
085 USB

PROTOCOL

| | |
|------------------|---|
| AUTOR: | Andreas König |
| VERSION: | 1.02 |
| LETZTE ÄNDERUNG: | 22.10.2009 |
| DATEIABLAG: | P:\085_MOEB_Zentrale\Dokumentation\085_u sb_protocol_v102_de.doc |



INHALTSVERZEICHNISS

| | |
|---|-----------|
| 1. ALLGEMEINES | 3 |
| 1.1 Datentypen und Konstanten | 4 |
| 1.2 Allgemeiner Paketaufbau | 5 |
| 2. DATENÜBERTRAGUNG | 6 |
| 2.1 Asynchrone Übertragungen vom PC → ZENTRALE | 6 |
| 2.1.1 PC → ZENTRALE: Firmware Info | 6 |
| 2.1.2 PC → ZENTRALE: Reset CPU | 6 |
| 2.1.3 PC → ZENTRALE: RF Send Data | 7 |
| 2.1.4 PC → ZENTRALE: Get Serial Number | 8 |
| 2.1.5 PC → ZENTRALE: Set Serial Number | 8 |
| 2.1.6 PC → ZENTRALE: Get Security | 9 |
| 2.1.7 PC → ZENTRALE: Set Security | 9 |
| 2.1.8 PC → ZENTRALE: Get Settings | 9 |
| 2.1.9 PC → ZENTRALE: Set Settings | 9 |
| 2.1.10 PC → ZENTRALE: Get DCC Settings | 10 |
| 2.1.11 PC → ZENTRALE: Set DCC Settings | 10 |
| 2.1.12 PC → ZENTRALE: Get Code | 10 |
| 2.1.13 PC → ZENTRALE: Get Network Children | 10 |
| 2.1.14 PC → ZENTRALE: Get Wireless Module Version | 11 |
| 2.1.15 PC → ZENTRALE: RMBus Get Data | 11 |
| 2.1.16 PC → ZENTRALE: RMBus Program Module | 11 |
| 2.1.17 PC → ZENTRALE: SystemState Get Data | 12 |
| 2.1.18 PC → ZENTRALE: RailCom Get Data | 12 |
| 2.1.19 PC → ZENTRALE: XPressNet | 12 |
| 2.1.20 PC → ZENTRALE: MultiMaus Update | 12 |
| 2.1.21 PC → ZENTRALE: Set Auto Inform Flags | 13 |
| 2.1.22 PC → ZENTRALE: Get Auto Inform Flags | 13 |
| 2.2 Asynchrone Übertragungen von ZENTRALE → PC | 14 |
| 2.2.1 ZENTRALE → PC: XPressNet | 14 |
| 2.2.2 ZENTRALE → PC: MultiMaus Update | 14 |
| 2.2.3 ZENTRALE → PC: RailCom Data Changed | 14 |
| 2.2.4 ZENTRALE → PC: SystemState Data Changed | 14 |
| 2.2.5 ZENTRALE → PC: RMBus Data Changed | 14 |
| 2.2.6 ZENTRALE → PC: NetworkChildren Data Changed | 16 |
| 2.2.7 ZENTRALE → PC: Received RF Data | 16 |
| 3. ÜBERARBEITUNG | 17 |
| 4. DURCHFÜHRUNGSNACHWEIS | 17 |
| 5. ANHANG – COPYRIGHTS | 18 |

1. ALLGEMEINES

Dieses Dokument beinhaltet die Protokolldokumentation für die USB Schnittstelle.

Der Typ der USB Schnittstelle wurde als HID (Human Input Device) implementiert, da auf diese Art kein für die meisten Betriebssysteme kein Treiber notwendig ist. Es gibt je eine Pipe für IN und OUT Data. Pro Datenpaket können maximal 64 Bytes übertragen werden.

