

1 YEAR
WARRANTY



Ω OMEGA® **Handbuch**



Online-Webshop
omega.de

E-Mail: info@omega.de
Aktuelle Handbücher:
www.omegamanual.info

TC-08
8 kanaliges USB-
Datenerfassungsmodul für
Thermoelemente

www.omega.de	E-Mail: info@omega.de
--	--

Technische Unterstützung und Applikationsberatung erhalten Sie unter:

Deutschland, Österreich, Schweiz
OMEGA Engineering GmbH
Daimlerstraße 26
D-75392 Deckenpfronn
Tel: +49 (0) 7056 9398-0, Fax: +49 (0) 7056 9398-29
Gebührenfrei: 0800 8266342
E-Mail: info@omega.de

Weltweit: www.omega.com/worldwide/

USA
OMEGA Engineering, Inc.
Customer Service: 1-800-622-2378 (nur USA und Kanada)
Engineering Service: 1-800-872-9436 (nur USA und Kanada)
Tel: (203) 359-1660, Fax: (203) 359-7700
Gebührenfrei: 1-800-826-6342 (nur USA und Kanada)
Website: www.omega.com
E-Mail: info@omega.com

Fester Bestandteil in OMEGAs Unternehmensphilosophie ist die Beachtung aller einschlägigen Sicherheits- und EMV-Vorschriften. Produkte werden sukzessive auch nach europäischen Standards zertifiziert und nach entsprechender Prüfung mit dem CE-Zeichen versehen.

Die Informationen in diesem Dokument wurden mit großer Sorgfalt zusammengestellt.

OMEGA Engineering, Inc. kann jedoch keine Haftung für eventuelle Fehler übernehmen und behält sich Änderungen der Spezifikationen vor.

WARNUNG: Diese Produkte sind nicht für den medizinischen Einsatz konzipiert und dürfen nicht an Menschen eingesetzt werden.

Inhaltsverzeichnis

1 Einführung

1.1 Übersicht	5
1.2 Installieren des Treibers	5
1.3 Anschließen des USB TC-08	6
1.4 Sicherheitswarnung	9

2 Produktinformationen

2.1 Technische Daten	10
2.2 So funktioniert ein Thermoelement	11

3. Technische Referenz

3.1 Einführung	12
3.2 Betriebsarten	13
3.2.1 Einführung	13
3.2.2 Streamingmodus	13
3.2.3 Einzelabfragemodus	15
3.2.4 Legacymodus	17
3.3 Treiberrouinen	19
3.3.1 Einführung.....	19
3.3.2 Neuer USB-Modus und Legacymodus	20
3.3.2.1 usb_tc08_open_unit	20
3.3.2.2 usb_tc08_open_unit_async	21
3.3.2.3 usb_tc08_open_unit_progress	23
3.3.2.4 usb_tc08_close_unit	24
3.3.2.5 usb_tc08_stop	24
3.3.2.6 usb_tc08_set_mains	24
3.3.2.7 usb_tc08_get_minimum_interval_ms	25
3.3.2.8 usb_tc08_get_unit_info	25
3.3.2.9 usb_tc08_get_formatted_info	26
3.3.2.10 usb_tc08_get_last_error	27
3.3.3 Nur neuer USB-Modus	29
3.3.3.1 usb_tc08_set_channel	29
3.3.3.2 usb_tc08_run	30
3.3.3.3 usb_tc08_get_single	31
3.3.3.4 usb_tc08_get_temp	32
3.3.3.5 usb_tc08_get_temp_deskew	33

Inhaltsverzeichnis

3.3.4 Nur Legacymodus	35
3.3.4.1 usb_tc08_legacy_run	35
3.3.4.2 usb_tc08_legacy_set_channel	36
3.3.4.3 usb_tc08_legacy_get_temp	37
3.3.4.4 usb_tc08_legacy_get_cold_junction	38
3.3.4.5 usb_tc08_legacy_get_driver_version	38
3.3.4.6 usb_tc08_legacy_get_version	39
3.3.4.7 usb_tc08_legacy_get_cycle	39
3.4 Programmierung	40
3.4.1 Einführung	40
3.4.2 C und C++	40
3.4.3 Delphi	40
3.4.4 Excel	41
3.4.5 LabVIEW	41
3.4.6 Visual Basic	41
3.4.7 Agilent VEE 3	41
3.5 Fehlersuche	42
3.6 Glossar	43

1. EINFÜHRUNG

1.1 Übersicht

Der USB TC-08 ist ein Logger für Temperaturen und Spannungen, der mehrere Thermoelemente unterstützt. Mit der zum Lieferumfang gehörigen TC-08-Software lässt sich das Gerät auf jedem Laptop unter Windows XP SP2 oder Vista einsetzen. Für technisch orientierte Anwender wird mit dem Gerät ein Treiber geliefert, der die Einbindung in eigene Programme ermöglicht.

Der Treiber stellt auch eine Vergleichsstellen-Kompensation für Thermoelemente bereit. Die TC-08-Software unterstützt bis zu 20 USB TC-08-Geräte. Bei entsprechender Ausstattung des PCs unterstützt der Treiber sogar bis zu 64 Geräte.

Diese Anleitung beschreibt die physikalischen und elektrischen Eigenschaften des USB TC-08 und erläutert die Verwendung der Software-Treiber. Informationen zur USB TC-08-Software finden Sie in der Hilfedatei der Software.

1.2 Installieren des Treibers

Der Treiber wird automatisch zusammen mit der TC-08-Software installiert. Alternativ können Sie den Treiber von unserer Website unter <http://www.omega.com> herunterladen.

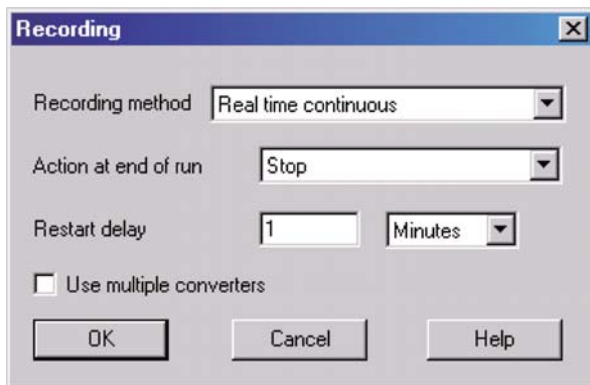
1.3 Anschließen des USB TC-08

Bitte beachten Sie: Schließen Sie den USB TC-08 erst an, nachdem Sie die Software installiert und Ihren PC neu gestartet haben.

Schließen Sie den USB TC-08 als ersten Schritt mit dem mitgelieferten Kabel an den USB-Port Ihres Computers an. Schließen Sie anschließend ein Thermoelement an einen der Thermoelementeingänge an. Um die Installation abzuschließen, wählen Sie den USB TC-08 und den Thermoelement-Typ in der TC-08-Software aus wie im Folgenden beschrieben:

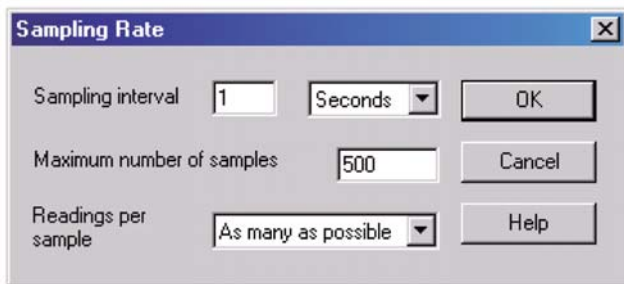
1. Starten Sie die TC-08-Rekorderanwendung.
2. Wählen Sie „New Settings“ (Neue Einstellungen) aus dem Menü „File“ (Datei).

Daraufhin wird das folgende Dialogfeld für die Aufzeichnung angezeigt:



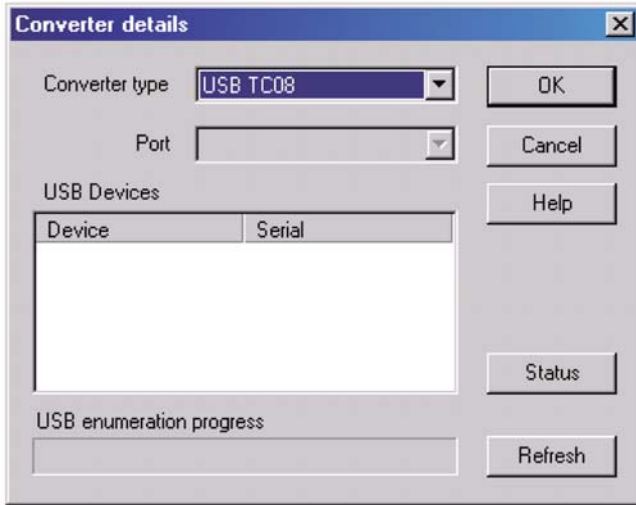
3. Klicken Sie auf die Schaltfläche „OK“.

Daraufhin wird das folgende Dialogfeld für die Abtastrate angezeigt:



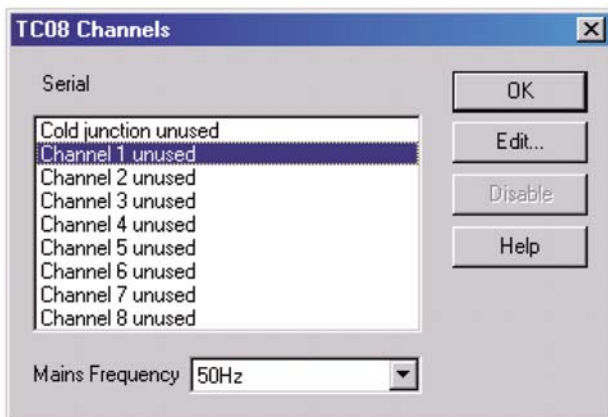
4. Klicken Sie auf die Schaltfläche „OK“.

