

Benutzerhandbuch PCI-Zählerkarte ZP051

Heilig & Schwab GmbH
Haystraße 24
D-55566 Bad Sobernheim
Telefon: +49 (0) 67 51 / 93 12-0
Telefax: +49 (0) 67 51 / 62 07
E-mail: info@heilig-schwab.de
Internet: www.heilig-schwab.de

Diese Dokumentation darf weder als Ganzes noch in Auszügen vervielfältigt, an Dritte weitergegeben, in einer Datenbank gespeichert oder in eine andere Sprache übersetzt werden ohne schriftliche Genehmigung der Heilig & Schwab GmbH.

© Copyright 2001 - 2005 Heilig & Schwab GmbH. Alle Rechte vorbehalten.

Achte Ausgabe: Bad Sobernheim, 04. Mai 2005

Die in diesem Dokument enthaltenen Informationen können ohne vorherige Mitteilung geändert werden. Die Firma Heilig & Schwab GmbH geht damit keinerlei Verpflichtungen ein.

Heilig & Schwab GmbH übernimmt keine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden, die auf den Gebrauch oder den Inhalt dieses Benutzerhandbuches zurückzuführen sind.

Weiterhin sei darauf hingewiesen, dass die Heilig & Schwab GmbH keine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf falschen Einsatz der Hard- bzw. Software zurückzuführen sind. Layout oder Design der Hardware können ohne vorherige Mitteilung geändert werden. Die Firma Heilig & Schwab GmbH geht damit keinerlei Verpflichtungen ein.

Alle anderen in diesem Handbuch verwendeten Warenzeichen und Produktbezeichnungen sind Eigentum der entsprechenden Firmen und Hersteller. Heilig & Schwab GmbH verzichtet auf alle Besitzrechte an den genannten Warenzeichen und Produktbezeichnungen, die nicht ihr Eigentum sind.

Inhalt

1	WICHTIGE HINWEISE	5
1.1	ÜBEREINSTIMMUNGSERKLÄRUNG	5
	Hersteller- und Produktname	5
	EMV-Spezifikationen	5
1.2	VERWENDUNGSZWECK	6
1.3	LIEFERUMFANG UND SYSTEMVORAUSSETZUNG	7
	Lieferumfang	7
	Systemvoraussetzungen	7
2	INSTALLATION	8
2.1	EINBAU DER ZÄHLERKARTE	8
2.2	INSTALLATION DER TREIBER- UND ANWENDER-SOFTWARE	9
3	KONFIGURATION	10
3.1	BASIS-BIBLIOTHEKSFUNKTIONEN (HS_ZP3B.DLL)	10
	Karte initialisieren	11
	Treiber-Version abfragen	12
	Zähler einzeln auslesen	13
	Alle Zähler auslesen	14
	Referenzmodus setzen	16
	Alle Zähler latchen	17
	Zähler-Referenz-Register auslesen	18
	Zähler-Referenz-Register beschreiben	19
	Kartentyp auslesen	20
	Tastererkennung auslesen	21
3.2	ERWEITERTE BIBLIOTHEKSFUNKTIONEN (HS_ZP3X.DLL)	22
	Karte initialisieren	23
	Treiber-Version abfragen	24
	Referenz- und Latchmodus setzen	25
	Multiplikator für Zählerwert setzen	27
	Zähler einzeln auslesen	28
	Alle Zähler auslesen	29
	Latcheingang zurücksetzen	31
	Referenz zurücksetzen	32
	Zähler ununterbrochen einzeln auslesen	33
	Alle Zähler ununterbrochen auslesen	34
	Abstandscodierung konfigurieren	36
	Alle Zähler latchen	37
	Kartentyp auslesen	38
	Tastererkennung auslesen	39
3.3	DLLs EINBINDEN	40
3.4	DEMO- UND MESSPROGRAMM WINMY	40
	Software-Lizenz	40

4	TECHNISCHE DATEN	41
4.1	STECKERANORDNUNG	41
4.2	STECKERBELEGUNG (ACHSE 1 UND 2)	42
	Strom-Eingänge (Standard-Ausführung / ZP051-20)	42
	TTL-Eingänge (optional / ZP051-20-TTL)	43
	Spannungs-Eingänge (optional / ZP051-20-V)	43
4.3	STECKERBELEGUNG (ACHSE 3 / LATSCHINGÄNGE ACHSE 1 – 3)	44
4.4	PHYSIKALISCHE UND MECHANISCHE KENNWERTE	44
5	GARANTIEBEDINGUNGEN	45

1 Wichtige Hinweise

1.1 Übereinstimmungserklärung

Hersteller- und Produktname

Hersteller: Heilig & Schwab GmbH
Haystraße 24
D-55566 Bad Sobernheim

Produkt: PCI-Zählerkarte für 3 Kanäle

Modell: ZP051

EMV-Spezifikationen

Die Zählerkarte erfüllt die Bedingungen der Normen EN 55022 (Störaussendung) und EN 61000-6-2 (Störfestigkeit).

Diese Grenzwerte vermitteln einen hinreichenden Schutz gegen gefährliche elektromagnetische Strahlung für die Umgebung. Dies setzt voraus, dass das vorliegende Produkt bestimmungsgemäß montiert und eingesetzt ist. Ferner ist es erforderlich, dass alle zu der Zählerkarte führenden Leitungen fachgerecht abgeschirmt und angeschlossen sind. Die Peripheriegeräte müssen ebenfalls geschirmt und geerdet sein.



Der Betrieb mit nicht zertifizierten Personal Computern oder nicht korrekt abgeschirmten Kabeln oder nicht ordnungsgemäß montierter Zählerkarte kann zu elektromagnetischen Störungen führen.

Alle Veränderungen oder Modifikationen, die nicht ausdrücklich vom Hersteller zugelassen werden, führen zum Erlöschen der Betriebserlaubnis.

1.2 Verwendungszweck

Die Zählerkarte ZP051 ist eine PC-Einsteckkarte für einen PCI-Steckplatz. Sie dient dem direkten Anschluss von Drehgebern, Messlinealen und Tastern an den Rechner. Die Karte verfügt über drei getrennte Zählkanäle (Achse 1, Achse 2 und Achse 3). Zwei Zählkanäle können über 9-polige Sub-D-Buchsenleisten im Bracket der Zählerkarte direkt angeschlossen werden, die dritte Achse und drei Latcheingänge können über eine 20-polige Wannenstiftleiste mit der Zählerkarte verbunden werden.