Als Vendor-ID wurde 0x16D0 und als Product-ID 0x04D3 gewählt. Die Enumeration der USB-Geräte ist dabei Aufgabe eines PC-Programms, wobei jede Zentrale auf Grund Ihrer Seriennummer auch eindeutig identifiziert werden kann. Es können mehrere Zentralen an einen PC angeschlossen werden, da jede Zentrale auf Grund Ihrer eindeutigen Seriennummer identifizierbar ist. Eine Implementierung auf der PC-Seite erfordert es asynchron zu arbeiten, da die Zentrale nicht unbedingt auf jede Frage sofort mit dem richtigen Befehl antwortet. Es können jederzeit ungefragte Informationen. (z.B. Kurzschluss, etc) auftreten!

**085 USB
PROTOCOL**

Version 1.02

1.1 Datentypen und Konstanten

Die Werte der Konstanten und die Definition der Typen (haben ein T am Anfang) sind im Header-File „commoncode.h“ zusammengefasst.

Liste der USB_xxx Konstanten

| Konstante | Wert |
|---------------------------------|-------------|
| USB_FIRMWARE_INFO | 0x01 |
| USB_RESET_CPU | 0x02 |
| USB_RF_SENDDATA | 0x03 |
| USB_RF_SENDSTATE | 0x04 |
| USB_RF_RECEIVEDDATA | 0x05 |
| USB_RF_ACK | 0x06 |
| USB_GET_SERIALNUMBER | 0x10 |
| USB_SET_SERIALNUMBER | 0x11 |
| USB_GET_SETTINGS | 0x12 |
| USB_SET_SETTINGS | 0x13 |
| USB_GET_SECURITY | 0x14 |
| USB_SET_SECURITY | 0x15 |
| USB_GET_DCCSETTINGS | 0x16 |
| USB_SET_DCCSETTINGS | 0x17 |
| USB_GET_CODE | 0x18 |
| USB_GET_WIRELESSMODULEVERSION | 0x20 |
| USB_XPRESSNET | 0x40 |
| USB_MMUPDATE | 0x41 |
| USB_SETAUTOINFORMFLAGS | 0x50 |
| USB_GETAUTOINFORMFLAGS | 0x51 |
| USB_RMBUS_DATACHANGED | 0x80 |
| USB_RMBUS_GETDATA | 0x81 |
| USB_RMBUS_PROGRAMMODULE | 0x82 |
| USB_SYSTEMSTATE_DATACHANGED | 0x84 |
| USB_SYSTEMSTATE_GETDATA | 0x85 |
| USB_NETWORKCHILDREN_DATACHANGED | 0x86 |
| USB_NETWORKCHILDREN_GETDATA | 0x87 |
| USB_RAILCOM_DATACHANGED | 0x88 |
| USB_RAILCOM_GETDATA | 0x89 |
| USB_SECTOR_INFO | 0xF8 |
| USB_SECTOR_ERASE | 0xF9 |
| USB_SECTOR_READ | 0xFA |
| USB_SECTOR_WRITE | 0xFB |
| USB_START_APP | 0xFC |
| USB_FINISH_PROGRAM | 0xFD |

1.2 Allgemeiner Paketaufbau

Da in einem HID-Paket keine Datenlänge angegeben werden kann, werden die Daten immer als 64-Byte Paket gesendet. Damit der jeweilige Empfänger dennoch über die Datenlänge bescheid weiß, wird folgender Paketaufbau verwendet. Dieser Aufbau gilt für beide Übertragungsrichtungen:

| DataLength | Header | Daten |
|---|---------------|--------------|
| Die Länge von DataLength+Header+Daten (2+Length(Daten)) | X | X |

DataLength und Header haben dabei jeweils die Größe von einem Byte.

Die in diesem Dokument nachfolgend beschriebenen Pakete beginnen nur noch mit dem Header. Dies bedeutet, dass eigentlich vor jedem beschriebenen Paket noch die Datenlänge hinzugefügt wird. Der Header beschreibt die prinzipielle Funktion der Daten, die Daten können daher eine maximale Länge von $64-2 = 62$ Bytes besitzen. Auf eine Prüfsumme wurde bewusst verzichtet, da die Datenkorrektur und CRC Prüfung bereits die USB-Schnittstelle als solches übernimmt.