Daraufhin wird das folgende Dialogfeld für die Messumformerdetails angezeigt:

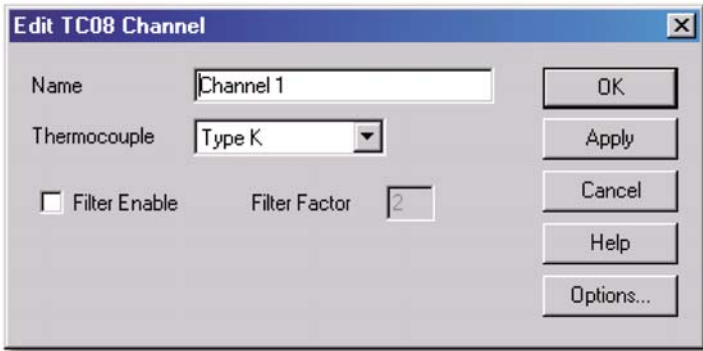


5. Wählen Sie aus der Liste der Messumformer „USB TC-08“.
Im Bereich „USB Devices“ (USB-Geräte) sollten nun Name und Seriennummer des Gerätes angezeigt werden, und die USB-Statusanzeige sollte 100% erreichen. Wenn sich die Statusanzeige nicht bewegt, trennen Sie den USB TC-08 vom Computer und schließen Sie ihn erneut an. Klicken Sie anschließend auf „Refresh“ (Aktualisieren).
6. Wählen Sie im Bereich „USB Devices“ (USB-Geräte) den USB TC-08. Wenn Sie mehr als einen USB TC-08 verwenden, überprüfen Sie die Seriennummer an der Unterseite des Gerätes, um das richtige Gerät aus der Liste zu wählen.
7. Klicken Sie auf die Schaltfläche „OK“.

Daraufhin wird das folgende Dialogfeld angezeigt:



8. Doppelklicken Sie im Kanalfenster des TC-08 auf den unverwendeten Kanal 1. Das Dialogfeld „Edit TC08 Channel“ (TC-08-Kanal bearbeiten) wird angezeigt:



9. Wählen Sie aus der Dropdownliste „Thermocouples“ (Thermoelemente) den verwendeten Thermoelement-Typ und klicken Sie auf „OK“.
Das Dialogfeld „TC-08 Channels“ (TC-08-Kanäle) wird wieder angezeigt.
10. Klicken Sie auf „OK“.

Im Überwachungsfenster wird nun Kanal 1 mit dem aktuellen Temperaturmesswert angezeigt.



1.4 Sicherheitswarnung

Wir empfehlen dringend, dass Sie die folgenden allgemeinen Sicherheitsinformationen lesen, bevor Sie das Produkt zum ersten Mal einsetzen. Wenn Sie die Geräte anders als hier angegeben einsetzen, können Sie eventuell Schutzvorrichtungen unwirksam machen. Dies kann zu Schäden an Ihrem Computer und/oder zu Verletzungen führen.

Maximaler Eingangsbereich

Der USB TC-08 ist auf die Messung von Spannungen im Bereich von ± 70 mV ausgelegt. Spannungen von mehr als ± 30 V können das Gerät dauerhaft beschädigen.

Netzspannungen

Diese Produkte sind nicht für den Betrieb mit Netzspannungen ausgelegt.

Schutzerdung

Die Masse des Produkts ist über das mitgelieferte Anschlusskabel direkt mit der Erdung Ihres Computers verbunden. Auf diese Weise werden Interferenzen so gering wie möglich gehalten. Wenn der PC (insbesondere ein Laptop) nicht geerdet ist, kann eine stabile Messung nicht gewährleistet werden. Unter Umständen kann eine manuelle Erdung des Gerätes erforderlich sein.

Achten Sie beim Anschluss der Eingangssignale an das Gerät darauf, dass keine gefährlichen Spannungen anliegen. Wenn Sie unsicher sind, überprüfen Sie mit einem Multimeter, dass keine gefährliche AC- oder DC-Spannung anliegt. Nichtbeachtung dieses Hinweises kann zu Schäden am Produkt und/oder zu Verletzungen führen.

Achten Sie darauf, dass Sie bei der Messung von Temperaturen in der Nähe von Netzspannungen mit der gebotenen Vorsicht vorgehen. Wenn ein Sensor versehentlich Teile berührt, die Netzspannungen führen, kann das Gerät und/oder Ihr Computer beschädigt werden. Weiterhin kann das Gehäuse des Computers unter Spannung stehen.

Sie sollten davon ausgehen, dass das Produkt nicht mit einer Schutzerde verbunden ist. Eine fehlerhafte Konfiguration oder das Messen von Spannungen über dem maximalen Eingangsbereich können gefährlich sein.

Reparaturen

Dieses Gerät enthält keine Teile, die vom Anwender gewartet werden können. Für Instandsetzung und Kalibrierung des Gerätes sind spezielle Prüfgeräte erforderlich. Diese Arbeiten müssen von OMEGA Engineering ausgeführt werden.

2. PRODUKTINFORMATIONEN

2.1 Technische Daten

Auflösung	20 Bits (16.25 NFR)
Thermoelement-Typen	B, E, J, K, N, R, S und T
Anzahl der Eingangskanäle	8
Wandlungszeit pro aktivem Kanal	100 ms
Wandlungszeit - Vergleichsstelle	100 ms
Unkalibrierte Genauigkeit	Die Summe von $\pm 0,2\%$ und $\pm 0,5^{\circ}\text{C}$
Eingangsendwert	± 70 mV
Gleichtaktunterdrückungs-Bereich	$\pm 7,5$ V
Überspannungsschutz	± 30 V
Eingangsimpedanz	2 MOhm
Eingangsanschlüsse	8 Thermoelement-Steckverbinder in Miniaturgröße
Ausgangsanschluss	USB-Stecker
Versorgungsspannung	Keine externe Versorgung erforderlich
Umgebungsbedingungen	0 bis 50°C , 25% bis 75% Feuchte Spezifizierte Genauigkeit von 20 bis 30°C NICHT Wasserfest

Anmerkung: Auflösung und Genauigkeit hängen von Thermoelement-Typ und Temperaturbereich ab. Die folgenden Thermoelemente und Temperaturbereiche werden vom USB TC-08 unterstützt.

	Tatsächlich messbarer Bereich	Theoretische Bereiche
Vergleichsstelle	-5°C bis 70°C	NA
Typ B	-20°C bis 1820°C (Vergleichsstelle muss $>20^{\circ}\text{C}$ sein)	-0°C bis 1820°C
Typ E	-270°C bis 910°C	-270°C bis 1000°C
Typ J	-210°C bis 1200°C	-210°C bis 1200°C
Typ K	-270°C bis 1370°C	-270°C bis 1370°C
Typ N	-270°C bis 1300°C	-270°C bis 1300°C
Typ R	-50°C bis 1760°C	-50°C bis 1760°C
Typ S	-50°C bis 1760°C	-50°C bis 1760°C
Typ T	-270°C bis 400°C	-270°C bis 400°C

2.2 So funktioniert ein Thermoelement

In einem geschlossenen elektrischen Kreis mit zwei unterschiedlichen Metallen (dem Thermoelement) fließt ein elektrischer Strom, wenn eine der beiden Übergangsstellen gegenüber der anderen erwärmt wird. Der Strom fließt, solange die beiden Übergangsstellen unterschiedliche Temperaturen haben. Der Größe und Richtung dieses Stroms ist eine Funktion der Temperaturdifferenz zwischen den Übergangsstellen sowie der thermischen Eigenschaften der beiden Metalle. Dieses Phänomen wird als Seebeck-Effekt bezeichnet.

Die beiden Leiter können aus zwei beliebigen unterschiedlichen Metallen bestehen, und wenn die eine Übergangsstelle (Messstelle) erwärmt wird, fließt ein Strom. Wenn die Positionen von Mess- und Vergleichsstelle vertauscht werden, fließt der Strom in die entgegengesetzte Richtung.

Ein Thermoelementkreis erzeugt eine messbare, niedrige Spannung, die fast direkt proportional zur Temperaturdifferenz zwischen der Messstelle (der "heißen" Seite) und der Vergleichsstelle (der "kalten" Seite) ist. Jede Änderung der Temperaturdifferenz führt zu einer Änderung der Spannung.

Anmerkung: Weitere Informationen zu Auswahl und Einsatz von Thermoelementen finden Sie auf www.omega.de.

3. TECHNISCHE REFERENZ

3.1 Einführung

Mit dem USB TC-08 werden Treiberrountinen geliefert, die Ihnen eine Einbindung des TC-08 in Ihre eigenen Programme erlaubt. Der USB TC-08-Treiber unterstützt Windows XP SP2 und Windows Vista.

Nach der Installation der Software enthält das Beispiele-Verzeichnis die Treiber und eine Reihe von Beispielen zur Verwendung des Treibers.

Der Treiber wird in Form einer Windows-DLL geliefert. Die DLL verwendet die stdcall-Standardkonventionen von C zum Aufruf von Funktionen und kann mit C, Delphi und Visual Basic eingesetzt werden. Außerdem lässt sie sich mit Programmen wie Microsoft Excel einsetzen, deren Makrosprache eine Form von Visual Basic ist.

3.2 Betriebsarten

3.2.1 Einführung

Der USB TC-08 verfügt über drei spezifische Betriebsarten für unterschiedliche Anwendungen. Die folgenden Betriebsarten werden unterstützt:

- Streamingmodus
- Einzelabfragemodus
- Legacymodus

3.2.2 Streamingmodus

Im Streamingmodus misst der USB TC-08 kontinuierlich und sendet diese Daten in einem festen Intervall, das durch die integrierte Uhr exakt gesteuert wird, an den Computer.

Das Gerät kann zwei Datensätze mit Messwerten intern puffern. Um den Verlust von Messwerten zu verhindern, dürfen andere Anwendungen den Treiber nicht länger als drei Abtastintervalle daran hindern, Messwerte vom Gerät abzurufen. Dies gilt auch für Ihr eigenes Programm, das Sie schreiben.

Um eine kontinuierliche Abfrage des Treibers zu ermöglichen, rufen Sie die Windows-Funktion `Sleep()` in allen Abfrageschleifen auf (s. Beispiel unten), um sicherzustellen, dass Ihre Anwendung nicht zuviel CPU-Zeit belegt.