Die integrierte Interpolationselektronik unterteilt die Signalperiode der Messsysteme. Die so erzeugten Zählsignale werden vorzeichenrichtig in drei Zählregistern aufsummiert und über eine Zählerzugriffssteuerung in entsprechende Register abgespeichert, die dann per PC-Software ausgelesen werden.

Die integrierte Auswertung der Referenzsignale für jede Achse getrennt ermöglicht die reproduzierbare Auswertung bezogen auf jeden beliebigen Punkt des Gebers.

Zur Verbesserung der Mess- und Betriebssicherheit werden die Eingangssignale der Messsysteme hinsichtlich der Amplitude und der Eingangsfrequenz überwacht und ein Fehlersignal ausgelöst, falls eine korrekte Signalauswertung nicht mehr gewährleistet ist. Dieses wird als Fehlerbit gespeichert und kann per Software ausgelesen bzw. gelöscht werden.

Mit der im Lieferumfang enthaltenen Treiber-Software für die verschiedenen Windows-Betriebssysteme kann die Karte schnell in Ihre Anwendung integriert werden. Diese Treiber ermöglichen Ihnen das einfache Auslesen der Zählerwerte sowie den Zugriff auf die unterschiedlichen Parameter und Betriebsarten der Karte.

1.3 Lieferumfang und Systemvoraussetzung

Lieferumfang

Im Lieferumfang zur Zählerkarte sind enthalten:

Pos.	Menge	Bezeichnung	Hinweis
1	1 St.	PCI-Zählerkarte ZP051-20	
2	1 St.	Datenträger mit <ul style="list-style-type: none">• Benutzerhandbuch• Treiber und DLLs für Windows-Betriebssysteme• Demo- und Messprogramm "WinMy"	Wird bei Rahmenaufträgen einmalig geliefert, ist aber bei Bedarf jederzeit erhältlich.

Bitte überprüfen Sie direkt nach Erhalt der Lieferung den Inhalt des Pakets.

Falls Sie Abweichungen zum oben beschriebenen Lieferumfang feststellen, setzen Sie sich bitte umgehend mit uns in Verbindung.

Tel.: +49 (0) 67 51 / 93 12-30

Systemvoraussetzungen

Für den Betrieb der PC-Zählerkarte ZP051 werden an Ihren Computer und Ihre Software die folgenden Systemanforderungen gestellt:

- Pentium-PC oder ein 100% kompatibles System
- Windows 95/98 und höher
- Ein freier PCI-Steckplatz
- VGA-Monitor

Bitte beachten Sie, dass Sie für den Einsatz der Software WinMy mindestens 8 MB Hauptspeicher und ca. 5 MB Festplattenkapazität benötigen.

2 Installation


Die Zählerkarte ZP051 darf nur von entsprechend geschultem Personal eingebaut und installiert werden.

Es wird ferner vorausgesetzt, dass vor Installation der Karte dieses Benutzerhandbuch gelesen wurde und die jeweiligen Sicherheits- und Bedienhinweise beachtet werden.

2.1 Einbau der Zählerkarte

Beachten Sie bitte im Umgang mit der Karte die allgemein üblichen Sicherheitsvorkehrungen bezüglich elektrostatischer Entladung. Im Zweifelsfalle entladen Sie sich durch Anfassen einer Erdverbindung, z.B. geerdete Gerätegehäuse, Heizkörper o.ä..

Beim Einsetzen der Zählerkarte in Ihren PC gehen Sie im Einzelnen wie folgt vor:

-  1. Schalten Sie den Rechner aus und ziehen Sie den Netzstecker.
2. Öffnen bzw. entfernen Sie das Gehäuse des Rechners.
3. Wählen Sie einen freien PCI-Steckplatz und entfernen Sie dessen Abdeckblech (Slotblech oder Bracket) an der Rückseite des Rechners.
4. Setzen Sie nun die Zählerkarte ZP051 in den Steckplatz auf dem Motherboard ein. - Achten Sie darauf, dass die Karte gerade sitzt und keine benachbarte Karte berührt wird. Die Signal-Anschlussbuchsen müssen außen frei zugänglich sein.
5. Befestigen Sie nun die Zählerkarte mit einer Schraube an der hierfür vorgesehenen Bohrung in der Rückwand des Rechners.
6. Wenn nötig, schließen Sie nun die dritte Achse oder die Latcheingänge über die Wannentiftleiste auf der Leiterkarte der Zählerkarte ZP051 an.
7. Schließen Sie das Rechnergehäuse ordnungsgemäß.
8. Schließen Sie die Messsysteme (erste und zweite Achse) an die Signal-Anschlussbuchsen der Zählerkarte ZP051 an.

Die Zählerkarte ist jetzt betriebsfertig montiert. Der Rechner kann eingeschaltet werden.

Zum Betreiben der Zählerkarte benötigen Sie jetzt nur noch die auf dem Datenträger mitgelieferte Treiber-Software für Ihr Betriebssystem. Die Installation der Software wird in Kapitel "Installation der Treiber- und Anwendersoftware" beschrieben.

2.2 Installation der Treiber- und Anwender-Software

Die Treiber und DLLs für die Windows-Betriebssysteme befinden sich auf dem mitgelieferten Datenträger.

Windows 95/98 und höher erkennt nach dem Einbau der Zählerkarte diese automatisch und fordert die Installation des entsprechenden Treibers. Nach dem Einlegen des Datenträgers erfolgt die Installation automatisch.

Bei Windows NT wird die Treibersoftware durch Starten des auf dem Datenträger beigefügten Setup-Programms installiert.

3 Konfiguration

Für den Zugriff auf den Treiber der Zählerkarte werden zwei DLLs mitgeliefert. Die DLL "HS_ZP3B.DLL" mit Basis-Bibliotheksfunktionen und die DLL "HS_ZP3X.DLL" mit erweiterten Bibliotheksfunktionen.

3.1 Basis-Bibliotheksfunktionen (HS_ZP3B.DLL)

Die Basis-DLL "HS_ZP3B.DLL" enthält die folgenden Aufrufe:

DLL Funktionsaufruf	Bedeutung
HS_ZP3B_Init	Lädt Treiber und DLL und prüft, ob die Zählerkarte im PC vorhanden ist. Initialisiert die Zählerkarte.
HS_ZP3B_GetVersion	Überprüft die Versionen von Treiber und DLL.
HS_ZP3B_ReadCounter	Liest den Zählerwert und den Status einer Achse (eines Zählers) der Zählerkarte aus.
HS_ZP3B_ReadAllCounter	Liest den Zählerwert und den Status aller Achsen der Zählerkarte aus.
HS_ZP3B_SetRefMode	Setzt den Referenzmodus einer Achse.
HS_ZP3B_StopAllCounter	Latcht alle Zählerwerte aller Zählerkarten.
HS_ZP3B_ReadRefReg	Liest das Zähler-Referenz-Register einer Achse aus.
HS_ZP3B_WriteRefReg	Schreibt in das Zähler-Referenz-Register einer Achse.
HS_ZP3B_ReadCardType	Liest den Kartentyp aus.
HS_ZP3B_ProbeAvail	Liest das Ergebnis der Tastererkennung aus.