2. DATENÜBERTRAGUNG

2.1 Asynchrone Übertragungen vom PC → ZENTRALE

2.1.1 PC → ZENTRALE: Firmware Info

| Header | Daten |
|-------------------|-------|
| USB_FIRMWARE_INFO | |

Antwort

| Header | Daten |
|-------------------|--------------|
| USB_FIRMWARE_INFO | FirmwareInfo |

FirmwareInfo ist dabei wie folgt aufgebaut:

| Offset | Feldgröße | Feld | Funktion |
|------------------------------------|----------------|--------------------|--|
| 0x00 | Byte | Mode | Gibt den Modus an (1=Bootloader, 2=Programm) |
| 0x01 | Byte | Valid | Gibt an ob das Programm gültig ist (1) oder ungültig (0) |
| 0x02 | Long | BootLoader Version | Gibt die Version des Bootloaders an |
| 0x06 | Long | Firmware Version | Gibt die Version der des Programms an |
| 0x0A | 0-term. String | Manufacturer | Zeichenkette auf den Herstellernamen |
| 0x0A+0x01+ Length(Manufacturer) | 0-term. String | Product | Zeichnkette auf den Produktnamen |

2.1.2 PC → ZENTRALE: Reset CPU

| Header | Daten |
|---------------|--------------------|
| USB_RESET_CPU | ClearFirmwareValid |

Das Byte ClearFirmwareValid muss nicht mitgesendet werden, wenn es gesendet wird und den Wert 0x5A hat, so wird die Firmware als ungültig markiert und die Zentrale bleibt nach dem Start im Modus Bootloader (Dient nur zum aktualisieren der Firmware über den Bootloader).

Antwort

| Header | Daten |
|---------------|-------|
| USB_RESET_CPU | Code |

Code beinhaltet nach erfolgreicher Ausführung den Wert 1, ansonsten den Wert 0

085 USB
PROTOCOL

Version 1.02

2.1.3 PC → ZENTRALE: RF Send Data

| Header | Daten |
|-----------------|------------|
| USB_RF_SENDDATA | TRadioData |

Der Record TRadioData hat dabei zwei Felder. Das Feld NetworkAddress gibt die Netzwerkadresse des Zielteilnehmers an (zur Zentrale: 0, ansonsten die MAC-Adresse des Teilnehmers). Das Feld AirData entspricht dem Record TAirData.

Der Record TAirData hat dabei wiederum zwei Felder. Das Feld Header gibt einen Wireless Header an, der über die Konstante WL_xxx definiert wird. Das Feld Data gibt dann die tatsächlichen Nutzdaten an. Die Länge errechnet sich dann aus den zu übertragenden Nutzdaten im Feld Data+1 (für den AirData.Header) + RADIODATA_AIRDATA_OFFSET (für die Netzwerkadresse).

Nur AirData wird dann auch tatsächlich über Funk übertragen.

Antwort: Die Zentrale sendet nur eine Antwort auf diesen Befehl, wenn zum einen der AirData.Header >= WL_BROADCAST_TO_PC ist und zum anderen das AutoInformFlag aiWireless gesetzt ist (Nähere Informationen zu den AutoInformFlags finden Sie hier: „PC → ZENTRALE: Set Auto Inform Flags“). Ansonsten wird keine Antwort auf den Befehl gesendet.

| Header | Daten |
|------------------|---------------|
| USB_RF_SENDSTATE | TRadioAckData |

ACHTUNG: Der Header der Antwort entspricht nicht dem Header der Anfrage!

TRadioAckData ist dabei wie folgt aufgebaut:

Byte0-1 (Word): NetworkAddress (beinhaltet die Netzwerkadresse des Ziels)

Byte 2 (Byte): Header (beinhaltet den gesendeten AirData.Header)

Byte 3 (Byte): Beinhaltet den Status (1=Erfolgreich, 0=Fehler)

Der Status gibt jedoch nur an, dass das Paket gesendet werden konnte, eine Bestätigung, dass das Paket auch vom Ziel empfangen wurde wird ggf. später gesendet.

Die Zentrale sendet die Sendebestätigung (Das Paket wurde nun erfolgreich zugestellt) nur, wenn das AutoInformFlag aiWireless gesetzt ist. Die Sendebestätigung kann längere Zeit benötigen. In der Zwischenzeit können weitere Befehle gesendet und empfangen werden!

| Header | Daten |
|------------|-------|
| USB_RF_ACK | Code |

Code beinhaltet den Status (1=Erfolgreich, 0=Fehler).