Tipp: Versuchen Sie nicht, die Funktion `Sleep()` mit einer Zeit unter 50-100 Millisekunden aufzurufen, z. B. `Sleep(50)` oder `Sleep(100)`. Wenn Sie eine Windows GUI-Anwendung programmieren, bietet die `WM_TIMER`-Meldung eine gute Alternative für Abfrageschleifen.

Beispiel

Der folgende Code ist ein Auszug aus einem C-Programm, das die Verwendung des Streamingmodus mit dem USB TC-08-Treiber zeigt:

```
//=====
// Setting up and running the unit in Streaming mode
//=====

usb_tc08_set_mains(handle, 0); // use 50Hz mains noise rejection

for (channel = 0; channel < 9; channel++)
{
    // set each channel up as a type K thermocouple
    // channel 0 is the cold junction and will be enabled
    // by setting the third argument to anything other than ' '
    usb_tc08_set_channel(handle, channel, 'K');
}

// find out how fast the unit can sample in its current setup state
minimum_interval = usb_tc08_get_minimum_interval_ms(handle);

usb_tc08_run(handle, minimum_interval); // sample as fast as possible

// not required (just illustrates that the application
// can be idle while the driver collects the readings)
Sleep(10000);

// use a two dimensional array with an array of readings for each
channel
// In a real application, this would be a nested loop to regularly poll
// the unit for readings
for (channel = 0; channel < 9; channel++)
{
    no_of_readings = usb_tc08_get_temp( handle,
        &reading_buffer[channel],
        &times_buffer[channel],
        buffer_length,
        &overflows[channel],
        channel,
        0, // degrees Celsius units
        0); // do not fill missing readings
}

// finished polling, now do something with the readings
// if overflows[channel] is high, then one of the readings
// in reading_buffer[channel] has exceeded the input range
// of the USB TC-08

// only stop the unit when we've completely finished streaming
usb_tc08_stop(handle);
```

Anmerkung: Schließen Sie alle anderen Anwendungen, während Sie zeitkritische Programme zur Datenerfassung ausführen. Überprüfen Sie, dass im Windows-Scheduler während der Logging-Session keine anderen Aktivitäten geplant sind.

3.2.3 Einzelabfragemodus

Der Einzelabfragemodus ist eine Betriebsart, in der Messwerte auf Abruf erfasst und ausgegeben werden. Dazu wird die Funktion `usb_tc08_get_single` verwendet. Da die Funktion vollständig vom Timing der aufrufenden Anwendung abhängig ist, bietet sie sich ideal für Intervalle ab 1 Minute an. Wenn Sie eine schnelle Messung benötigen, verwenden Sie den Streamingmodus.

Anmerkung: Der Overhead beim Aufruf der Funktion kann durchaus erheblich sein, da es ca. 360 ms dauert, alle 9 Kanäle zu konvertieren. Dies entspricht 40 ms pro Kanal. Um diesen Overhead zu reduzieren, können Sie nicht benötigte Kanäle deaktivieren.

Beispiel

Der folgende Code ist ein Auszug aus einem C-Programm, das die Verwendung des Einzelabfragemodus mit dem USB TC-08-Treiber zeigt:

```
//=====
// Setting up and converting readings with Get Single mode
//=====

usb_tc08_set_mains (handle, 0); // use 50Hz mains noise rejection

for (channel = 0; channel < 9; channel++)
{
    // set each channel up as a type K thermocouple
    // channel 0 is the cold junction and will be enabled
    // by setting the third argument to anything other than ' '
    usb_tc08_set_channel (handle, channel, 'K');
}

// find out the approximate conversion time
// for a call to usb_tc08_get_single
minimum_interval = usb_tc08_get_minimum_interval_ms (handle);
printf("Conversion time: %d\n", minimum_interval);

// Collect 10 readings over approximately 9 minutes
last_time = GetTickCount();
for (i = 0, i < 10, i++)
{
    // do the conversion for all channels
    usb_tc08_get_single( handle,
                        value_array, // short value_array[9]
                        &overflow_flags,
                        0); // degrees Celsius units

    // print out the values
    printf("\n\nTime: %d minute(s)", i);
    for (c = 0; c < 9; c++)
    {
        // check for overflows on each channel
        // with a bitwise & comparator
        // shift the comparison bit to match the channel
        if (overflow_flags & (1 << c))
        {
            printf("\nChannel %d overflowed", c);
        }
        else // no overflow
        {
            printf("\nChannel %d: %f", c, value_array[c]);
        }
    }
}

if (i < 9)
{
    while (60000 > (GetTickCount() - last_time)) // 60000ms = 1 minute
    {
        Sleep(100); // let other applications run
    }
    last_time = GetTickCount();
}
}
```


3.2.4 Legacymodus

Der Legacymodus ist für Entwickler gedacht, die bereits Code für die serielle Version des TC-08 geschrieben haben und diesen nicht auf den Streaming- oder Einzelabfragemodus der USB-Version ändern möchten. Der Legacy-Support wird jedoch schrittweise eingestellt und schließlich nicht mehr in neueren Treibern implementiert sein. Für einen vollen Treibersupport verwenden Sie bitte die Streaming- oder Einzelabfragemodi.

Zur einfacheren Entwicklung von Code für die serielle Version und die USB-Version des TC-08 wurden alle Funktionsnamen geändert. Die Funktion `usb_tc08_legacy_set_channel` unterstützt die Einstellung Offset und Steilheit nicht mehr. Diese Informationen werden jetzt nur noch als Kalibrierungsdaten im Gerät selbst gespeichert. Die Legacy-Routinen werden Ihnen vertraut vorkommen, wenn Sie bereits mit der seriellen Version des TC-08 gearbeitet haben, zur Konvertierung von Legacy-Anwendungen sollten jedoch die folgenden Änderungen vorgenommen werden:

- Referenzieren der neuen Header-Datei
- Referenzieren der neuen Bibliotheksdatei
- Kopieren der DLL in das Verzeichnis der Anwendung
- Einstellen der Netzfrequenz
- Starten und Stoppen des Gerätes
- Speichern eines von `usb_tc08_open_unit` zurückgegebenen Handles anstelle der Schnittstellenummer

Beispiel

Der folgende Code ist ein Auszug aus einem C-Programm, das die Verwendung des Legacymodus mit dem USB TC-08-Treiber zeigt:

```
//=====
// Setting up and running the unit in Legacy mode
// This is designed to make it easier to adapt code written
// for the Serial TC08 for use with the USB TC08
//=====

usb_tc08_set_mains(handle, 0); // use 50Hz mains noise rejection

for (channel = 1; channel < 9; channel++)
{
    // set each channel up as a type K thermocouple
    // switch off filtering for all channels
    usb_tc08_legacy_set_channel(handle, channel, 'K', 0);
}

usb_tc08_legacy_run(handle);

last_cycle_no = 0;
no_of_readings = 0;
while (no_of_readings < 50) // collect 50 readings
{
    usb_tc08_legacy_get_cycle(handle, &this_cycle_no);

    if (last_cycle_no != this_cycle_no)
    {
        last_cycle_no = this_cycle_no;
        no_of_readings++;
    }

    for (channel = 1; channel < 9; channel++)
    {
        usb_tc08_legacy_get_temp( &reading[channel],
                                handle,
                                channel,
                                0);
    }
    // now do something with the readings
    // check that they have not overflowed
    // by comparing each reading with 2147483647L
    // or LONG_MAX (include limits.h)

    usb_tc08_legacy_get_cold_junction(handle, &cold_junction);
    // now do something with the cold junction temperature
}
}

usb_tc08_stop(handle);
```

3.3 Treiberroutinen

3.3.1 Einführung

Die folgende Tabelle erklärt die einzelnen USB TC-08-Routinen:

Routine	Beschreibung
Neuer USB- und serieller Modus	
usb_tc08_open_unit	Öffnet einen USB TC-08 und holt einen gültigen USB-Handle.
usb_tc08_open_unit_async	Öffnet die Einheit im asynchronen Modus.
usb_tc08_open_unit_progress	Fragt den Fortschritt der Enumeration während des asynchronen Betriebs ab.
usb_tc08_close_unit	Schließt den Handle.
usb_tc08_stop	Stoppt die Streamingausgabe des Gerätes.
usb_tc08_set_mains	Stellt den Filter zur Störunterdrückung auf 50 Hz oder 60 Hz ein.
usb_tc08_get_minimum_interval_ms	Gibt das Mindest-Abtastintervall für die aktuelle Einrichtung zurück.
usb_tc08_get_unit_info	Ruft Informationen von einem gegebenen Gerät ab und stellt sie als Struktur dar.
usb_tc08_get_formatted_info	Ruft Informationen von einem gegebenen Gerät ab und stellt sie als String dar.
usb_tc08_get_last_error	Gibt den letzten Fehler für ein spezifiziertes Gerät oder beim Öffnen eines Gerätes zurück.
Nur neuer USB-Modus	
usb_tc08_set_channel	Richtet einen USB TC-08-Kanal ein.
usb_tc08_run	Startet das Streaming des USB TC-08.
usb_tc08_get_single	Konvertiert auf Abruf die Messwerte der derzeit eingerichteten Kanäle.
usb_tc08_get_temp	Ruft im Streamingmodus die Temperaturmesswerte eines angegebenen Kanals ab.
usb_tc08_get_temp_deskew	Ruft im Streamingmodus die Temperaturmesswerte eines angegebenen Kanals mit bereinigter Zeitangabe ab.
Nur Legacymodus	
usb_tc08_legacy_run	Startet den USB TC-08 im Legacymodus.
usb_tc08_legacy_set_channel	Richtet einen USB TC-08-Kanal ein.
usb_tc08_get_temp	Ruft im Legacymodus die Temperaturmesswerte eines angegebenen Kanals ab.
usb_tc08_legacy_get_cold_junction	Ruft den Temperaturmesswert der Vergleichsstelle ab.
usb_tc08_legacy_get_driver_version	Gibt die Treiberversion zurück.
USB_tc08_legacy_&Get_Version	Gibt die Hardware-Version des USB TC-08 zurück.
usb_tc08_legacy_get_cycle	Gibt die Anzahl der bis jetzt erfassten Messwerte zurück.

