auswertung von Kalibrierstapeln wird in den nachfolgend aufgeführten Datenstrukturen bereitgestellt:

```

/** calibration stack evaluation for single calibration ring */
typedef struct bore_calstack_eval_ {
    /* segmentation result */
    int ring_processed;    /**< whether processed successfully */
    int start_rev;        /**< start revolution index, included */
    int end_rev;          /**< end revolution index, included */
    int inner_start_rev;  /**< start revolution index inner_height, included */
    int inner_end_rev;    /**< end revolution index inner_height, included */
    double height;        /**< detected height in um */
    double inner_height;  /**< used inner height of ring in um (def 5mm) */

    /* averaging result */
    double avg_diam;      /**< average diameter (inner part only)*/
    double avg_refbary;   /**< average refbary position */
    double avg_measbary;  /**< average measbary position */
    double avg_center_x;  /**< average center x position */
    double avg_center_y;  /**< average center y position */
    double start_center_x; /**< center x position at start of ring */
    double start_center_y; /**< center y position at start of ring */
    double end_center_x;  /**< center x position at end of ring */
    double end_center_y;  /**< center y position at end of ring */
} bore_calstack_eval_t;

/** maximum number of rings in calibration stack */
#define BORE_CALSTACK_INFO_MAX 25

/** calibration ring stack information */
typedef struct bore_calstack_info_ {
    /* required input information */
    int count;                /**< number of rings present */
    double diameter[BORE_CALSTACK_INFO_MAX]; /**< ring diameter in micron units */
    double height[BORE_CALSTACK_INFO_MAX];    /**< ring height in micron units */

    /* computation result */
    bore_calstack_eval_t eval[BORE_CALSTACK_INFO_MAX]; /**< evaluation result */
} bore_calstack_info_t;

```

Die Einträge in der Datenstruktur bore_calstack_info_t werden normalerweise aus Kalibrierstack-Info-Dateien mittels „boreCalStackReadInfo“ geladen, können aber auch manuell befüllt werden:

- **count:** Anzahl der enthaltenen Kalibrierringe, maximal BORE_CALSTACK_INFO_MAX
- **diameter:** Feld von Durchmesserwerten der einzelnen Kalibrierringe
- **height:** Feld mit Höhen der einzelnen Kalibrierringe
- **eval:** Feld mit Auswertungsergebnissen aus einer Kalibrierstackanalyse

Für jeden Ring sind in bore_calstack_eval_t folgende Auswertungsergebnisse verfügbar:

- **ring_processed:** Markierung, ob die Auswertung erfolgreich war
- **start_rev:** Index der ersten Umdrehung im Ring
- **end_rev:** Index der letzten Umdrehung im Ring

Beschreibung der Datenstrukturen

- **inner_start_rev**: Index der ersten zum ausgewerteten inneren Bereich des Rings gehörenden Umdrehung (der Ringrand weist stärker schwankende Durchmesserwerte auf und wird daher für die Auswertung nicht berücksichtigt)
- **inner_end_rev**: Index der letzten zum ausgewerteten inneren Bereich des Rings gehörenden Umdrehung (der Ringrand weist stärker schwankende Durchmesserwerte auf und wird daher für die Auswertung nicht berücksichtigt)
- **height**: Ermittelte Höhe des Rings
- **inner_height**: Verwendete Höhe im Ringinneren
- **avg_diam**: Durchschnittlicher Durchmesser im inneren Bereich des Rings
- **avg_refbary**: Durchschnittliche Position des Referenzpeaks
- **avg_measbary**: Durchschnittliche Position des Messpeaks
- **avg_center_x**: Durchschnittliche Zentrumsposition X
- **avg_center_y**: Durchschnittliche Zentrumsposition Y
- **start_center_x**: X-Koordinate der Zentrumsposition am oberen Ende des Rings
- **start_center_y**: Y-Koordinate der Zentrumsposition am oberen Ende des Rings
- **end_center_x**: X-Koordinate der Zentrumsposition am unteren Ende des Rings
- **end_center_y**: Y-Koordinate der Zentrumsposition am unteren Ende des Rings

Mit den Auswertungsinformationen lässt sich insbesondere mittels „boreRefCalibRecalibrateStack“ eine Rekalibrierung einer bestehenden Kalibriertabelle durchführen.