**085 USB
PROTOCOL**

Version 1.02

2.1.4 PC → ZENTRALE: Get Serial Number

| Header | Daten |
|----------------------|-------|
| USB_GET_SERIALNUMBER | |

Mit diesem Befehl wird die Seriennummer der Zentrale angefragt.

Antwort

| Header | Daten |
|----------------------|--------------|
| USB_GET_SERIALNUMBER | SerialNumber |

Als Antwort wird die Seriennummer der Zentrale (Typ: long) zurückgegeben.

2.1.5 PC → ZENTRALE: Set Serial Number

| Header | Daten |
|----------------------|--------------|
| USB_SET_SERIALNUMBER | SerialNumber |

Mit diesem Befehl wird die Seriennummer der Zentrale gesetzt (und von der Zentrale im EEPROM dauerhaft abgespeichert).

Antwort: Keine

**085 USB
PROTOCOL**

Version 1.02

2.1.14 PC → ZENTRALE: Get Wireless Module Version

| Header | Daten |
|-------------------------------|-------|
| USB_GET_WIRELESSMODULEVERSION | |

Mit diesem Befehl wird die Versionsinformation des Funkmoduls ausgelesen.

Antwort:

| Header | Daten |
|-------------------------------|------------------------|
| USB_GET_WIRELESSMODULEVERSION | TWirelessModuleVersion |

Als Antwort wird der Record TWirelessModuleVersion zurückgegeben, wobei Version die Version enthält und Name den Name des Funkmoduls (inkl. Version) enthält.

2.1.15 PC → ZENTRALE: RMBus Get Data

| Header | Daten |
|-------------------|------------|
| USB_RMBUS_GETDATA | GroupIndex |

Mit diesem Befehl wird/werden die Informationen über die Rückmelder angefragt.

GroupIndex enthält den Index der Rückmeldermodulkette (0=Rückmelder 1-10, 1=Rückmelder 11-20, 2=Rückmelder 21-30 und 3=Rückmelder 31-40) wobei derzeit nur die Werte 0 und 1 gültig sind.

Soll die Zentrale die Daten für alle Rückmelder zurücksenden, so wird der Wert GroupIndex weggelassen (nicht gesendet). Die Zentrale antwortet daraufhin mit mehreren RMBus Data Changed Paketen. Ansonsten wird nur ein einzelnes RMBus Data Changed Paket gesendet.

Als Antwort wird der Befehl „ZENTRALE → PC: RMBus Data Changed“ gesendet.

2.1.16 PC → ZENTRALE: RMBus Program Module

| Header | Daten |
|-------------------------|-------------|
| USB_RMBUS_PROGRAMMODULE | ModuleIndex |

Mit diesem Befehl wird ein Rückmelde-Modul programmiert. Der entsprechende Programmierbefehl wird anschließend so lange ausgegeben, bis dieser Befehl wieder aufgerufen wird und der ModuleIndex wieder auf 0 gesetzt wird. Die Vorgehensweise beim Programmieren eines Rückmeldemoduls ist dabei der Bedienungsanleitung des Rückmeldemoduls zu entnehmen.

Der ModuleIndex zum programmieren bezeichnet die Adresse des Moduls (gültige Werte sind 1-40, wobei derzeit nur die Adressen von 1-20 verwendet werden).

Antwort: Keine

085 USB
PROTOCOL

Version 1.02

2.1.17 PC → ZENTRALE: SystemState Get Data

| Header | Daten |
|-------------------------|-------|
| USB_SYSTEMSTATE_GETDATA | |

Mit diesem Befehl wird der Systemstatus angefragt.

Als Antwort wird der Befehl „ZENTRALE → PC: SystemState Data Changed“ gesendet.

2.1.18 PC → ZENTRALE: RailCom Get Data

| Header | Daten |
|---------------------|-------|
| USB_RAILCOM_GETDATA | |

Mit diesem Befehl werden die RailCom Informationen angefragt.