3.3.2 Neuer USB-Modus und Legacymodus

3.3.2.1 usb_tc08_open_unit

```
short usb_tc08_open_unit (void)
```

Diese Routine gibt einen gültigen Handle auf den USB TC-08 zurück, wenn der Treiber diesen erfolgreich öffnen konnte. Falls die Routine nicht erfolgreich ausgeführt wurde, finden Sie im Abschnitt zu `usb_tc08_get_last_error` Erläuterungen zum Fehlercode. Wenn Sie mehrere USB TC-08 verwenden, rufen Sie diese Routine einmal für jedes an den PC angeschlossene Gerät auf. Wenn keine weiteren Geräte mehr gefunden wurden, gibt die Funktion den Wert 0 zurück. Der Treiber ist threadsicher und ermöglicht immer nur einer Anwendung den Zugriff auf ein gegebenes Gerät. Wenn die Funktion `usb_tc08_open_unit` ein Gerät nicht findet, sollten Sie daher prüfen, ob nicht bereits andere Anwendungen den USB TC-08 verwenden. Dazu gehören auch in anderen Benutzerkonten auf dem gleichen Computer ausgeführte Anwendungen, wo schnelle Benutzerwechsel unterstützt werden.

Anmerkung: Die Funktion `usb_tc08_open_unit` bietet eine einfache Möglichkeit zum Öffnen von USB TC-08-Geräten. Die Funktion blockiert jedoch den aufrufenden Thread, bis die angeschlossenen USB TC-08-Geräte vollständig aufgelistet sind. Wenn in Anwendungen mit nur einem Thread gleichzeitig andere Verarbeitungsfunktionen wie das Anzeigen einer Statusanzeige erforderlich sind, verwenden Sie die Funktion `usb_tc08_open_unit_async`.

Argumente	Keine	
Rückgabe	Short, größer 0	Der Handle des Gerätes.
	0	Es wurden keine weiteren Geräte gefunden.
	-1	Das Gerät konnte nicht geöffnet werden. Rufen Sie <code>usb_tc08_get_last_error</code> mit 0 als Handle auf, um den Fehlercode abzurufen.

Beispiel

Der folgende Code ist ein Auszug aus einem C-Programm, das das Öffnen mehrerer Geräte mit dem USB TC-08-Treiber zeigt. Die Handles der geöffneten Geräte werden für spätere Zugriffe in einem Array abgelegt:

```
//=====
// Opening multiple units
//=====

for (i = 0; (new_handle = usb_tc08_open_unit()) > 0; i++)
{
    // store the handle in an array
    handle_array[i] = new_handle;
}
no_of_units = i;

// deal with the error if there is one,
// if new_handle was zero, then there was no error
// and we reached the last available unit
if (new_handle == -1)
{
    error_code = usb_tc08_get_last_error(0);
    printf("Unit failed to open\nThe error code is %d", error_code);
    // could terminate the application here
}

//
// Start using the open units
//
```

3.3.2.2 usb_tc08_open_unit_async

```
short usb_tc08_open_unit_async (void)
```

Diese Routine beginnt mit dem Auflisten der angeschlossenen USB TC-08-Geräte im Hintergrund und kehrt sofort zurück, so dass der aufrufende Thread anderen Code ausführen kann.

Anmerkung: Der Treiber ist threadsicher und ermöglicht immer nur einer Anwendung den Zugriff auf ein gegebenes Gerät. Wenn die Funktion `usb_tc08_open_unit_async` ein Gerät nicht findet, sollten Sie daher prüfen, ob nicht bereits andere Anwendungen diesen USB TC-08 verwenden. Dazu gehören auch in anderen Benutzerkonten auf dem gleichen Computer ausgeführte Anwendungen, wo schnelle Benutzerwechsel unterstützt werden.

Argumente	Keine	
Rückgabe	1	Der Aufruf war erfolgreich.
	0	Es wurden keine weiteren Geräte gefunden.
	-1	Es ist ein Fehler aufgetreten. Rufen Sie <code>usb_tc08_get_last_error</code> mit 0 als Handle auf, um den Fehlercode abzurufen.

Beispiel

Der folgende Code ist ein Auszug aus einem C-Programm, das das Öffnen eines Gerätes mit der Funktion zum asynchronen Öffnen zeigt:

```
//=====
// Opening a unit asynchronously
//=====

// Tell the driver to start enumerating the unit in the background
// (usb_tc08_open_unit_async returns immediately)
result = usb_tc08_open_unit_async();
// handle any error conditions
if (result == -1)
{
    error_code = usb_tc08_get_last_error(0);
    printf("Unit failed to open\nThe error code is %d", error_code);

    // could terminate the application here
}
else if (result == 0)
{
    printf("No USB TC08 units found");

    // could terminate the application here
}

// No errors, so start polling usb_tc08_open_unit_progress
// continuously for its enumeration state
do
{
    result = usb_tc08_open_unit_progress(&handle, &progress);

    switch (result)
    {
        case USBTC08_PROGRESS_FAIL: // enum equates to: -1
            error_code = usb_tc08_get_last_error(0);
            printf("Unit failed to open\nThe error code is %d", error_code);

            // could terminate the application here
            break;

        case USBTC08_PROGRESS_PENDING: // enum equates to: 0
            printf("\nThe unit is %d percent enumerated", progress);
            Sleep(500); // wait for approx. half a second
            break;

        case USBTC08_PROGRESS_COMPLETE: // enum equates to: 1
            printf("\n\nThe unit with handle '%d', opened successfully",
            handle);
            break;
    }
}
while (result == USBTC08_PROGRESS_PENDING);

//
// Start using the open unit
//
```

3.3.2.3 usb_tc08_open_unit_progress

```
short usb_tc08_open_unit_progress
(
    short * handle,
    short * progress
)
```

Rufen Sie diese Funktion nach dem Aufruf von `usb_tc08_open_unit_async` auf. Rufen Sie sie wiederholt auf, um den Status des im Hintergrund ausgeführten Auflistungsprozesses zu ermitteln. Ein Beispiel für die Verwendung der Funktion finden Sie bei `usb_tc08_open_unit_async`.

Argumente	handle	(Ausgabe) Der Handle (Short, größer 0) des Gerätes, wenn die Auflistung abgeschlossen ist. Solange die Auflistung nicht abgeschlossen ist, ist der Handle immer 0.
	progress	(Ausgabe) (Optional, kann NULL übergeben) gibt eine Zahl von 0 bis 100 zurück, die angibt, zu wie viel Prozent die Auflistung eines Gerätes abgeschlossen ist.
Rückgabe	-1 USBTC08_PROGRESS_FAIL	Es ist ein Fehler aufgetreten. Rufen Sie <code>usb_tc08_get_last_error</code> mit 0 als Handle auf, um den Fehlercode abzurufen.
	0 USBTC08_PROGRESS_PENDING	Die Auflistung ist nicht abgeschlossen (rufen Sie <code>calling usb_tc08_open_unit_progress</code> weiter auf).
	1 USBTC08_PROGRESS_COMPLETE	Die Auflistung ist abgeschlossen und der Handle ist jetzt gültig.

3.3.2.4 usb_tc08_close_unit

```
short usb_tc08_close_unit
(
    short handle
)
```

Diese Routine schließt das Gerät mit dem angegebenen Handle.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
Rückgabe	0	Verwenden Sie usb_tc08_get_last_error.
	1	Gerät erfolgreich geschlossen.

Anmerkung: Wenn Sie USB TC-08-Geräte erfolgreich geöffnet haben, rufen Sie für jeden Handle `usb_tc08_close_unit` auf, bevor Sie das Programm beenden. Anderenfalls besteht die Möglichkeit, dass Sie das Gerät nicht erneut öffnen können, bevor es physikalisch vom Rechner getrennt und wieder angeschlossen wurde.

3.3.2.5 usb_tc08_stop

```
short usb_tc08_stop
(
    short handle
)
```

Diese Routine stoppt ein laufendes Gerät.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
Rückgabe	0	Ungültiger Parameter.
	1	Das Gerät hat das Streaming erfolgreich beendet.

3.3.2.6 usb_tc08_set_mains

```
short usb_tc08_set_mains
(
    short handle,
    short sixty_hertz
)
```

Diese Routine setzt die Netzstörunterdrückung des USB TC-08 auf 50 oder 60 Hz.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	sixty_hertz	Wählt zwischen 50 Hz und 60 Hz. Mit 1 wird das Gerät auf 60 Hz eingestellt, mit 0 wird das Gerät auf 50 Hz eingestellt.
Rückgabe	0	Verwenden Sie usb_tc08_get_last_error.
	1	Netzstörunterdrückung erfolgreich eingestellt.

Anmerkung: Wenn die Netzstörunterdrückung nicht korrekt eingestellt ist, ist das Gerät empfindlicher für Netzstörungen.

3.3.2.7 usb_tc08_get_minimum_interval_ms

```
long usb_tc08_get_minimum_interval_ms
(
    short handle
)
```

Diese Routine gibt das Mindest-Abtastintervall (oder das schnellste Intervall in Millisekunden) zurück, welches das Gerät in seiner aktuellen Konfiguration erreichen kann. Die Konfiguration wird durch Aufrufen von `usb_tc08_set_channel()` definiert.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	Mindestintervall	Das Mindest-Abtastintervall in Millisekunden für die aktuelle Konfiguration.

Anmerkung: Der USB TC-08 kann einen Kanal mit einer Abtastrate von 10 Messungen pro Sekunde abtasten. Das absolute Mindest-Abtastintervall mit allen 8 Kanälen und der Vergleichsstelle beträgt 900 ms. Sie müssen alle gewünschten Kanäle einrichten, bevor sie diese Routine aufrufen.

3.3.2.8 usb_tc08_get_unit_info

```
short usb_tc08_get_unit_info
(
    short handle,
    USBTC08_INFO * info
)
```

Diese Routine holt die Geräteinformationen und kopiert sie in die Struktur `USBTC08_INFO`, die in der Header-Datei `usbtc08.h` deklariert ist. Wenn Sie der Funktion "0" als Handle übergeben, ist in der Struktur nur die Treiberversion gültig, die Funktion gibt jedoch eine 1 (erfolgreich) zurück.