5. Beschreibung der Fehlercodes

Die Textblöcke sind ein Auszug aus dem Quelltext und daher in Englisch kommentiert.

5.1 Allgemeine Fehler

```
// general error codes
BORE_OK = 0, ///< successful, no error
BORE_ERROR = -1, ///< unspecified general error
BORE_ERROR_OUT_OF_MEMORY = -2, ///< allocation out of memory
BORE_ERROR_UNSUPPORTED = -3, ///< requested feature not supported
BORE_ERROR_UNINITIALIZED = -4, ///< interface not initialized
BORE_ERROR_RANGE = -5, ///< illegal range
BORE_ERROR_INCONSISTENT = -6, ///< internal inconsistency
BORE_ERROR_BUFFER_OVERFLOW = -7, ///< buffer overflow
BORE_ERROR_ACTIVE = -8, ///< not allowed while data is sent
BORE_ERROR_BUFFER_TOO_SHORT = -9, ///< provided buffer is too short
BORE_ERROR_FEATURE_DISABLED = -10, ///< software-disabled feature
```

5.2 Fehler Rotationseinheit

```
// rotation unit related errors
BORE_ERROR_ROTATION_UNIT_UNKNOWN = -200, ///< unknown rotation unit on connect
BORE_ERROR_ROTATION_OPEN_FAILED = -201, ///< failed to open serial port
BORE_ERROR_ROTATION_STATE_FAILED = -202, ///< failed to set port state
BORE_ERROR_ROTATION_TIMEOUTS_FAILED = -203, ///< failed to set port timeouts
BORE_ERROR_ROTATION_SPEED_TOO_LOW = -204, ///< rotation speed too low
BORE_ERROR_ROTATION_SPEED_TOO_HIGH = -205, ///< rotation speed too high
BORE_ERROR_ROTATION_SPEED_FAILED = -206, ///< rotation speed command failed
BORE_ERROR_ROTATION_START_FAILED = -207, ///< rotation start command failed
BORE_ERROR_ROTATION_STOP_FAILED = -208, ///< rotation stop command failed
BORE_ERROR_ROTATION_COMM_FAILED = -209, ///< rotation initial communication failed
BORE_ERROR_ROTATION_SPEED_UNDEFINED = -210, ///< no rotation speed set before start
BORE_ERROR_ROTATION_NEEDS_STOP = -211, ///< rotation unit must be stopped
```

5.3 Fehler Sensor-Controller

```
// controller related errors
BORE_ERROR_CONTROLLER_UNKNOWN = -300, ///< unknown or no controller specified
BORE_ERROR_CONTROLLER_INSTANCE_FAILED = -301, ///< failed to create controller instance
BORE_ERROR_CONTROLLER_OPEN_FAILED = -302, ///< failed to open controller connection
BORE_ERROR_CONTROLLER_CLOSE_FAILED = -303, ///< failed to close controller connection
BORE_ERROR_CONTROLLER_RELEASE_FAILED = -304, ///< failed to release controller instance
BORE_ERROR_CONTROLLER_START_FAILED = -305, ///< failed to start data acquisition
BORE_ERROR_CONTROLLER_STOP_FAILED = -306, ///< failed to stop data acquisition
BORE_ERROR_CONTROLLER_INITIALIZATION = -307, ///< failed to set initial parameters
```

5.4 Fehler bei der Auswertung

```
// evaluation related errors
BORE_ERROR_EVAL_TOO_FEW_VALID_POINTS = -401, ///< too few valid points for circle fit
```