Karte initialisieren

Funktionsname:	HS_ZP3B_Init
Syntax:	Visual C++: long HS_ZP3B_Init (long CardNo) Delphi: HS_ZP3B_Init (CardNo :longint) :longint
Beschreibung:	Lädt Treiber und DLL und prüft, ob die Zählerkarte im PC vorhanden ist. Initialisiert die Zählerkarte.
Übergabeparameter:	CardNo Logische Kartennummer (1...15)
Rückgabewert:	0 OK -1 Fehler

Treiber-Version abfragen

Funktionsname:	HS_ZP3B_GetVersion
Syntax:	Visual C++: long HS_ZP3B_GetVersion (char* DriverVersion, char* DllVersion)
	Delphi: HS_ZP3B_GetVersion (DriverVersion :pchar, DllVersion :pchar) :longint
Beschreibung:	Überprüft die Versionen von Treiber und DLL.
Übergabe- parameter:	DriverVersion: Zeiger auf den String, in den die Treiberversion eingetragen wird. DllVersion: Zeiger auf den String, in den die DLL-Version eingetragen wird. Die Strings sollten eine Mindestgröße von 80 Zeichen haben.
Rückgabewert:	0 OK -1 Fehler

Zähler einzeln auslesen

Funktionsname: HS_ZP3B_ReadCounter

Syntax: **Visual C++:** long HS_ZP3B_ReadCounter (long CardNo,
long CounterNo,
long *CountVal,
long *CountStat)

Delphi: HS_ZP3B_ReadCounter (CardNo: :longint;
CounterNo :longint;
var CountVal :longint;
var CountStat :longint)

:longint

Beschreibung: Liest den Zählerwert und den Status einer Achse (eines Zählers) der Zählerkarte aus.

**Übergabe-
parameter:**

CardNo Logische Kartenummer (1...15)

CounterNo Zählernummer (1...3)

CountVal Zeiger auf Variable, die den Zählerwert aufnimmt.

CountStat Zeiger auf Variable, die den Zählerstatus aufnimmt.

01h: Zustand Latcheingang

02h: Fehler

04h: Referenzpunkt überfahren

08h: Änderung Zustand Latcheingang

Rückgabewert: **0** OK

-1 Fehler

Alle Zähler auslesen

Funktionsname: HS_ZP3B_ReadAllCounter

Syntax:	Visual C++:	long HS_ZP3B_ReadAllCounter (long CardNo, long *Count1Val, long *Count1Stat, long *Count2Val, long *Count2Stat, long *Count3Val, long *Count3Stat);
----------------	--------------------	---

[illegible]

:longint

Beschreibung: Liest den Zählerwert und den Status aller Achsen der Zählerkarte aus.

Übergabe- parameter:	CardNo	Logische Kartennummer (1...15)
---------------------------------	---------------	--------------------------------

Count1Val Zeiger auf Variable, die den Zählerwert von Zähler 1 aufnimmt.

Count1Stat Zeiger auf Variable, die den Zählerstatus von Zähler 1 aufnimmt.

01h:	Zustand Latcheingang
02h:	Fehler
04h:	Referenzpunkt überfahren
08h:	Änderung Zustand Latcheingang

Count2Val Zeiger auf Variable, die den Zählerwert von Zähler 2 aufnimmt.

Count2Stat Zeiger auf Variable, die den Zählerstatus von Zähler 2 aufnimmt.

01h:	Zustand Latcheingang
02h:	Fehler
04h:	Referenzpunkt überfahren
08h:	Änderung Zustand Latcheingang

Alle Zähler auslesen (Fortsetzung)

Übergabe- parameter:	Count3Val	Zeiger auf Variable, die den Zählerwert von Zähler 3 aufnimmt.
	Count3Stat	Zeiger auf Variable, die den Zählerstatus von Zähler 3 aufnimmt.
		01h: Zustand Latcheingang
		02h: Fehler
		04h: Referenzpunkt überfahren
Rückgabewert:		08h: Änderung Zustand Latcheingang
	0	OK
	-1	Fehler

Referenzmodus setzen

Funktionsname: HS_ZP3B_SetRefMode

Syntax:	Visual C++: long HS_ZP3B_SetRefMode (long CardNo, long CounterNo, long RefMode);
----------------	---

```

Delphi:      HS_ZP3B_SetMode (CardNo   :longint;
                               CounterNo :longint;
                               RefMode    :longint)
                               :longint

```

Beschreibung: Setzt den Referenzmodus einer Achse.

Übergabe- parameter:	CardNo	Logische Kartennummer (1...15)
---------------------------------	---------------	--------------------------------

CounterNo Zählernummer (1...3)

RefMode Referenzmodus

- 00h: Ohne Referenz (Default)
- 01h: Einfach-Referenz
- 02h: Mehrfach-Referenz

Rückgabewert:	0	OK
	-1	Fehler

Alle Zähler latches

Funktionsname:	HS_ZP3B_StopAllCounter
Syntax:	Visual C++: long HS_ZP3B_StopAllCounter (long CardNo); Delphi: HS_ZP3B_StopAllCounter (CardNo :longint) :longint
Beschreibung:	Latcht alle Zählerwerte aller Zählerkarten.
Übergabe- parameter:	CardNo Logische Kartennummer (1...15)
Rückgabewert:	0 OK -1 Fehler

Zähler-Referenz-Register auslesen

Funktionsname: HS_ZP3B_ReadRefReg

Syntax:	Visual C++: long HS_ZP3B_ReadRefReg (long CardNo, long CounterNo, long *RefVal);
----------------	---

Delphi: HS_ZP3B_ReadRefReg (CardNo :longint;
 CounterNo :longint;
 var RefVal :longint)

```
:longint
```

Beschreibung:	Liest das Zähler-Referenz-Register einer Achse aus.
----------------------	---

Übergabe- parameter:	CardNo	Logische Kartennummer (1...15)
---------------------------------	---------------	--------------------------------

CounterNo Zählernummer (1...3)

RefVal Zeiger auf Variable, die den Referenzwert aufnimmt.