Argumente	handle	Spezifiziert das TC-08-Gerät.
	info	Ein Pointer auf eine Struktur, die Geräteinformationen enthält.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	1	Die Routine wurde erfolgreich ausgeführt.

Bevor Sie diese Routine aufrufen, müssen Sie dem Größelfeld Ihrer `USBTC08_INFO`-Struktur den korrekten Wert zuweisen. An einem Beispiel in C: Wenn „devinfo“ Ihre Struktur ist, verwenden Sie folgenden Code:

```
devinfo.size = sizeof(USBTC08_INFO);
usb_tc08_get_unit_info(hTC08, &devinfo);
```

3.3.2.9 usb_tc08_get_formatted_info

```
short usb_get_formatted_info
(
    short   handle,
    char *  unit_info,
    short   string_length
)
```

Diese Funktion ähnelt der Funktion `usb_get_unit_info`, die Informationen werden jedoch als formatierter String zurückgegeben. Der String enthält die folgenden Elemente, die jeweils in einer eigenen Zeile stehen: Treiberversion; Hardwareversion; Varianteninfo; Seriennummer; Kalibrierungsdatum.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	unit_info	Ein String, in dem die Geräteinformation abgelegt werden soll.
	string_length	Die Länge des zu kopierenden Strings. Sie muss mindestens 256 Zeichen betragen (USBTC08_MAX_INFO_CHARS).
Rückgaben	0	Zu viele Bytes zu kopieren. Es werden so viele volle Zeilen kopiert wie möglich.
	1	Die Routine wurde erfolgreich ausgeführt.

3.3.2.10 usb_tc08_get_last_error

```
short usb_tc08_get_last_error
(
    short handle
)
```

Diese Routine gibt den letzten Fehler des durch den Handle spezifizierten Gerätes zurück. Wenn der Funktion anstelle eines Handles 0 übergeben wird, gibt sie den letzten zum letzten Aufruf von `usb_tc08_open_unit` oder `usb_tc08_open_unit_async` gehörigen Fehler zurück.

Anmerkung: Wenn der Funktion ein ungültiger Handle übergeben wird, schlägt die Funktion fehl. Der Fehlercode kann jedoch keinem Gerät zugeordnet werden, so dass `usb_tc08_get_last_error` in diesem Fall keinen Fehlercode bereithält.

`usb_tc08_get_last_error` schlägt ebenfalls fehl, wenn der Funktion der ungültige Handle übergeben wird.

Die Fehlercodes sind im Folgenden aufgeführt und auch in der C-Header-Datei hinterlegt:

Fehlercodes für Anwender/Entwickler:

Fehler- kode	Fehler	Weitere Informationen
0	<code>USBTC08_ERROR_OK</code>	Es ist kein Fehler aufgetreten.
1	<code>USBTC08_ERROR_OS_NOT_SUPPORTED</code>	Der Treiber unterstützt Windows XP SP2 und Vista.
2	<code>USBTC08_ERROR_NO_CHANNELS_SET</code>	Es ist ein Aufruf von <code>usb_tc08_set_channel</code> erforderlich.
3	<code>USBTC08_ERROR_INVALID_PARAMETER</code>	Mindestens ein Funktionsargument war ungültig.
4	<code>USBTC08_ERROR_VARIANT_NOT_SUPPORTED</code>	Die Hardwareversion wird nicht unterstützt. Laden Sie den aktuellen Treiber herunter.
5	<code>USBTC08_ERROR_INCORRECT_MODE</code>	Es wurde eine nicht kompatible Kombination von Legacy- und Nicht-Legacy-Funktionen aufgerufen (oder <code>usb_tc08_get_single</code> wurde aufgerufen, während sich das Gerät im Streamingmodus befand).
6	<code>USBTC08_ERROR_ENUMERATION_IN</code>	<code>usb_tc08_open_unit_async</code> wurde erneut aufgerufen, während im Hintergrund bereits eine Auflistung ausgeführt wird.

Anmerkung: Weitere Einzelheiten zu Fehlercodes finden Sie im Abschnitt "Fehlersuche".

Reservierte Fehlerkodes

Fehler- kode	Fehler	Weitere Informationen
7	USBTC08_ERROR_NOT_RESPONDING	Von einem USB TC-08 wurde keine Antwort erhalten.
8	USBTC08_ERROR_FW_FAIL	Die Firmware konnte nicht heruntergeladen werden.
9	USBTC08_ERROR_CONFIG_FAIL	EEPROM fehlt oder ist defekt.
10	USBTC08_ERROR_NOT_FOUND	Aufgelistetes Gerät kann nicht gefunden werden.
11	USBTC08_ERROR_THREAD_FAIL	Eine Threading-Funktion ist fehlgeschlagen.
12	USBTC08_ERROR_PIPE_INFO_FAIL	Es konnten keine USB-Pipe-Informationen abgerufen werden.
13	USBTC08_ERROR_NOT_CALIBRATED	Es wurde kein Kalibrierdatum gefunden.
14	USBTC08_ERROR_PICOPP_TOO_OLD	Auf dem System wurde ein alter picopp.sys-Treiber gefunden.

Anmerkung: Diese reservierten Fehlerkodes sind nur für unseren technischen Support aussagefähig, werden hier jedoch mit aufgeführt, damit Entwickler in ihren Anwendungen entsprechende Hinweise anzeigen können. Weitere Einzelheiten zu Fehlerkodes finden Sie im Abschnitt "Fehlersuche".

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
Rückgabe	-1	Ungültiger Handle.
	Fehlerkode	Weitere Informationen finden Sie oben bei der Beschreibung der Fehlerkodes.

3.3.3 Nur neuer USB-Modus

3.3.3.1 usb_tc08_set_channel

```
short usb_tc08_set_channel
(
    short handle,
    short channel,
    char tc_type
)
```

Rufen Sie diese Routine einmal für jeden Kanal auf, den Sie verwenden möchten. Dies kann jederzeit nach dem Aufruf von `usb_tc08_open_unit` erfolgen. Als Standard sind alle Kanäle deaktiviert.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	channel	Gibt den Kanal an, den Sie einrichten möchten: Dieser Wert muss zwischen 0 - 8 liegen (0 = Vergleichsstelle).
	tc_type	Spezifiziert den an diesen Kanal angeschlossenen Thermoelement-Typ. Wählen Sie eines der folgenden Zeichen: 'B', 'E', 'J', 'K', 'N', 'R', 'S', 'T.' Ein Leerzeichen zwischen den Anführungszeichen deaktiviert den Kanal. Ein 'X' setzt den Kanal auf die Spannungsmessung.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	1	Die Routine wurde erfolgreich ausgeführt.

Anmerkung: Die Vergleichsstelle wird automatisch aktiviert, wenn ein Thermoelement verwendet wird. Wenn kein Kanal auf ein Thermoelement eingerichtet ist, kann die Vergleichsstelle optional aktiviert oder deaktiviert werden.

3.3.3.2 usb_tc08_run

```
long usb_tc08_run  
(  
    short handle,  
    long interval  
)
```

Diese Routine startet das Gerät mit einem in Millisekunden angegebenen Abtastintervall. Diese Routine sollte nach `usb_tc08_set_channel` aufgerufen werden.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	interval	Spezifiziert das angeforderte Abtastintervall. Sie können das kleinste für die gegebene Konfiguration zulässige Abtastintervall mit der Funktion <code>usb_tc08_get_minimum_interval_ms</code> abfragen.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	Intervall	Das tatsächliche, vom Treiber zugelassene Intervall.

3.3.3.3 usb_tc08_get_single

```
short usb_tc08_get_single
(
    short    handle,
    float *  temp,
    short *  overflow_flags,
    short    units
)
```

Vor dem Aufruf dieser Funktion müssen Sie die Kanäle einrichten. Weiterhin dürfen Sie das Gerät nicht mit `usb_tc08_run` in den Streamingmodus versetzt haben, anderenfalls würde `usb_tc08_get_single` fehlschlagen. Die Funktion konvertiert alle Messwerte auf Abruf. Weitere Einzelheiten und ein Beispiel finden Sie im Abschnitt "Einzelabfragemodus".

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	temp	Pointer auf ein Array mit der Länge [9]. Der USB TC-08 besitzt 9 Kanäle (8 Messkanäle + Vergleichsstelle). Die Messwerte werden mit dem Index der Kanalnummer im Array abgelegt, wobei 0 die Vergleichsstelle ist. Für nicht aktivierte Kanäle wird im Array QNaN abgelegt.
	overflow_flags	Pointer auf eine Variable mit einer Reihe von Bitflags, die bei einem Überlauf des entsprechenden Kanals auf High gesetzt werden. Ein Überlauf tritt auf, wenn das Eingangssignal größer ist als der Messbereich des USB TC-08. Das niederwertigste Bit (Bit 0) gibt Kanal 0 wieder (die Vergleichsstelle) und Bit 8 gibt Kanal 8 wieder (den letzten Thermoelement-Kanal). Der Überlaufstatus der einzelnen Kanäle lässt sich durch einen bitweisen Vergleich feststellen.
	units	Spezifiziert die Temperatureinheiten für die zurückgegebenen Daten: 0: USBTC08_UNITS_CENTIGRADE 1: USBTC08_UNITS_FAHRENHEIT 2: USBTC08_UNITS_KELVIN 3: USBTC08_UNITS_RANKINE
Rückgaben	0	Es ist ein Fehler aufgetreten. Verwenden Sie <code>usb_tc08_get_last_error</code> .
	1	Die Funktion wurde erfolgreich ausgeführt.

3.3.3.4 usb_tc08_get_temp

```

long usb_tc08_get_temp
(
    float * temp_buffer,
    long * times_ms_buffer,
    long buffer_length,
    short * overflow,
    short channel,
    short units,
    short fill_missing
)
    
```

Nachdem Sie den Treiber geöffnet und einige Kanäle eingerichtet haben, können Sie die Routine `usb_tc08_run` aufrufen. Der Treiber übernimmt anschließend kontinuierlich Messwerte vom USB TC-08. Verwenden Sie die Routine `usb_tc08_get_temp`, um regelmäßig Messwerte aus dem Puffer des Treibers abzurufen. Sie müssen die Funktion mindestens einmal pro Minute aufrufen, um Datenverluste zu vermeiden. Der Puffer des Treiber ist als Ringpuffer strukturiert, so dass ältere Messwerte durch neue überschrieben werden.