5.5 Fehler bei der Offset-Kalibrierung

```
// calibration related errors
BORE_ERROR_CALIB_UNSUCCESSFUL = -500, ///< calibration not successful
BORE_ERROR_CALIB_OFFSET_NOT_SET = -501, ///< offset calibration value not set (obsolete)
BORE_ERROR_CALIB_ANGLE = -502, ///< angle offset reference error
BORE_ERROR_CALIB_DATA_AMBIGUOUS = -503, ///< data not suitable for slope correction
BORE_ERROR_CALIB_DATA_NOT_LINEAR = -504, ///< factor fit data is not linear
```

5.6 Fehler bei der Referenz-Kalibrierung

```
// reference calibration related errors
BORE_ERROR_REFCALIB_OPEN_FAILED = -600, ///< file-open of refcalib file failed
BORE_ERROR_REFCALIB_READ_FAILED = -601, ///< wrong file format or access problem
BORE_ERROR_REFCALIB_WRITE_FAILED = -602, ///< problem while writing
BORE_ERROR_REFCALIB_UNINITIALIZED = -603, ///< reference calibration not loaded
BORE_ERROR_REFCALIB_ILLEGAL_MODE = -604, ///< unknown reference calibration mode
BORE_ERROR_REFCALIB_STOP_NEEDED = -605, ///< acquisition running, need to stop
BORE_ERROR_REFCALIB_ONREQUEST_NEEDED = -606, ///< refcalib on request mode required
BORE_ERROR_REFCALIB_FEW_DATA = -607, ///< too few data supplied
BORE_ERROR_REFCALIB_BOREPARAM_FAILED = -608, ///< failed setting internal parameters
```

5.7 Fehler beim Datenexport und Datenimport

```
// data io related errors
BORE_ERROR_DATAIO_OPEN_READ      = -700, ///< open of file to read failed
BORE_ERROR_DATAIO_OPEN_WRITE     = -701, ///< open of file to write failed
BORE_ERROR_DATAIO_READ           = -702, ///< read failed
BORE_ERROR_DATAIO_WRITE          = -703, ///< write failed
BORE_ERROR_DATAIO_FORMAT         = -704  ///< read failed because of file format
```

5.8 Fehler beim Verarbeiten von Kalibrierstapeln

```
// calibration stack related errors
BORE_ERROR_CALSTACK_RING_MISSING = -800, ///< ring not found in segmentation
BORE_ERROR_CALSTACK_RING_INCOMPLETE = -801, ///< ring too small in segmentation
BORE_ERROR_CALSTACK_FEWS_REVS     = -802  ///< too few revolutions inside of ring
```

6. SDK und Beispielapplikationen

Software und Dokumentation werden als Zip-Datei bereitgestellt. Die Datei ist an der gewünschten Stelle zu entpacken. Unter dem Wurzelverzeichnis boreCONTROL-SDK befinden sich folgende Verzeichnisse:

- **Development:** Software Development Kit - Development\SDK
 - **Include:** boreCONTROL.h
 - **Lib:** boreCONTROL.lib, boreCONTROL.dll und MedaqLib.dll
 - **Samples:** Beispiel-Projekte
 boreCONTROLDemo (einfache Demonstration)
 boreCONTROLExtDemo (ausführliches Testprogramm)
 boreCONTROLOffline (Auswertung von Exportdateien)
 boreCONTROLRecalib (Rekalibrierung)
 - **Samples\Bin:** Ausführbare Versionen von boreCONTROLDemo, boreCONTROLExtDemo, boreCONTROLOffline und boreCONTROLRecalib, zusammen mit den zur Ausführung benötigten DLLs.
- **Documentation:** Dokumentation Hardware, Dll, Schnittstellendokumentation
 - **Documentation.exe:** Einstiegspunkt in die Dokumentation
 - **What's new.txt:** Änderungen gegenüber den Vorgänger-Versionen
 - **deutsch:** Dokumentation boreCONTROL Hardware und Dll in deutscher Sprache
 - **english:** Dokumentation boreCONTROL Hardware und Dll in englischer Sprache
 - **interface:** Automatisch generierte Schnittstellen-Dokumentation
 - **html:** Hilfsdateien für die Dokumentation
- **Support:**
 - **CME2:** Software zur Motoransteuerung und Konfiguration
 - **SensorFinder:** Software zum Auffinden von Ethernet-basierten Controllern im Netzwerk
 - **vcredist_x86:** Microsoft Redistributable, im Falle von fehlenden Bibliotheken nachzuinstallieren