Rückgabewert: 0 OK

-1 Fehler

Zähler-Referenz-Register beschreiben

Funktionsname: HS_ZP3B_WriteRefReg

Syntax:	Visual C++: long HS_ZP3B_WriteRefReg (long CardNo, long CounterNo, long RefVal);
----------------	---

Delphi: HS_ZP3B_WriteRefReg (CardNo :longint;
CounterNo :longint;
RefVal :longint)

```
:longint
```

Beschreibung: Schreibt in das Zähler-Referenz-Register einer Achse.

Übergabe- parameter:	CardNo	Logische Kartennummer (1...15)
---------------------------------	---------------	--------------------------------

CounterNo Zählernummer (1...3)

RefVal Referenzwert

Rückgabewert: 0 OK

-1 Fehler

Kartentyp auslesen

Funktionsname: HS_ZP3B_ReadCardType

Syntax:

Visual C++: long HS_ZP3B_ReadCardType (long CardNo
long *CardType);

Delphi: HS_ZP3B_ReadCardType (CardNo :longint
var CardType :longint)
:longint

Beschreibung: Liest den Kartentyp aus.

**Übergabe-
parameter:**

CardNo Logische Kartennummer (1...15)

CardType Zeiger auf Variable, die den Kartentyp aufnimmt.

01h: 11 μ A_{SS}-Eingänge
02h: 1 V_{SS}-Eingänge
03h: TTL-Eingänge

Rückgabewert:

0 OK

-1 Fehler

Tastererkennung auslesen

Funktionsname:	HS_ZP3B_ProbeAvail		
Syntax:	Visual C++:	long HS_ZP3B_ProbeAvail	(long CardNo long *ProbeAvail);
	Delphi:	HS_ZP3B_ProbeAvail	(CardNo :longint var ProbeAvail :longint) :longint
Beschreibung:	Liest das Ergebnis der Tastererkennung aus.		
Übergabe- parameter:	CardNo	Logische Kartennummer (1...15)	
	ProbeAvail	Zeiger auf Variable, die das Ergebnis der Tastererkennung aufnimmt.	
		00h:	kein Taster angeschlossen
		01h:	Taster angeschlossen
Rückgabewert:	0	OK	
	-1	Fehler	

3.2 Erweiterte Bibliotheksfunktionen (HS_ZP3X.DLL)

Die erweiterten Bibliotheksfunktionen der DLL "HS_ZP3X.DLL" enthalten die folgenden Aufrufe:

DLL Funktionsaufruf	Bedeutung
HS_ZP3X_Initialize	Lädt den Treiber und prüft, ob die Zählerkarte im PC vorhanden ist. Initialisiert die Zählerkarte.
HS_ZP3X_GetVersion	Überprüft die Versionen von Treiber und erweiterter DLL.
HS_ZP3X_SetMode	Setzt den Referenz- und den Latchmodus einer Achse (eines Zählers).
HS_ZP3X_SetCounterMul	Setzt den Multiplikator für eine Achse.
HS_ZP3X_GetCounter	Liest den Zählerwert und den Status einer Achse der Zählerkarte aus.
HS_ZP3X_GetAllCounter	Liest den Zählerwert und den Status aller Achsen der Zählerkarte aus.
HS_ZP3X_ResetCounterStop	Setzt den Steuereingang einer Achse der Zählerkarte zurück.
HS_ZP3X_ResetRef	Setzt die Referenz der Zählerkarte zurück.
HS_ZP3X_GetCounterCurrent	Liest den Zählerwert und den Status einer Achse ununterbrochen aus.
HS_ZP3X_GetAllCounterCurrent	Liest den Zählerwert und den Status aller Achsen einer Zählerkarte ununterbrochen aus.
HS_ZP3X_SetDistCode	Konfiguriert die Parameter für abstandscodierte Maßstäbe.
HS_ZP3X_StopAllCounter	Latcht alle Zählerwerte aller Zählerkarten.
HS_ZP3X_ReadCardType	Liest den Kartentyp aus.
HS_ZP3X_ProbeAvail	Liest das Ergebnis der Taster-Erkennung aus.

Karte initialisieren

Funktionsname:	HS_ZP3X_Initialize
Syntax:	Visual C++: long HS_ZP3X_Initialize (long CardNo) Delphi: HS_ZP3X_Initialize (CardNo :longint) :longint
Beschreibung:	Lädt den Treiber und prüft, ob die Zählerkarte im PC vorhanden ist. Initialisiert die Zählerkarte und setzt die Referenz- und Latch-Parameter auf die Defaulteinstellung .
Übergabe-parameter:	CardNo Logische Kartennummer (1...15)
Rückgabewert:	0 OK -1 Fehler

Treiber-Version abfragen

Funktionsname:	HS_ZP3X_GetVersion
Syntax:	Visual C++: long HS_ZP3X_GetVersion (char* DriverVersion, char* DllVersion)
	Delphi: HS_ZP3X_GetVersion (DriverVersion :pchar; DllVersion :pchar) :longint
Beschreibung:	Überprüft die Versionen von Treiber und erweiterter DLL.
Übergabe- parameter:	DriverVersion Zeiger auf den String in den die Treiberversion eingetragen wird. DllVersion Zeiger auf den String in den die DLL-Version eingetragen wird. Die Strings sollten eine Mindestgröße von 80 Zeichen haben.
Rückgabewert:	0 OK -1 Fehler

Referenz- und Latchmodus setzen

Funktionsname: HS_ZP3X_SetMode

Syntax:	Visual C++: long HS_ZP3X_SetMode (long CardNo, long CounterNo, long RefMode, long StopMode);
----------------	---

Delphi: HS_ZP3X_SetMode (CardNo :longint;
CounterNo :longint;
RefMode :longint;
StopMode :longint)

```
:longint
```

Beschreibung: Setzt den Referenz- und den Latchmodus einer Achse (eines Zählers).
Bitte beachten Sie auch die untenstehende Funktionserläuterung.

Übergabe-	CardNo	Logische Kartennummer (1...15)
------------------	---------------	--------------------------------

CounterNo Zählernummer (1...3)

RefMode Referenzmodus

00h:	Ohne Referenz (Default)
01h:	Einfach-Referenz
02h:	Mehrfach-Referenz
03h:	Abstandscodierte Referenz

StopMode Latchmodus

00h:	Ohne Latch (Default)
01h:	Manueller Latch
02h:	Automatischer Latch
00h:	Latch bei steigender Flanke (Default)
10h:	Latch bei fallender Flanke
20h:	Latch bei steigender und fallender Flanke

Rückgabewert: 0 OK

-1 Fehler

Referenz- und Latchmodus setzen (Fortsetzung)

Funktionserläuterungen

Für die praktische Anwendung ist es notwendig, die Funktionalität der Referenzauswertung und der Zähler-Latch-Funktion zu erläutern.