Für den Streamingmodus muss der Treiber die Messwerte ohne Unterbrechungen puffern können. Wenn der Treiber nicht ausreichend Prozessorzeit erhält, gehen Messwerte verloren und werden im Puffer durch den Fließkommawert QNaN ersetzt. Die häufigste Ursache hierfür ist, dass zu viele Anwendungen gleichzeitig ausgeführt werden.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	temp_buffer	Pointer auf den Speicherort der Messwerte.
	times_ms_buffer	Gibt die Zeit zurück, zu der der erste Kanal konvertiert wurde (optional.)
	Puffer_Länge	Die Länge der Datenpuffer.
	overflow	Pointer auf eine Variable, die auf 1 gesetzt wird, wenn bei einem der in temp_buffer kopierten Messwerte ein Überlauf aufgetreten ist oder 0, wenn kein Überlauf aufgetreten ist. Ein Überlauf tritt auf, wenn das Eingangssignal größer ist als der Messbereich des USB TC-08.
	channel	Spezifiziert den Kanal, dessen Temperatur ausgelesen wird.
	units	Spezifiziert die Temperatureinheiten für die zurückgegebenen Daten: 0: USBTC08_UNITS_CENTIGRADE 1: USBTC08_UNITS_FAHRENHEIT 2: USBTC08_UNITS_KELVIN 3: USBTC08_UNITS_RANKINE Spannungen werden immer in mV zurückgegeben.

	fill_missing	Wählen Sie, ob fehlende Messwerte durch QNaN oder den letzten bekannten Wert aufgefüllt werden: 0 - Fehlende Messwerte werden durch QNaNs ersetzt. 1 - Fehlende Messwerte werden durch letzten bekannten Wert ersetzt.
Rückgaben	-1	Es ist ein Fehler aufgetreten. Verwenden Sie <code>usb_tc08_get_last_error</code> .
	0	Zurzeit liegen keine abzurufenden Messwerte vor.
	> 0	Anzahl der in das Array kopierten Messwerte. (Der interne Puffer des Treibers kann noch weitere Messwerte enthalten.)

Warnung: Das Auffüllen des Puffers ist auch von der Leistung des PCs abhängig. Bei starker Auslastung des Prozessors ist das Auffüllen unter Umständen nicht immer genau.

3.3.3.5 `usb_tc08_get_temp_deskew`

```
long usb_tc08_get_temp_deskew
(
    short   handle,
    float * temp,
    long *  times,
    long   buffer_length,
    short * overflow,
    short  channel,
    short  units,
    short  fill_missing
)
```

Wie `_tc08_get_temp`, jedoch unter Berücksichtigung der kleinen Zeitunterschiede durch die Reihenfolge, in der die Kanäle konvertiert werden. Anmerkung: Sofern kein spezifischer Grund zur Verwendung von `usb_tc08_get_temp_deskewed` besteht, verwenden Sie anstelle dessen `usb_tc08_get_temp`.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	temp	Pointer auf den Speicherort für die Temperatur.
	times_ms_buffer	Gibt die exakte Zeit zurück, zu der dieser Kanal konvertiert wurde (optional.)
	Puffer_Länge	Spezifiziert die Länge des Messwertpuffers.
	overflow	Pointer auf eine Variable, die auf 1 gesetzt wird, wenn bei einem der in temp kopierten Messwerte ein Überlauf aufgetreten ist oder 0, wenn kein Überlauf aufgetreten ist. Ein Überlauf tritt auf, wenn das Eingangssignal größer ist als der Messbereich des USB TC-08.

	channel	Spezifiziert den Kanal, dessen Temperatur ausgelesen wird.
	units	Spezifiziert die Temperatureinheiten für die zurückgegebenen Daten: 0: USBTC08_UNITS_CENTIGRADE 1: USBTC08_UNITS_FAHRENHEIT 2: USBTC08_UNITS_KELVIN 3: USBTC08_UNITS_RANKINE Spannungen werden immer in mV zurückgegeben.
	fill_missing	Wählen Sie, ob fehlende Messwerte durch QNaN oder den letzten bekannten Wert aufgefüllt werden: 0 - Fehlende Messwerte werden durch QNaNs ersetzt. 1 - Fehlende Messwerte werden durch letzten bekannten Wert ersetzt.
Rückgaben	-1	Es ist ein Fehler aufgetreten. Verwenden Sie <code>usb_tc08_get_last_error</code> .
	0	Zurzeit liegen keine abzurufenden Messwerte vor.
	> 0	Anzahl der in das Array kopierten Messwerte. (Der interne Puffer des Treibers kann noch weitere Messwerte enthalten.)

3.3.4 Nur Legacymodus

3.3.4.1 usb_tc08_legacy_run

```
short usb_tc08_legacy_run  
(  
    short handle  
)
```

Diese Routine startet den Abtastungs-Thread und setzt das spezifizierte Gerät auf den Legacymodus.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
Rückgabe	0	Verwenden Sie usb_tc08_get_last_error.
	1	Legacymodus erfolgreich gestartet.

3.3.4.2 usb_tc08_legacy_set_channel

```
short usb_tc08_legacy_set_channel
(
    short handle,
    short channel,
    char tc_type,
    short filter_factor,
    short offset,
    short slope
)
```

Rufen Sie diese Routine einmal für jeden Kanal auf, den Sie verwenden möchten. Dies kann jederzeit nach dem Aufruf von `usb_tc08_open_unit` erfolgen.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	channel	Gibt den Kanal an, den Sie einrichten möchten: Dieser Wert muss zwischen 0 und 8 liegen.
	tc_type	Spezifiziert den an diesen Kanal angeschlossenen Thermoelement-Typ. Wählen Sie eines der folgenden Zeichen: 'B', 'E', 'J', 'K', 'N', 'R', 'S', 'T'. Ein Leerzeichen zwischen den Anführungszeichen deaktiviert den Kanal.
	filter_factor	Spezifiziert die Größe des Mittelungsfilters. Der Treiber aktualisiert den gefilterten Wert für den Kanal bei jedem Abruf, indem der neue Messwert in den Mittelungsfiler geschrieben wird. Der Filterkoeffizient kann auf einen beliebigen Wert zwischen 0 und 255 eingestellt werden (0 oder 1 schalten den Filter ab).
	filter_factor	Diese Parameter dienen der Abwärtskompatibilität mit dem TC-08 mit serieller Schnittstelle und haben beim USB-Modell keinen Einfluss.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	1	Der Kanal wurde erfolgreich im Legacymodus eingestellt.

Anmerkung: Rufen Sie diese Funktion nur auf, wenn Sie den USB TC-08 im Legacymodus betreiben, nachdem Sie `usb_tc08_legacy_run` aufgerufen haben.

3.3.4.3 usb_tc08_legacy_get_temp

```
short usb_tc08_legacy_get_temp
(
    long * temp,
    short handle,
    short channel,
    short filtered
)
```

Nachdem Sie den Treiber geöffnet und einige Kanäle eingerichtet haben, können Sie die Routine `usb_tc08_legacy_run` aufrufen. Der Treiber übernimmt anschließend kontinuierlich Messwerte vom USB TC-08. Temperaturen werden in Hundertstel Grad Celsius zurückgegeben, Spannungen in Mikrovolt.

Argumente	channel	Spezifiziert den Kanal, dessen Temperatur ausgelesen werden soll. Dies ist 0 für CJC, 1 für Kanal 1, 2 für Kanal 2 usw.
	temp	Pointer auf den Speicherort der Messwerte. Zur Prüfung auf einen eventuellen Überlauf sollte jeder Messwert mit 2147483647L oder <code>LONG_MAX</code> (in <code>limits.h</code> deklariert) verglichen werden. Ein Überlauf tritt auf, wenn das Eingangssignal größer ist als der Messbereich des USB TC-08.
	filtered	Spezifiziert, ob die Daten gefiltert werden oder nicht. In der Einstellung 1 wird ein Mittelungsfiler auf die Messwerte angewendet. In der Einstellung 0 werden die ungefilterten Daten in <code>temp</code> abgelegt. Der Filter hat die durch den Filterkoeffizienten definierte Filtertiefe, wie beim Aufruf von <code>usb_legacy_set_channel</code> definiert.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	1	Die Temperatur wurde erfolgreich abgerufen.

Anmerkung: Rufen Sie diese Funktion nur auf, wenn Sie den USB TC-08 im Legacymodus betreiben, nachdem Sie `usb_tc08_legacy_run` aufgerufen haben.

3.3.4.4 usb_tc08_legacy_get_cold_junction

```
short usb_tc08_legacy_get_cold_junction
(
    long * temp,
    short handle
)
```

Diese Routine ruft die Temperaturmesswerte der Vergleichsstelle ab. Diese Daten können auch über `usb_tc08_get_temp` mit `channel = 0` abgerufen werden. Temperaturen werden in Hundertstel Grad Celsius zurückgegeben. In der Regel muss die Temperatur der Vergleichsstelle nicht abgerufen werden, da der Treiber diese bei der Messung von Thermoelementen automatisch verwendet, Dieser Wert gibt jedoch Aufschluss über die Umgebungstemperatur.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	temp	Pointer auf den Speicherort für die Temperatur.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	1	Der Temperaturmesswert der Vergleichsstelle wurde erfolgreich abgerufen.

3.3.4.5 usb_tc08_legacy_get_driver_version

```
short usb_tc08_legacy_get_driver_version
(
    void
)
```

Diese Routine gibt die Treiberversion zurück. Dies ist nützlich, um festzustellen, ob der neuste Treiber verwendet wird.

Argumente	Keine
Rückgabe	Versionsnummer des Treibers

3.3.4.6 usb_tc08_legacy_get_version

```
short usb_tc08_legacy_get_version
(
    short * version,
    short handle
)
```

Diese Routine setzt die Variable 'Version' auf die Versionsnummer des gerade verwendeten USB TC-08.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	version	Pointer auf den Speicherort für die Version.
Rückgabe	0	Ungültiger Handle.
	1	Die Versionsnummer wurde erfolgreich abgerufen.