Das SDK ist erstellt und getestet mit Microsoft VS2008 unter Windows 7, Service Pack 1. Verwenden Sie die Demo-Projekte als Grundlage für Ihre eigenen Applikationen.

6.1 boreCONTROLDemo

In diesem einfachen Demonstrationsprogramm wird die Verbindung zur boreCONTROL-Hardware hergestellt, die Rotationseinheit in Betrieb genommen, die Datenakquisition gestartet, ein Vollkreis mit Rohdaten akquiriert und in das Dateisystem exportiert. Anschließend werden Datenakquisition und Rotation gestoppt und die Verbindung zu boreCONTROL getrennt. Vor Beendigung des Programms wartet die Applikation auf Drücken der Eingabe-Taste.

6.2 boreCONTROLExtDemo

Dieses komplexe Demonstrationsprogramm erlaubt es, über Kommandozeilen-Parameter fast alle zentralen Funktionen der boreCONTROL.dll zu verwenden. Um die boreCONTROLExtDemo-Applikation auszuführen, öffnen Sie unter „Start->Ausführen ...“ die Windows-Eingabeaufforderung „cmd.exe“ und wechseln mit dem Kommando „cd“ in das „Development\Samples\Bin“-Verzeichnis der Installation.

Bei Aufruf von „boreCONTROLExtDemo -h“ wird eine Hilfe zur Verwendung der Programmoptionen in Englisch ausgegeben:

```
Usage: boreCONTROLExtDemo [options]
-----
boreCONTROL extended demo application with command line parameters.
Default is IFC2461 controller at TCP/IP address 169.254.168.150
and rotation unit at com port 1.
If no complex command is issued, a single measurement is done.
Options are:
```

```

Connection options
-ifc2451 addr      connect with ifc2451 at supplied tcpip address
-ifc2461 addr      connect with ifc2461 at supplied tcpip address
-ifc2471 addr      connect with ifc2471 at supplied tcpip address
-comport value     set com port for rotation unit (default 1)
Rotation unit and z-axis finetuning (rarely used)
-anglezero deg     set angle zero position in degree
-zscale value      set z scaling factor tick to um (default 0.1)
-zoffset value     set z offset in um (default 0)
-zzero             set z to 0 for current position
-zsetpos value     set z to value in um for current position
Data acquisition options
-speed value       set rotation speed in Hz range 0.1-10 (default 2)
-shutter value     set shutter time in um (default 200, 0 is auto)
-rate value        set scan rate in Hz (default 2500)
Evaluation
-refcalib file     set reference calibration file
-offset value      set offset for absolute measurement (default 0)
-thres value       set intensity threshold in % (default 5)
-uthres value      set upper intensity threshold in % (default 95)
-baryfit           circle fit using barycenter data, not distance
Optional reference peak acquisition before measurement
-refrate value     set refpeak scan rate in Hz (def 2500)
-refshutter value  set refpeak shuttertime in us (def 400, 0 auto)
Complex commands (only one per command line)
-info              print version information
-dark value        calibrate dark with split position
-angleadj thres    set zero angle by marked target, line thres (-1 auto)
-calib value       determine offset using ring diameter in um
-center value      center adjust mode, exit after iteration count
Output features
-export prefix     write data using prefix, number and .txt are added
-wait              wait for return key press before exit
-quiet            disable trace output
-repeat value      number of repeats in normal evaluation mode
-sleep seconds     seconds between acquisitions for repeat
-v                be verbose (default off)
-h                this help page
    
```

6.2.1 Beispielaufufe boreCONTROLExtDemo

Nachfolgend sind einige typische Aufrufe als Verwendungsbeispiele zusammengestellt. Einige Aufrufe benötigen die zum eingesetzten Sensor passende Kalibriertabelle als Parameter. Diese wird mit `-refcalib` gesetzt.