Referenzauswertung

- Bei **zyklischer (mehrfacher) Referenzauswertung** wird der Zählerwert bei jedem Überfahren des Referenzpunktes wieder auf 0 gesetzt. Dies wird häufig bei Drehgebern verwendet, wodurch automatisch die Winkel auf 0-360° genormt werden.
- Die **einmalige Referenzauswertung** setzt den Zähler lediglich beim erstmaligen Überfahren des Referenzpunktes auf 0. Weitere Referenzsignale werden dann von der Karte ignoriert. Solange in diesem Modus kein Referenzsignal gesetzt ist, stellt der Zähler keinen definierten Wert zur Verfügung.
- Bei **abstandscodierter Referenzauswertung** wird die Referenz nach kurzem Fahrweg erkannt und ausgewertet. Weitere Referenzsignale werden dann von der Karte ignoriert. Solange in diesem Modus kein Referenzsignal gesetzt ist, stellt der Zähler keinen definierten Wert zur Verfügung.

Zähler-Latch-Funktion

- **Automatische Zähler-Latch-Freigabe** bedeutet, dass die Zählerwertausgabe solange blockiert wird, wie das Latchsignal anliegt. Sobald dieses weggeschaltet wird, ist die Ausgabe wieder automatisch freigegeben. Der Zustand des Latch-Registers wird beim Auslesen der Zählerwerte übermittelt und kann so für die weitere Auswertung verwendet werden. Eine typische Anwendung hierfür ist die Kantenantastung mittels eines Kantensensors.
- Bei der **manuellen Zähler-Latch-Freigabe** bleibt die Zählerwertausgabe auch dann noch blockiert, wenn das Latchsignal nicht mehr anliegt. Diese muss dann explizit mit dem "HS_ZP3X_ResetCounterStop-Befehl" für jede Achse getrennt freigegeben werden.

Multiplikator für Zählerwert setzen

Funktionsname: HS_ZP3X_SetCounterMul

Syntax: **Visual C++:** long HS_ZP3X_SetCounterMul (long CardNo,
long CounterNo,
long Multiplier)

Delphi: HS_ZP3X_SetCounterMul (CardNo :longint;
CounterNo :longint;
Multiplier :longint)

:longint

Beschreibung: Setzt den Multiplikator für eine Achse.

Mit Hilfe des Multiplikators kann auch einfach die Zählrichtung gedreht werden. Der Multiplikator ist dann auf -1 zu setzen.

**Übergabe-
parameter:** **CardNo** Logische Kartennummer (1...15)

CounterNo Zählernummer (1...3)

Multiplier Multiplikator

Rückgabewert: **0** OK

-1 Fehler

Zähler einzeln auslesen

Funktionsname: HS_ZP3X_GetCounter

Syntax: **Visual C++:** long HS_ZP3X_GetCounter (long CardNo,
long CounterNo,
long *Count,
long *Status)

Delphi: HS_ZP3X_GetCounter (CardNo :longint;
CounterNo :longint;
var Count :longint;
var Status :longint)

:longint;

Beschreibung: Liest den Zählerwert und den Status einer Achse der Zählerkarte aus. Der Zählerwert wird mit dem Referenzwert und dem Multiplikator verrechnet.

Hinweis: Wenn der Latcheingang aktiv ist (siehe auch Funktion "Referenz- und Latchmodus setzen"), wird der Zählerwert eingefroren.

**Übergabe-
parameter:**

CardNo Logische Kartenummer (1...15)

CounterNo Zählernummer (1...3)

Count Zeiger auf Variable, die den Zählerwert aufnimmt.

Status Zeiger auf Variable, die den Zählerstatus aufnimmt.

01h: Fehler
02h: Referenzpunkt bereits überfahren
04h: Zähler gelatcht
08h: Zustand Latcheingang
10h: Änderung Zustand Latcheingang

Rückgabewert: **0** OK

-1 Fehler

Alle Zähler auslesen

Funktionsname: HS_ZP3X_GetAllCounter

Syntax:	Visual C++: long HS_ZP3X_GetAllCounter (long CardNo, long *Count1, long *Status1, long *Count2, long *Status2, long *Count3, long *Status3)
----------------	--

Delphi:

```
HS_ZP3X_GetAllCounter (CardNo :longint;  
var Count1 :longint;  
var Status1 :longint;  
var Count2 :longint;  
var Status2 :longint;  
var Count3 :longint;  
var Status3 :longint)
```

```
:longint;
```

Beschreibung:	Liest den Zählerwert und den Status aller Achsen der Zählerkarte aus. Der Zählerwert wird mit dem Referenzwert und dem Multiplikator verrechnet.
----------------------	--

Hinweis: Wenn der Latcheingang aktiv ist (siehe auch Funktion "Referenz- und Latchmodus setzen"), wird der Zählerwert eingefroren.

Übergabe- parameter:	CardNo	Logische Kartennummer (1...15)
---------------------------------	---------------	--------------------------------

Count1	Zeiger auf Variable, die den Zählerwert von Zähler 1 aufnimmt.
---------------	--

Status1 Zeiger auf Variable, die den Zählerstatus von Zähler 1 aufnimmt.

01h:	Fehler
02h:	Referenzpunkt bereits überfahren
04h:	Zähler gelatcht
08h:	Zustand Latcheingang
10h:	Änderung Zustand Latcheingang

Alle Zähler auslesen (Fortsetzung)

Übergabe- parameter:	Count2	Zeiger auf Variable, die den Zählerwert von Zähler 2 aufnimmt.
	Status2	Zeiger auf Variable, die den Zählerstatus von Zähler 2 aufnimmt.
		01h: Fehler
		02h: Referenzpunkt bereits überfahren
		04h: Zähler gelatcht
		08h: Zustand Latcheingang
		10h: Änderung Zustand Latcheingang
	Count3	Zeiger auf Variable, die den Zählerwert von Zähler 3 aufnimmt.
	Status3	Zeiger auf Variable, die den Zählerstatus von Zähler 3 aufnimmt.
		01h: Fehler
		02h: Referenzpunkt bereits überfahren
		04h: Zähler gelatcht
		08h: Zustand Latcheingang
		10h: Änderung Zustand Latcheingang
Rückgabewert:	0	OK
	-1	Fehler

Latcheingang zurücksetzen

Funktionsname:	HS_ZP3X_ResetCounterStop
Syntax:	Visual C++: long HS_ZP3X_ResetCounterStop (long CardNo, long CounterNo) Delphi: HS_ZP3X_ResetCounterStop (CardNo :longint; CounterNo :longint) :longint;
Beschreibung:	Setzt den Steuereingang einer Achse der Zählerkarte zurück. Der Zähler zählt danach sofort wieder weiter.
Übergabe- parameter:	CardNo Logische Kartennummer (1...15) CounterNo Zählernummer (1...3)
Rückgabewert:	0 OK -1 Fehler
Hinweis:	Die Freigabe muss getrennt für jede Achse erfolgen.