3.3.4.7 usb_tc08_legacy_get_cycle

```
short usb_tc08_legacy_get_cycle
(
    long * cycle,
    short handle
)
```

Diese Routine gibt die Anzahl der vollständigen Messzyklen an, die von einem gegebenen USB TC-08 abgerufen wurden. Der Aufruf von `usb_tc08_legacy_get_temp` führt dazu, dass die sofort die aktuellsten Messwerte für den angegebenen Kanal zurückgegeben werden. Wenn Sie Werte nur aufzeichnen möchten, wenn der Treiber neue Messwerte hat, können Sie diese Routine verwenden, um festzustellen, wie viele vollständige Messzyklen der Treiber abgerufen hat. Anschließend können Sie `usb_tc08_legacy_get_temp` aufrufen, sobald sich die Zyklusanzahl geändert hat.

Argumente	handle	Spezifiziert das USB TC-08-Gerät.
	cycle	Pointer auf den Speicherort für die Anzahl der Zyklen.
Rückgabe	0	Verwenden Sie <code>usb_tc08_get_last_error</code> .
	1	Legacy-Funktion zum Abruf der Zyklen erfolgreich ausgeführt.

Tipp: Testen Sie nicht auf eine bestimmte Anzahl von Zyklen, sondern auf eine Änderung der Zyklenanzahl, da Ihre Anwendung eventuell einen Zyklus verpasst haben kann. Ein Beispiel finden Sie im Abschnitt über den Legacymodus.

Anmerkung: Rufen Sie diese Funktion nur auf, wenn Sie den USB TC-08 im Legacymodus betreiben, nachdem Sie `usb_tc08_legacy_run` aufgerufen haben.

3.4 Programmierung

3.4.1 Einführung

Die Dokumentation enthält Beispiele für folgende Programmiersprachen:

- C und C++
- Delphi 41
- Excel 42
- LabVIEW 42
- Visual Basic
- Agilent VEE 42

Die Beispielprogramme werden im Unterverzeichnis "Examples" Ihres TC-08-Programmverzeichnisses installiert.

3.4.2 C und C++

C

Das C-Beispielprogramm ist eine generische Windows-Anwendung: es verwendet weder den Borland AppExpert noch den Microsoft AppWizard. Um das Programm zu kompilieren, erstellen Sie ein Projekt mit folgenden Dateien:

```
usb_tc08.c  
usb_tc08.rc
```

entweder `usbtc08bc.lib` (Borland 32-Bit-Anwendungen)
oder `usbtc08.lib` (Microsoft Visual C 32-Bit-Anwendungen)

Die folgenden Dateien müssen sich im gleichen Verzeichnis befinden:

```
usb_tc08.rch  
usbtc08.h  
usbtc08.dll (alle 32-Bit-Anwendungen)
```

C++

C++-Programme können auf alle Versionen des Treibers zugreifen. Wenn `usbtc08.h` in ein C++-Programm eingebunden ist, wird das `PREF1`-Makro auf Extern "C" expandiert:
###

3.4.3 Delphi

Das Unterverzeichnis "WIN" enthält einfaches Programm namens `usbtc08.dpr`, das den Treiber öffnet und Temperaturen von drei Kanälen liest. Zum Kompilieren des Programms werden folgende Dateien benötigt:

```
USBTC08FM.dfm  
USBTC08FM.PAS  
usbtc08.inc
```

Die Datei `usbtc08.inc` enthält Prozedur-Prototypen für die Treiberrouinen. Sie können diese Dateien in Ihre Anwendung einbinden.

Das Beispiel wurde mit den Delphi-Versionen 1, 2 und 3 getestet.

3.4.4 Excel

Wenn Sie Daten an Excel übertragen möchten, bietet TC-08-Software eine einfache Möglichkeit dazu.

Sollten Sie Funktionen benötigen, die die TC-08-Software nicht bietet, können Sie ein Excel-Makro schreiben, das `usbtc08.dll` aufruft, um einen Datensatz zu lesen. Die Makrosprache von Excel ähnelt Visual Basic.

Das Beispiel `usbTC0832.XLS` ruft 20 Werte von Vergleichsstelle und Kanal 1 in einem Intervall von 1 Sekunde ab und weist sie den Zellen A1..B20 zu.

3.4.5 LabVIEW

Sie können alle beschriebenen Treiberroutinen aufrufen, die spezielle LabVIEW-Zugriffsroutine bietet einen einfacheren Weg.

Um diese Routine zu verwenden, kopieren Sie `usbtc08.vi` aus dem Beispiele-Unterverzeichnis in Ihr LabVIEW 42 `user.lib`-Verzeichnis. Anschließend benötigen Sie `usb_tc08` sub-vi und einen Beispiel-Sub-VI, die die Verwendung der Dateien zeigen. Sie können jede dieser Sub-VIs für beliebige Kanäle verwenden, die Sie messen möchten. Die Sub-VI gibt für Thermoelemente eine Temperatur zurück.

3.4.6 Visual Basic

Das Beispielvezeichnis enthält die folgenden, in Visual Basic 6 erstellten Beispiele:

```
usbtc0832.vbp  
usbtc0832.bas  
usbtc0832.Frm  
usbtc0832.vbw
```

3.4.7 Agilent VEE 3

Das Beispielprogramm `usb_tc08.vee` zeigt, wie ein Datenblock vom USB TC-08 erfasst wird. `usb_tc08.vh` enthält die Prototypen

3.5 Fehlersuche

Die folgende Tabelle listet die im Abschnitt zu `usb_tc08_get_last_error` beschriebenen Fehlerkodes auf und teilt sie in verschiedene Kategorien ein, um die Fehlerbehandlung zu vereinfachen.

Fehlercode	Beschreibung	Kategorie
1	USBTC08_ERROR_OS_NOT_SUPPORTED	P*
2	USBTC08_ERROR_NO_CHANNELS_SET	P
3	USBTC08_ERROR_INVALID_PARAMETER	P
4	USBTC08_ERROR_VARIANT_NOT_SUPPORTED	P*
5	USBTC08_ERROR_INCORRECT_MODE	P
6	USBTC08_ERROR_ENUMERATION_INCOMPLETE	S
7	USBTC08_ERROR_NOT_RESPONDING	S
8	USBTC08_ERROR_FW_FAIL	S
9	USBTC08_ERROR_CONFIG_FAIL	S
10	USBTC08_ERROR_NOT_FOUND	S
11	USBTC08_ERROR_THREAD_FAIL	S
12	USBTC08_ERROR_PIPE_INFO_FAIL	S
13	USBTC08_ERROR_NOT_CALIBRATED	S
14	USBTC08_ERROR_PICOPP_TOO_OLD	S

Schlüssel

S	<p>Fehler in dieser Kategorie bedeuten, dass ein Fehler am USB TC-08 oder auf Ihrem PC aufgetreten ist. Trennen Sie den TC-08 vom PC, schließen Sie ihn wieder an und versuchen Sie es erneut. Falls dies nicht zum Erfolg führt, starten Sie Ihren PC neu.</p> <p>Sollte der Fehler damit nicht behoben sein, führen Sie folgende Schritte aus:</p> <ol style="list-style-type: none"> 1. Deinstallieren Sie die Software und starten Sie den PC erneut. 2. Installieren Sie die Software neu und starten Sie den PC erneut. 3. Sollte dies nicht zum Erfolg führen, laden Sie die neueste Software-Version von http://www.omega.com/ herunter, installieren Sie sie und starten Sie den PC neu. 4. Falls der Fehler weiterhin besteht, wenden Sie sich bitte an OMEGA Engineering.
P	<p>Fehler in dieser Kategorie werden intern in der Software behandelt. Die Bedeutung ist nur für Programmierer wichtig. Suchen Sie die Routine, die in diese Kategorie von Fehlerkodes fällt.</p>
P*	<p>Fehler in dieser Kategorie sind benutzerabhängig, und Programmierer sollten sie mit geeigneten Fehlermeldungen abfangen.</p>

3.6 Glossar

Vergleichsstelle Eine Methode zur Kompensation von Vergleichsstellenkompensation. Siehe CJC.

Common mode range. The voltage range, relative to the ground of the data logger, within which both inputs of a differential measurement must lie in order to achieve an accurate measurement.

DLL. Dynamic Link Library. Files with this file extension contain a collection of Windows functions designed to perform a specific class of operations.

Eingangsimpedanz. The resistance measured between the input terminals of a circuit.

NFR. Rauschfreie Auflösung. The effective number of bits of resolution that can be considered noise-free.

Überspannungsschutz. The maximum input voltage that can be applied without damaging the unit.

QNaN. Quiet Not a Number. In the context of the USB TC-08, QNaNs are numbers created artificially to fill in gaps in sampling. These gaps are interruptions caused by lack of available PC or laptop processor time, normally caused by too many applications being open simultaneously. QNaNs are defined in the IEEE 754-1985 ISO standard and are indeterminate, meaning no two QNaNs have the same value.

In C/C++, the `int _isnan(double)` function in the `<float.h>` header can be used to identify QNaN float representations, cast to a double first. QNaNs will not cause an error if arithmetic operations are performed on them - however, the results will remain indeterminate.

Auflösung. A value in bits, related to the number of increments of an analog input signal that can be detected by a digital measurement system. A high-resolution measurement system detects smaller signal increments than a low-resolution measurement system.

Thermoelement. A device consisting of two dissimilar metals joined together. The thermoelectric voltage developed between the two junctions is proportional to the temperature difference between the junctions.

Thermoelement Typ B. Type B thermocouples are made from platinum and rhodium and are suitable for high temperature measurements of up to 1820°C. Unusually, due to the shape of their temperature / voltage curve, type B thermocouples give the same output at 0°C as at 42°C.

Thermoelement Typ E. Type E thermocouples are made from chromel and constantan. They have a high output (68 $\mu\text{V}/^\circ\text{C}$), making them well suited to low-temperature (cryogenic) use. They are non-magnetic.

Thermoelement Typ J. Type J thermocouples are made from iron and constantan. They measure temperatures in the range -210 to +1200°C. The main application is with old equipment that can not accept the more modern thermocouple. J types should not be used above 760°C, as an abrupt magnetic transformation will cause permanent decalibration.