- Die Dunkelwertkalibrierung wird bei erster Verwendung nach Installation der Sensorlanze durchgeführt. Die Lanzenöffnung muss dabei abgedeckt sein oder es darf sich kein Messobjekt im Messbereich befinden. Für den Dunkelabgleich des IFC2461 Controllern sind wie bei der Werkskalibrierung die Scanrate auf 300 Hz und die Belichtungszeit auf 3000 μ s voreingestellt. Der Parameter 65 für `-dark` im Beispiel unten ist die Position des Trennpixels zwischen Referenzpeak und Messpeak und muss eventuell entsprechend dem Eintrag `SplitPixel` in der Kalibrierdatei angepasst werden. Intern werden 300 Videosignale gemittelt.

```
boreCONTROLExtDemo -rate 300 -shutter 3000 -dark 65
```

oder einfacher

```
boreCONTROLExtDemo -dark 65
```

IFC2451 Controller verwenden stattdessen 100 Hz mit 10000 μ s und Splitpixelposition 110.

- Die Sensorlanze wird innerhalb des Kalibrierrings, beispielsweise mit 6 mm Durchmesser, platziert und grob zentriert. Mit folgendem Aufruf kann, soweit vorhanden, eine manuelle Feinjustierung unterstützt werden. Der Parameter nach der Option `-center` gibt an, nach wie vielen Messungen mit einer Veränderung des Zentrums unter 3 μ m das Programm abbricht. Hinweis: Für die nachfolgende Offset-Kalibrierung reichen Zentrumskoordinaten unter 10 μ m vollkommen aus.

```
boreCONTROLExtDemo -refcalib calfile.txt -center 10
```


Die Angabe einer Kalibriertabelle ist erforderlich, um Zentrumskoordinaten in μm zu erhalten.

- Die Lanze ist damit innerhalb des Passrings zentriert. Um den Offset-Wert für die Absolutmessung zu bestimmen, wird nun eine Kalibrierung durchgeführt. Der Parameter nach `-calib` gibt den Durchmesser des verwendeten Kalibrierrings in μm an. Der resultierende Offset-Wert sollte unter $30\ \mu\text{m}$ liegen.

```
boreCONTROLExtDemo -refcalib calfile.txt -calib 6000.0
```

- Um eine Absolutmessung mit dem zuvor bestimmten Offset durchzuführen, wird `boreCONTROLExtDemo` mit dem Offsetwert aus der Kalibrierung, hier beispielsweise 2.4, aufgerufen.

```
boreCONTROLExtDemo -refcalib calfile.txt -offset 2.4
```

- Die Parameter für die Messung lassen sich ebenso einstellen. Nachfolgend ein Beispiel für 10 Wiederholmessungen mit 1 kHz Scanrate, 4 Hz Rotationsgeschwindigkeit und $200\ \mu\text{s}$ Belichtungszeit (eine Zeile, Zeilenumbruch zur besseren Lesbarkeit):

```
boreCONTROLExtDemo -refcalib calfile.txt
  -offset 2.4 -rate 1000 -speed 4 -shutter 200 -repeat 10
```

- Um die manuelle Zentrierung zu vereinfachen, bietet `boreCONTROLExtDemo` die Möglichkeit, analog zu Abschnitt 2.5 eine Justierung des Koordinatensystems vorzunehmen. Während mittels der Option `-anglezero` das Koordinatensystem wie gewünscht gedreht werden kann, erledigt dies die Option `-angleadj` in einem mit schwarzem Strich an der 0° Position versehenen Target automatisch. Der Parameter nach `-angleadj` gibt den Extraktionsschwellwert in % an, -1 stellt dabei den Schwellwert automatisch ein. Der ermittelte Nullwinkel muss dann bei der Zentrierung oder Messung mittels `-anglezero` angegeben werden.

```
boreCONTROLExtDemo -angleadj -1
```