Als Antwort wird der Befehl „ZENTRALE → PC: RailCom Data Changed“ gesendet.

2.1.19 PC → ZENTRALE: XPressNet

| Header | Daten |
|---------------|-------|
| USB_XPRESSNET | Daten |

Mit diesem Befehl werden XPressNet Befehle an die Zentrale gesendet.

Die Daten enthalten dabei den reinen X-PressNet Befehl.

2.1.20 PC → ZENTRALE: MultiMaus Update

| Header | Daten |
|--------------|-------|
| USB_MMUPDATE | Daten |

Mit diesem Befehl werden MultiMaus Update Befehle an die Zentrale gesendet.

Die Daten enthalten dabei den reinen X-PressNet Befehl.

085 USB
PROTOCOL

Version 1.02

2.1.21 PC → ZENTRALE: Set Auto Inform Flags

| Header | Daten |
|------------------------|-----------------|
| USB_SETAUTOINFORMFLAGS | AutoInformFlags |

Mit diesem Befehl werden die AutoInformFlags gesetzt. Das Feld AutoInformFlags ist vom Typ long und entspricht einer oder Verknüpfung von nachfolgenden Werten:

| Flag | Wert | Bedeutung |
|-------------------|------------|---|
| aiXPressNet | 0x00000001 | X-Pressnet-Änderungen werden automatisch an den PC gesendet |
| aiRMFeedBack | 0x00000002 | Rückmelder-Änderungen werden automatisch an den PC gesendet |
| aiRailCom | 0x00000004 | RailCom-Änderungen werden automatisch an den PC gesendet |
| aiMMUpdate | 0x00000008 | MMUpdate-Änderungen werden automatisch an den PC gesendet |
| aiSystemState | 0x00000100 | SystemState-Änderungen werden automatisch an den PC gesendet. |
| aiWireless | 0x00000200 | Spezielle Funk-Befehle werden automatisch an den PC gesendet. (Nur für Bootloader etc. notwendig) |
| aiWirelessConnect | 0x00000400 | Anmelde und Abmeldevorgänge von Endgeräten werden automatisch an den PC gesendet. |

Antwort: Keine

2.1.22 PC → ZENTRALE: Get Auto Inform Flags

| Header | Daten |
|------------------------|-------|
| USB_GETAUTOINFORMFLAGS | |

Mit diesem Befehl werden die AutoInformFlags abgefragt.

Antwort:

| Header | Daten |
|------------------------|-----------------|
| USB_GETAUTOINFORMFLAGS | AutoInformFlags |

Die Antwort liefert die aktuellen AutoInformFlags zurück. Die Bedeutung dieser Flags entnehmen Sie bitte dem Punkt „PC → ZENTRALE: Set Auto Inform Flags“.

2.2 Asynchrone Übertragungen von ZENTRALE → PC

2.2.1 ZENTRALE → PC: XPressNet

| Header | Daten |
|---------------|-------|
| USB_XPRESSNET | Daten |

Dieser Befehl wird nur gesendet, wenn das Flag AutoInformFlag aiXPressNet gesetzt ist. Weitere Informationen finden Sie unter „PC → ZENTRALE: Set Auto Inform Flags“. Des weiteren können dann auch ungefragte Antworten gesendet werden.

2.2.2 ZENTRALE → PC: MultiMaus Update

| Header | Daten |
|---------------|-------|
| USB_XPRESSNET | Daten |

Dieser Befehl wird nur gesendet, wenn das Flag AutoInformFlag aiMMUpdate gesetzt ist. Weitere Informationen finden Sie unter „PC → ZENTRALE: Set Auto Inform Flags“. Des weiteren können dann auch ungefragte Antworten gesendet werden.

2.2.3 ZENTRALE → PC: RailCom Data Changed

| Header | Daten |
|-------------------------|--------------|
| USB_RAILCOM_DATACHANGED | TRailComData |

Als Befehl wird der Record TRailComData zurückgegeben.
TRailComData ist in der Datei „Global.h“ der MultiZentralePro Firmware definiert.