Thermoelement Typ K. Type K thermocouples are low-cost, general-purpose thermocouples, made from chromel and alumel, operating in the -270°C to +1370°C temperature range. Sensitivity is approx 41 $\mu\text{V}/^\circ\text{C}$.

Thermoelement Typ N. Type N thermocouples are made from nicrosil and nilsil. The high stability and resistance to high-temperature oxidation of these thermocouples make them suitable for measuring high temperatures. They are less expensive than platinum types B,R, and S and were designed to be an improved type K.

Thermoelement Typ R. Type R thermocouples are made from platinum and rhodium, and are suitable for high-temperature measurements of up to 1760°C. Low sensitivity (10 $\mu\text{V}/^\circ\text{C}$) and high cost make them unsuitable for general purpose use.

Thermoelement Typ S. Type S thermocouples are made from platinum and rhodium, and are suitable for high-temperature measurements of up to 1760°C. Low sensitivity (10 $\mu\text{V}/^\circ\text{C}$) and high cost make these thermocouples unsuitable for general purpose use. Due to their high stability, type S thermocouples are used as the standard of calibration for the melting point of gold.

Thermoelement Typ T. Type T thermocouples are made from copper and constantan, are highly accurate, and operate in the -270°C to $+400^\circ\text{C}$ temperature range.

USB. Universal Serial Bus. This is a standard port that enables you to connect external devices to PCs. A typical USB 1.1 port supports a data transfer rate of 12 Mbps (12 megabits per second), and is much faster than a COM port.

USBTC08_INFO. Diese Struktur dient dazu, Informationen von der Funktion `usb_tc08_get_unit_info` aufzunehmen und ist in der Header-Datei `usbtc08.h` definiert. Anmerkung: If the programming language you are using does not support structures, use the `usb_tc08_get_formatted_info` function.

```
typedef struct tUSBTC08Info
{
    short size;
    short DriverVersion;
    short PicoppVersion;
    short HardwareVersion;
    short Variant;
    char szSerial[ USBTC08_MAX_SERIAL_CHARS ] ;
    char szCalDate[ USBTC08_MAX_DATE_CHARS ] ;
} USBTC08_INFO, *LPUSBTC08_INFO;
```

GARANTIEBEDINGUNGEN

OMEGA garantiert, dass die Geräte frei von Material- und Verarbeitungsfehlern sind. Die Garantiedauer beträgt 13 Monate, gerechnet ab dem Verkaufsdatum. Weiterhin räumt OMEGA eine zusätzliche Kulanzzzeit von einem Monat ein, um Bearbeitungs- und Transportzeiten Rechnung zu tragen und sicherzustellen, dass diese nicht zu Lasten des Anwenders gehen.

Wenn eine Fehlfunktion auftreten sollte, muss das betroffene Instrument zur Überprüfung an OMEGA eingeschickt werden. Bitte wenden Sie sich schriftlich oder telefonisch an die Kundendienstabteilung, um eine Rückgabenummer (AR) zu erhalten. Wenn OMEGA das Instrument bei der Überprüfung als defekt befindet, wird es kostenlos ausgetauscht oder instandgesetzt. OMEGAs Garantie erstreckt sich nicht auf Defekte, die auf Handlungen des Käufers zurückzuführen sind. Dies umfasst, jedoch nicht ausschließlich, fehlerhafter Umgang mit dem Instrument, falscher Anschluss an andere Geräte, Betrieb außerhalb der spezifizierten Grenzen, fehlerhafte Reparatur oder nicht autorisierte Modifikationen. Diese Garantie ist ungültig, wenn das Instrument Anzeichen unbefugter Eingriffe zeigt oder offensichtlich aufgrund einer der folgenden Ursachen beschädigt wurde: exzessive Korrosion, zu hoher Strom, zu starke Hitze, Feuchtigkeit oder Vibrationen, falsche Spezifikationen, Einsatz in nicht dem Gerät entsprechenden Applikationen, zweckfremder Einsatz oder andere Betriebsbedingungen, die außerhalb OMEGAs Einfluss liegen. Verschleißteile sind von dieser Garantie ausgenommen. Hierzu zählen, jedoch nicht ausschließlich, Kontakte, Sicherungen oder Triacs.

OMEGA ist gerne bereit, Sie im Bezug auf Einsatz- und Verwendungsmöglichkeiten unserer Produkte zu beraten.

OMEGA übernimmt jedoch keine Haftung für Fehler, Irrtümer oder Unterlassungen sowie für Schäden, die durch den Einsatz der Geräte entsprechend der von OMEGA schriftlich oder mündlich erteilten Informationen entstehen.

OMEGA garantiert ausschließlich, dass die von OMEGA hergestellten Produkte zum Zeitpunkt des Versandes den Spezifikationen entsprechen und frei von Verarbeitungs- und Materialfehlern sind. Jegliche weitere Garantie, ob ausdrückliche oder implizit angenommene, einschließlich der der Handelsfähigkeit sowie der Eignung für einen bestimmten Zweck ist ausdrücklich ausgeschlossen. Haftungsbegrenzung: Der Anspruch des Käufers ist auf den Wert des betroffenen Produkts/Teiles begrenzt. Ein darüber hinausgehende Haftung ist ausgeschlossen, unabhängig davon, ob diese aus Vertragsbestimmungen, Garantien, Entschädigung oder anderen Rechtsgründen hergeleitet werden. Insbesondere haftet OMEGA nicht für Folgeschäden und Folgekosten.

SONDERBEDINGUNGEN: Die von OMEGA verkauften Produkte sind weder für den Einsatz in medizintechnischen Applikationen noch für den Einsatz in kerntechnischen Anlagen ausgelegt. Sollten von OMEGA verkaufte Produkte in medizintechnischen Applikationen, in kerntechnischen Einrichtungen, an Menschen oder auf andere Weise missbräuchlich oder zweckfremd eingesetzt werden, übernimmt OMEGA keinerlei Haftung. Weiterhin verpflichtet sich der Käufer, OMEGA von jeglichen Ansprüchen und Forderungen schadlos zu halten, die aus einem derartigen Einsatz der von OMEGA verkauften Produkte resultieren.

RÜCKGABEN/REPARATUREN

Bitte richten Sie alle Reparaturanforderungen und Anfragen an unsere Kundendienstabteilung. Bitte erfragen Sie vor dem Rücksenden von Produkten eine Rückgabenummer (AR), um Verzögerungen bei der Abwicklung zu vermeiden. Die Rückgabenummer muss außen auf der Verpackung sowie in der entsprechenden Korrespondenz angegeben sein.

Der Käufer ist für Versandkosten, Fracht und Versicherung sowie eine ausreichende Verpackung verantwortlich, um Beschädigungen während des Versands zu vermeiden.

Wenn es sich um einen Garantiefall handelt, halten Sie bitte die folgenden Informationen bereit, bevor Sie sich an OMEGA wenden:

1. Die Auftragsnummer, unter der das Produkt bestellt wurde.
2. Modell und Seriennummer des Produkts.
3. Reparaturanweisungen und/oder Fehlerbeschreibung.

Wenn es sich nicht um einen Garantiefall handelt, teilt Ihnen OMEGA gerne die aktuellen Preise für Reparaturen mit. Bitte halten Sie die folgenden Informationen bereit, bevor Sie sich an OMEGA wenden:

1. Die Auftragsnummer, unter der die Instandsetzung bestellt wird.
2. Modell und Seriennummer des Produkts.
3. Reparaturanweisungen und/oder Fehlerbeschreibung.

OMEGA behält sich technische Änderungen vor. Um Ihnen jederzeit den neuesten Stand der Technologie zur Verfügung stellen zu können, werden technische Verbesserungen auch ohne Modellwechsel implementiert.

OMEGA ist eine eingetragene Marke der OMEGA ENGINEERING, INC.

© Copyright OMEGA ENGINEERING, INC. Alle Rechte vorbehalten. Dieses Dokument darf ohne vorherige schriftliche Zustimmung der OMEGA ENGINEERING, INC weder vollständig noch teilweise kopiert, reproduziert, übersetzt oder in ein elektronisches Medium oder eine maschinenlesbare Form übertragen werden.

Für Ihren gesamten Bedarf der Mess- und Regeltechnik **OMEGA ... Ihr Partner**

Online-Webshop www.omega.de

TEMPERATUR

- Thermoelement-, Pt100- und Thermistorfühler, Steckverbinder, Zubehör
- Leitungen: für Thermoelemente, Pt100 und Thermistoren
- Kalibriergeräte und Eispunkt-Referenz
- Schreiber, Regler und Anzeiger
- Infrarot-Pyrometer

DRUCK UND KRAFT

- Dehnungsmessstreifen, DMS-Brücken
- Wägezellen und Druckaufnehmer
- Positions- und Wegaufnehmer
- Instrumente und Zubehör

DURCHFLUSS UND FÜLLSTAND

- Massedurchflussmesser und Durchflussrechner
- Strömungsgeschwindigkeit
- Turbinendurchflussmesser
- Summierer und Instrumente für Chargenprozesse

pH/LEITFÄHIGKEIT

- pH-Elektroden, pH-Messgeräte und Zubehör
- Tisch- und Laborgeräte
- Regler, Kalibratoren, Simulatoren und Kalibriergeräte
- Industrielle pH- und Leitfähigkeitsmessung

DATENERFASSUNG

- Kommunikations-gestützte Erfassungssysteme
- PC-Einsteckkarten
- Drahtlose Sensoren, Messumformer, Empfänger und Anzeigen
- Datenlogger, Schreiber, Drucker und Plotter
- Software zur Datenerfassung und -analyse

HEIZELEMENTE

- Heizkabel
- Heizpatronen und -streifen
- Eintaachelemente und Heizbänder
- Flexible Heizelemente
- Laborheizungen

UMWELTMESSTECHNIK

- Mess- und Regelinstrumentierung
- Refraktometer
- Pumpen und Schläuche
- Testkits für Luft, Boden und Wasser
- Industrielle Brauchwasser- und Abwasserbehandlung
- Instrumente für pH, Leitfähigkeit und gelösten Sauerstoff