- Für optisch kompensierte Sensorlanzen steht die option `-refcalib` zur Verfügung, der zusätzlich der Dateiname der zu ladenden Kalibriertabelle mitgegeben wird. Sobald mindestens eine der beiden Optionen `-refshutter` oder `-refrate` angegeben wurde, wird der Kompensationsmodus `BORE_REFCALIB_ONREQUEST` verwendet, die Referenzpeak-Position zur Temperaturkompensation also vor den Messungen ermittelt. Andernfalls wird der Sensor im Modus `BORE_REFCALIB_ALWAYS` betrieben.

Beispiel für `BORE_REFCALIB_ALWAYS`:

```
boreCONTROLExtDemo -refcalib calfile.txt
```

Beispiel für `BORE_REFCALIB_ONREQUEST` auf dunklen Messobjekten:

```
boreCONTROLExtDemo -refcalib calfile.txt
  -rate 1000 -shutter 1000 -refrate 2500 -refshutter 400
```

6.3 boreCONTROLOffline

Mit `boreCONTROLOffline` können zuvor exportierte Daten (`boreDataIOWrite`) nochmals mit geänderten Parametern ausgewertet und Statistikinformationen generiert werden.

Bei Aufruf von „`boreCONTROLOffline -h`“ wird eine Hilfe zur Verwendung der Programmoptionen in Englisch ausgegeben:

```
Usage: boreCONTROLOffline [options] filename
-----
Demonstration for boreCONTROL offline data analysis

Parameters for typical scenarios:"
1) Evaluate a boreCONTROL export file, append evaluation to eval.txt
   Examples: single file, directory, or file pattern
   -write eval.txt -all exportfilename.txt
   -write eval.txt -all dir
   -write eval.txt -all directory/ex_*.txt
2) Evaluate a boreCONTROL export file with recalibration
   and (optional) offset
   -refcalib cal_file.txt -offset 10.0 exportfilename.txt
3) Evaluate a boreCONTROL export file restricting the intensity
   range for evaluation
   -thres 10 -uthres 80 exportfilename.txt
```

- 4) Evaluate a boreCONTROL export file after applying median filter, and export filtered result (export_prefix is prepended to files)
-median 8 -export export_prefix exportfilename.txt
- 5) specify diameter to compare evaluation with expected result
-ring 5001.7 exportfilename.txt
- 6) Evaluation for heavily decentered data with known angle positions for minimum and maximum distance.
Also available with automatic angle determination and sector size selection: 2nd example sector size 90° up to offset 50.0, 5° for offset larger 200.0, ramp from (50.0,90°) to (200.0,5°)
-selangle 30.0 5.0 -selangle 210.0 5.0 exportfilename.txt
-autoangle 50.0 200.0 5 exportfilename.txt
- 7) Evaluation of a set of numbered calibration ring export files
-ring 3997 stacklsingle/ex0*.txt \<\
-ring 5998 stacklsingle/ex1*.txt \<\
-ring 7997 stacklsingle/ex2*.txt \<\
-ring 10007 stacklsingle/ex3*.txt \<\
-all -write evalstack.txt
- 8) Evaluation of a set of calibration ring export files with diameters taken from stack information file (file format is BCCSI 1 nrofrings\\r\\nringdiam ringheight\\r\\n)
Offset computation from multiple export files cutting away the smallest and largest diameter and averaging the rest
2nd example with temperature specified for writing to results file.
-multicut 1 -si calstackinfo.txt ex*.txt
-multicut 1 -si calstackinfo.txt -temp 22.5 ex_temp22_5_*.txt \<\
-results results.txt

Options are:

- thres value evaluation intensity threshold (5)
- uthres value evaluation intensity upper threshold (95)
- refcalib file optional reference calibration file
- offset value offset to add to distances (needs refcalib)
- ring value ring diameter in um for following files/filesets
- baryfit barycenter circle fit for data without distances

Optional filters

- median halves median filter, kernel 2*halves+1
- vmmedian halves median filter valid points only, kernel 2*halves+1
- selangle ctr dev invalidate outside angle ctr+-dev (2x possible)
- autoangle s e a auto sector inval, start and end offset, end angle

Multiple files processing options"

- si file stackinfo mode, generates patterns for each fileset
- sisingle file stackinfo mode, single set, ring auto determination
- multicut cnt discard cnt largest and smallest results for ring"
- multimediam use median instead of average on remaining results"

Output options

- export name export data to disk (useful after filtering)
- write name append evaluation data to file
- old use old evaluation format for compatibility
- calwrite name append ring diameter and barycenters to file"
- results name append to result file, same format as AutoCalib"
- temp celsius temperature for results file"

General options

- wait wait for key press before exit (default off)
- quiet disable trace output (default full output)
- v be verbose, also show dll version (default off)
- h output this help page

Ein typisches Anwendungsbeispiel ist die Neuauswertung von exportierten Daten mit geänderter Kalibrierinformation (eine Zeile):

```
boreCONTROLOffline.exe -refcalib newcalib.txt -thres 5 -uthres 95
    -write statistics.txt exportdata.txt
```

Die Exportdatei „exportdata.txt“ wird geladen und die enthaltenen Peak-Positionsinformationen mittels der Kalibrierdatei „newcalib.txt“ neu in μm umgerechnet. Punkte mit Helligkeit unter 5% und über 95% werden ausgeschlossen, die verbleibenden Messpunkte einer Kreispassung zugeführt. Die Berechnungsergebnisse werden ausgegeben und zusammen mit berechneten Statistikinformationen

zusätzlich an die Ausgabedatei „statistics.txt“ angehängt. Diese Datei lässt sich für weitere Analysen beispielsweise in Microsoft Excel importieren.

6.4 boreCONTROLRecalib

Das boreCONTROL-System wird mit einer Werkskalibrierung ausgeliefert. In Ausnahmefällen kann es vorkommen, dass die Messwerte nicht mehr exakt mit der Kalibrierung übereinstimmen. In diesem Fall ist es kundenseitig möglich, vorhandene Kalibrierdateien an neue Stützstellendaten von Kalibrierringen anzupassen. Dazu wird das Programm boreCONTROLRecalib eingesetzt.

Als Eingabe benötigt boreCONTROLRecalib sowohl die Werkskalibrierung einschließlich der Temperaturkompensationsinformation, sowie eine Anzahl von Dateien mit exportierten Messdaten aus verschiedenen, mindestens jedoch zwei, Kalibrierringen, die den verwendeten Messbereich abdecken sollten. Da selbst Kalibrierringe fertigungsbedingte Durchmesserschwankungen aufweisen und vereinzelt auch nach einer Reinigung Verschmutzungen im Kalibrierring verbleiben können, ist es vorteilhaft, mehrere Messungen an unterschiedlichen Positionen für jeden Kalibrierring bereitzustellen.

Um die Zuordnung zum verwendeten Kalibrierring sicherzustellen, wird vorausgesetzt, dass der Dateiname nach dem ersten einleitenden Unterstrich „_“ den Durchmesser des verwendeten Kalibrierrings in Mikrometern enthält. Es sind zwei Formate möglich: Entweder eine Punkt-separierte Fließkomma-Darstellung oder eine Ganzzahl-Notation, die nach einem weiteren Unterstrich optional die Nachkommastellen enthält. Beispiele sind:

```
export_6350.2_01.txt      (6350,2 µm, Messung 01)
calib_6350_2_14.txt      (6350,2 µm, Messung 14)
ring_8890_set_7.txt      (8890,0 µm, Messung 7)
```