Referenz zurücksetzen

Funktionsname:	HS_ZP3X_ResetRef
Syntax:	Visual C++: long HS_ZP3X_ResetRef (long CardNo, long CounterNo); Delphi: HS_ZP3X_ResetRef (CardNo :longint; CounterNo :longint) :longint;
Beschreibung:	Setzt die Referenz der Zählerkarte zurück. Der Referenzpunkt muss erneut überfahren werden.
Übergabe- parameter:	CardNo Logische Kartennummer (1...15) CounterNo Zählernummer (1...3)
Rückgabewert:	0 OK -1 Fehler

Zähler ununterbrochen einzeln auslesen

Funktionsname: HS_ZP3X_GetCounterCurrent

Syntax: **Visual C++:** long HS_ZP3X_GetCounterCurrent (long CardNo,
long CounterNo,
long *Count,
long *Status);

Delphi: HS_ZP3X_GetCounterCurrent (CardNo :longint;
CounterNo :longint;
var Count :longint;
var Status :longint)
:longint;

Beschreibung: Liest den Zählerwert und den Status einer Achse ununterbrochen aus. Der Zählerwert wird mit dem Referenzwert und dem Multiplikator verrechnet.

Übergabeparameter: **CardNo** Logische Kartenummer (1...15)

CounterNo Zählernummer (1...3)

Count Zeiger auf Variable, die den Zählerwert aufnimmt.

Status Zeiger auf Variable, die den Zählerstatus aufnimmt.

01h: Fehler
02h: Referenzpunkt bereits überfahren
04h: Zähler gelatcht
08h: Zustand Latcheingang
10h: Änderung Zustand Latcheingang

Rückgabewert: **0** OK
-1 Fehler

Alle Zähler ununterbrochen auslesen

Funktionsname: HS_ZP3X_GetAllCounterCurrent

Syntax: **Visual C++:** long HS_ZP3X_GetAllCounterCurrent (long CardNo,
long *Count1,
long *Status1,
long *Count2,
long *Status2,
long *Count3,
long *Status3);

Delphi: HS_ZP3X_GetAllCounterCurrent (CardNo :longint;
var Count1 :longint;
var Status1 :longint;
var Count2 :longint;
var Status2 :longint;
var Count3 :longint;
var Status3 :longint)

:longint;

Beschreibung: Liest den Zählerwert und den Status aller Achsen einer Zählerkarte ununterbrochen aus. Der Zählerwert wird mit dem Referenzwert und dem Multiplikator verrechnet.

**Übergabe-
parameter:**

CardNo Logische Kartenummer (1...15)

Count1 Zeiger auf Variable, die den Zählerwert von Zähler 1 aufnimmt.

Status1 Zeiger auf Variable, die den Zählerstatus von Zähler 1 aufnimmt.

01h: Fehler
02h: Referenzpunkt bereits überfahren
04h: Zähler gelatcht
08h: Zustand Latcheingang
10h: Änderung Zustand Latcheingang

Count2 Zeiger auf Variable, die den Zählerwert von Zähler 2 aufnimmt.

Status2 Zeiger auf Variable, die den Zählerstatus von Zähler 2 aufnimmt.

01h: Fehler
02h: Referenzpunkt bereits überfahren
04h: Zähler gelatcht
08h: Zustand Latcheingang
10h: Änderung Zustand Latcheingang

Alle Zähler ununterbrochen auslesen (Fortsetzung)

Übergabe- parameter:	Count3	Zeiger auf Variable, die den Zählerwert von Zähler 3 aufnimmt.
	Status3	Zeiger auf Variable, die den Zählerstatus von Zähler 3 aufnimmt. 01h: Fehler 02h: Referenzpunkt bereits überfahren 04h: Zähler gelatcht 08h: Zustand Latcheingang 10h: Änderung Zustand Latcheingang
Rückgabewert:	0	OK
	-1	Fehler

Abstandscodierung konfigurieren

Funktionsname: HS_ZP3X_SetDistCode

Syntax: **Visual C++:** long HS_ZP3X_SetCistCode (long CardNo,
long CounterNo,
long Distance,
long Pitch,
long Offset,
long Direction);

Delphi: HS_ZP3X_SetDistCode (CardNo :longint;
CounterNo :longint;
Distance :longint;
Pitch :longint;
Offset :longint;
Direction :longint)
:longint

Beschreibung: Konfiguriert die Parameter für abstandscodierte Maßstäbe.

**Übergabe-
parameter:** **CardNo** Logische Kartenummer (1...15)

CounterNo Zählernummer (1...3)

Distance Grundabstand der Referenzmarken in mm

Pitch Signalperiode in μm

Offset Maßstabsanfang in mm

Direction Grundzählrichtung des Maßstabs

00h: positiv (Default), z. B. Acu-Rite
01h: negativ, z. B. Heidenhain

Rückgabewert: **0** OK

-1 Fehler

Hinweis: Folgende Reihenfolge der Funktionsaufrufe muss eingehalten werden:

HS_ZP3X_Initialize(...)
HS_ZP3X_SetDistCode(...)
HS_ZP3X_SetMode(...)

Soll mit Default-Werten (Distance=20mm; Pitch=20 μm ; Offset=0mm) gearbeitet werden, kann der Aufruf "HS_ZP3X_SetDistCode(...)" entfallen.

Alle Zähler latches

Funktionsname:	HS_ZP3X_StopAllCounter
Syntax:	Visual C++: long HS_ZP3X_StopAllCounter (long CardNo); Delphi: HS_ZP3X_StopAllCounter (CardNo :longint) :longint
Beschreibung:	Latcht alle Zählerwerte aller Zählerkarten.
Übergabe- parameter:	CardNo Logische Kartennummer (1...15)
Rückgabewert:	0 OK -1 Fehler

Kartentyp auslesen

Funktionsname: HS_ZP3X_ReadCardType

Syntax: **Visual C++:** long HS_ZP3X_ReadCardType (long CardNo
long *CardType);

Delphi: HS_ZP3X_ReadCardType (CardNo :longint
var CardType :longint)
:longint

Beschreibung: Liest den Kartentyp aus.

**Übergabe-
parameter:** **CardNo** Logische Kartenummer (1...15)

CardType Zeiger auf Variable, die den Kartentyp aufnimmt.

01h: 11 μ A_{SS}-Eingänge

02h: 1 V_{SS}-Eingänge

03h: TTL-Eingänge

Rückgabewert: **0** OK

-1 Fehler

Tastererkennung auslesen

Funktionsname:	HS_ZP3X_ProbeAvail		
Syntax:	Visual C++:	long HS_ZP3X_ProbeAvail	(long CardNo long *ProbeAvail);
	Delphi:	HS_ZP3X_ProbeAvail	(CardNo :longint var ProbeAvail :longint) :longint
Beschreibung:	Liest das Ergebnis der Tastererkennung aus.		
Übergabe- parameter:	CardNo	Logische Kartennummer (1...15)	
	ProbeAvail	Zeiger auf Variable, die das Ergebnis der Tastererkennung aufnimmt.	
		00h:	kein Taster angeschlossen
		01h:	Taster angeschlossen
Rückgabewert:	0	OK	
	-1	Fehler	

3.3 DLLs einbinden

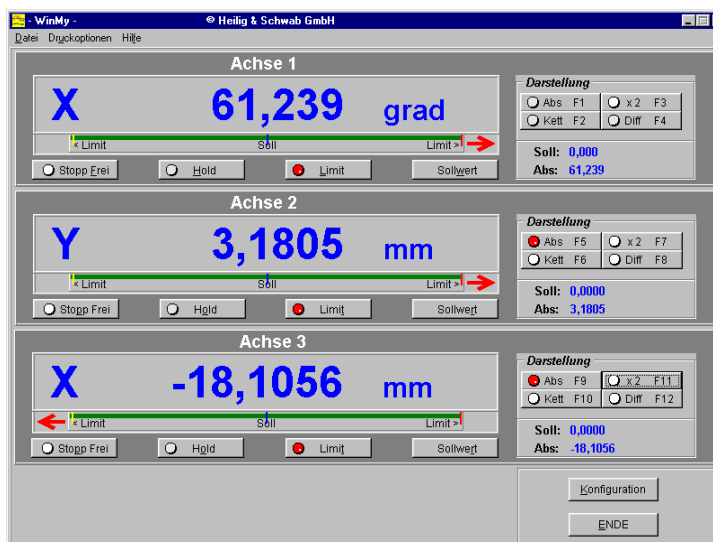
Die DLLs werden über ein Interface-Modul der jeweiligen Programmiersprache eingebunden, das die notwendigen Deklarationen enthält, um auf die Bibliotheksfunktionen zuzugreifen.

Dieses Interface-Modul ist für Visual C++ und für Borland Delphi auf dem im Lieferumfang enthaltenen Datenträger vorhanden.

Für andere Programmiersprachen, wie z.B. Visual Basic, müssen Sie sich dieses Modul selbst erstellen.

3.4 Demo- und Messprogramm WinMy

Die Windows-Software WinMy dient zum Testen der Zählerkarte und ermöglicht Ihnen den direkten Einstieg ins Messen mit inkrementalen Linear- und Rotationsgebern. Das Programm unterstützt die Absolut-, Differenz- und Kettenmaßdarstellung der Zählergebnisse.



Oberfläche des Demoprogrammes WinMy

Software-Lizenz

Diese Vereinbarung stellt die Lizenzbedingungen der mitgelieferten Software auf. Diese Software ist ausschließlich für den Betrieb der Zählerkarte vorgesehen und darf nur in Verbindung mit dieser installiert werden. Die Software oder auch Teile davon dürfen nicht in andere Anwendungen eingebracht und Dritten zugänglich gemacht werden.

4 Technische Daten

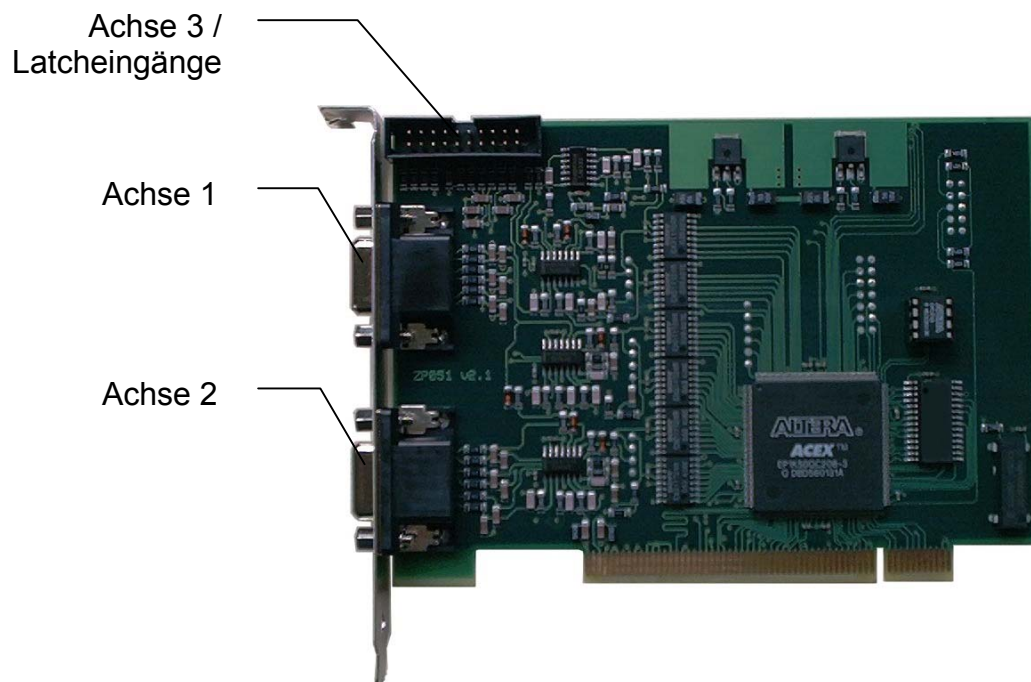
4.1 Steckeranordnung

Zwei der drei Gebersignale (Achse 1 und Achse 2) werden über zwei 9-polige Sub-D-Buchsenleisten, die sich im Bracket der Zählerkarte befinden, direkt an die Zählerkarte angeschlossen.