Diese Dateien werden beim Aufruf von boreCONTROLRecalib, zwingend aufsteigend nach Durchmessern, zusammen mit der Werkskalibrierdatei übergeben (eine Zeile):

```
boreCONTROLRecalib.exe -c cal3mm.txt
    Ex_4999_5_000.txt  Ex_4999_5_100.txt  Ex_4999_5_200.txt
    Ex_6998_7_000.txt  Ex_6998_7_100.txt  Ex_6998_7_200.txt
    Ex_9501_0_000.txt  Ex_9501_0_100.txt  Ex_9501_0_200.txt
```

Die erzeugte Kalibrierdatei wird als „out.txt“ im aktuellen Verzeichnis abgelegt. Der Name kann aber durch die Kommandozeilenoption „-o“, gefolgt vom gewünschten Dateinamen, auch anders festgelegt werden.

Bei Aufruf von „boreCONTROLRecalib -h“ wird eine Hilfe zur Verwendung der Programmoptionen in Englisch ausgegeben:

```
Usage: boreCONTROLRecalib [options] -c calfile list-of-data-files
-----
Recalibration of calibration files using calibration ring data
Uses either single revolution data in several calibration rings
or a complete scan through a calibration ring stack

The calibration stack variant takes a calibration stack information
file containing ring count, diameters and heights as parameter and
needs a file path (with wildcard in case of multiple export files),
or a single export files containing a complete scan, as argument(s).
File sets with wildcard character are loaded in alphabetical order.

The single revolution data variant needs needs a list of export files
as parameters. It's required that the export file names contain the
diameter of the according gauge ring after the first underscore,
e.g. ex_6350.2_a.txt or ex_6350_2_b.txt (in micron units)
Data files should be supplied in a sequence with ascending diameters.
Consecutive multiple data files for one diameter are averaged before
being used for recalibration.

Options are:
-h          Print this help page
-c calfile  Calibration file to modify (required)
-stack file Switches to calstack mode using information from file
-cd diff_file Write calibration difference analysis to named file
-o name     Name of output calfile (default: out.txt)
-thres val  Lower evaluation threshold for data files (default 5)
```

-uthres val	Upper evaluation threshold for data files (default 95)
-parse n	Diameter is after nth underscore (default is 0 auto)
-nowait	Do not wait before exit

Die betreffende Rekalibrier-Funktionalität wird auch über das SDK bereitgestellt, um sie einfach in kundenseitige Applikationen integrieren zu können. Siehe dazu „boreRefCalibRecalibrate“ in der SDK-Schnittstellen-Dokumentation.

Hinweis: Die neu erzeugte, veränderte Kalibrierung kann nur so gut wie die zugrundeliegenden Daten sein. Diese sollten deswegen mit höchster Sorgfalt akquiriert werden. Insbesondere ist auf folgende Punkte zu achten:

- eine gute Zentrierung (idealerweise unter 5 μm Zentrumsoffset von der Sensorachse)
- eine ausreichende Anzahl von verwendeten Kalibrierringen (3 oder mehr)
- eine ausreichende Anzahl von Positionen im Kalibrierring zur Mittelung (5 oder mehr)
- eine gute Abdeckung des verwendeten Messbereichs
- gut gereinigte und ölfreie Kalibrierringe ohne Roststellen und Ablagerungen
- möglichst alle Intensitäten von Messpunkten sollten zwischen 20% und 95% liegen
- möglichst mindestens 95% der Messpunkte sollten als gültig markiert sein (5% bis 99% Helligkeit)

7. Hinweise zum Umgang mit boreCONTROL

Beachten Sie bei der Bedienung von boreCONTROL folgenden Hinweis:

Trennen Sie bei laufender Software niemals die Ethernet-Verbindung zwischen boreCONTROL und der Systemeinheit.

Schalten Sie bei laufender Software niemals die Spannungsversorgung von boreCONTROL aus.

Dies erfordert eventuell einen Neustart des PC und/oder des Controllers.



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.de

X9750240-A061097JHS