Die dritte Achse kann über die 20-polige Wannenstiftleiste auf der Leiterkarte der ZP051 angeschlossen werden (siehe Bild "Steckeranordnung der ZP051-20").

Werden Latcheingänge benötigt, so können bis zu drei Latcheingänge zusätzlich über die 20-polige Wannenstiftleiste angeschlossen werden.

Die technischen Spezifikationen der Eingänge wird in den nachfolgenden Abschnitten beschrieben.



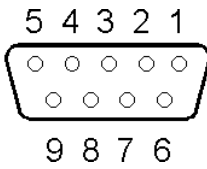
Steckeranordnung der ZP051-20

4.2 Steckerbelegung (Achse 1 und 2)



Wenn Achse 3 über das optional erhältliche Slotblech ZP051/1 oder über die Zähleradapterkarte ZA101 an eine 9-polige Sub-D-Buchsenleiste angeschlossen wird, sind die folgenden Informationen für alle drei Achsen gültig.

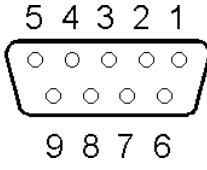
Strom-Eingänge (Standard-Ausführung / ZP051-20)

Pin	Signal	Hinweis
1	- φ_0	 9-pol. Sub-D-Buchsenleisten
2	0 V	
3	- φ_{90}	
4	Schirm	
5	- REF	
6	+ φ_0	
7	+ 5 V	
8	+ φ_{90}	
9	+ REF	

Signalkenndaten:

Signal:	7 - 15 μA_{ss} , typ. 11 μA_{ss} (sinusförmig)
Signalteilung:	256-fache Interpolation
Referenzsignale:	3,5 - 8 μA_{ss} , typ. 5 μA_{ss}
Zählerbreite:	28 Bit
Phasenwinkel φ_0 / φ_{90} :	$90^\circ \pm 10^\circ$
Eingangsfrequenz:	0 - 75 kHz
Anzeigeschritt:	frei wählbar

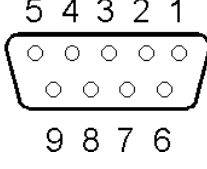
TTL-Eingänge (optional / ZP051-20-TTL)

Pin	Signal	Hinweis
1	/ U _{a1}	 9-pol. Sub-D-Buchsenleisten
2	0 V	
3	/ U _{a2}	
4	Schirm	
5	/ U _{a0}	
6	U _{a1}	
7	+ 5 V	
8	U _{a2}	
9	U _{a0}	

Signalkenndaten:

Signal:	$U_H \geq 2 \text{ V}$, $U_L \leq 0,8 \text{ V}$ (EIA Standard RS422)
Signalteilung:	4-fache Interpolation
Referenzsignale:	$U_H \geq 2 \text{ V}$, $U_L \leq 0,8 \text{ V}$ (EIA Standard RS422)
Zählerbreite:	28 Bit
Eingangsfrequenz:	0 - 500 kHz
Anzeigeschritt:	frei wählbar

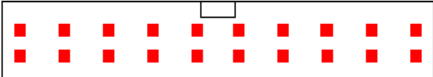
Spannungs-Eingänge (optional / ZP051-20-V)

Pin	Signal	Hinweis
1	A -	 9-pol. Sub-D-Buchsenleisten
2	0 V	
3	B -	
4	Schirm	
5	R -	
6	A +	
7	+ 5 V	
8	B +	
9	R +	

Signalkenndaten:

Signal:	0,8 - 1,2 V _{ss} , typ. 1 V _{ss} (sinusförmig)
Signalteilung:	256-fache Interpolation
Referenzsignale:	0,2 - 0,85 V _{ss}
Zählerbreite:	28 Bit
Phasenwinkel A / B:	90° ± 10°
Eingangsfrequenz:	0 - 75 kHz
Anzeigeschritt:	frei wählbar

4.3 Steckerbelegung (Achse 3 / Latcheingänge Achse 1 – 3)

Pin	Signal			Hinweis
	Strom	(optional)		
		TTL	Spannung	
1	- φ_0	/ U_{a1}	A -	<div></div> <p>20-polige Wannenstiftleiste auf der Leiterkarte</p>
2	+ φ_0	U_{a1}	A +	
3	GND	0 V	0 V	
4	+ 5V	+ 5 V	+ 5 V	
5	- φ_{90}	/ U_{a2}	B -	
6	+ φ_{90}	U_{a2}	B +	
7	GND	0 V	0 V	
8	+ REF	U_{a0}	R +	
9	- REF	/ U_{a0}	R -	
10	+ 5V	+ 5 V	+ 5 V	
11	GND	0 V	0 V	
12				
13				
14				
15	GND	GND	GND	
16				
17	Latch 1	Latch 1	Latch 1	
18	Latch 2	Latch 2	Latch 2	
19	Latch 3	Latch 3	Latch 3	
20	+ 5V	+ 5V	+ 5V	

4.4 Physikalische und mechanische Kennwerte

Abmessungen: 140 x 100 mm (B x H)

Gewicht: ca. 140g (ohne Kabel)

Lagertemperatur: - 30° bis + 70° C

Betriebstemperatur: 0° bis + 45°C

Rel. Luftfeuchte: < 75 %

5 Garantiebedingungen

Der Hersteller garantiert die Funktion ihrer Hard - und Softwareprodukte für die Dauer von einem Jahr nach Lieferdatum. Während dieser Garantiefrist erklärt sich der Hersteller bereit, Produkte, die sich als fehlerhaft erwiesen haben, wahlweise im Herstellerwerk zu reparieren oder zu ersetzen.

Es wird weiterhin vorausgesetzt, dass die Karte nur von entsprechend geschultem und ausgebildeten Personal bedient wird.

Ausgenommen von Garantieleistungen sind:

- Schäden durch unsachgemäße oder unangemessene Reparatur durch den Kunden.
- Schäden durch Software des Kunden.
- Schäden durch nicht korrekten Einsatz der Software.
- Schäden an Verbindungen zu Messsystemen des Kunden.
- Schäden durch nicht genehmigte Veränderungen.
- Schäden durch Nichteinhaltung der Lager - und Betriebsbedingungen.
- Geräte, bei denen die Seriennummer entfernt worden ist.
- Schäden, die durch Hochspannung oder elektrostatische Entladung verursacht worden sind.