Der Befehl wird nur gesendet, wenn das AutoInformFlag aiRailCom gesetzt ist oder wenn die Daten explizit über den Befehl „PC → ZENTRALE: RailCom Get Data“ angefragt wurden. Weitere Informationen finden Sie unter „PC → ZENTRALE: Set Auto Inform Flags“.

2.2.4 ZENTRALE → PC: SystemState Data Changed

| Header | Daten |
|-----------------------------|--------------|
| USB_SYSTEMSTATE_DATACHANGED | TSystemState |

Als Befehl wird der Record TSystemState zurückgegeben.
TSystemState ist in der Datei „Global.h“ der MultiZentralePro Firmware definiert.

Der Befehl wird nur gesendet, wenn das AutoInformFlag aiSystemState gesetzt ist oder wenn die Daten explizit über den Befehl „PC → ZENTRALE: SystemState Get Data“ angefragt wurden. Weitere Informationen finden Sie unter „PC → ZENTRALE: Set Auto Inform Flags“.

2.2.5 ZENTRALE → PC: RMBus Data Changed

| Header | Daten |
|-----------------------|-------|
| USB_RMBUS_DATACHANGED | Daten |

Als Befehl werden folgende Daten zurückgegeben.
Byte 0 enthält den Gruppenindex (derzeit nur 0 oder 1).
Byte 1..10 enthält den Status der Rückmelder (Jedes Byte entspricht genau einem Rückmelder, jedes Bit in jedem Byte enthält daher genau einen Kontakt).

**085 USB
PROTOCOL**

Version 1.02

Der Befehl wird nur gesendet, wenn das AutoInformFlag aiRMFeedBack gesetzt ist oder wenn die Daten explizit über den Befehl „PC → ZENTRALE: RMBus Get Data“ angefragt wurden. Weitere Informationen finden Sie unter „PC → ZENTRALE: Set Auto Inform Flags“

2.2.6 ZENTRALE → PC: NetworkChildren Data Changed

| Header | Daten |
|---------------------------------|------------------|
| USB_NETWORKCHILDREN_DATACHANGED | TNetworkChildren |

Der Record TNetworkChildren beinhaltet dabei als ersten Wert ein Byte mit der Anzahl der mit der Zentrale verbunden Handregler. Danach sind 4 Word Werte enthalten, welche die MAC-Adresse der einzelnen Handregler enthalten.

Der Befehl wird nur gesendet, wenn das AutoInformFlag aiWirelessConnect gesetzt oder wenn die Daten explizit über den Befehl „PC → ZENTRALE: Get Network Children“ angefragt wurden. Weitere Informationen finden Sie unter „PC → ZENTRALE: Set Auto Inform Flags“.

2.2.7 ZENTRALE → PC: Received RF Data

| Header | Daten |
|---------------------|------------|
| USB_RF_RECEIVEDDATA | TRadioData |

Der Record TRadioData hat dabei zwei Felder. Das Feld NetworkAddress gibt die Netzwerkadresse des Teilnehmers an, von dem das Paket stammt (entspricht der MAC-Adresse des Teilnehmers). Das Feld AirData entspricht dem Record TAirData.

Der Record TAirData hat dabei wiederum zwei Felder. Das Feld Header gibt einen Wireless Header an, der über die Konstante WL_xxx definiert wird. Das Feld Data gibt dann die tatsächlichen Nutzdaten an. Die Länge errechnet sich dann aus den zu übertragenden Nutzdaten im Feld Data+1 (für den AirData.Header) + RADIODATA_AIRDATA_OFFSET (für die Netzwerkadresse).

Der Befehl wird nur gesendet, wenn das AutoInformFlag aiWireless gesetzt ist und wenn der AirData.Header >= WL_BROADCAST_TO_PC st. Weitere Informationen finden Sie unter „PC → ZENTRALE: Set Auto Inform Flags“.

3. ÜBERARBEITUNG

| DATUM | ART DER ÜBERARBEITUNG | AUTOR | VERSION |
|----------|---------------------------|-------|---------|
| 20.09.09 | ERSTELLUNG DES DOKUMENTES | AK | 1.00 |
| 12.10.09 | RÜCKMELDER ÜBERARBEITET | AK | 1.01 |
| 22.10.09 | USB_GET_CODE EINGEFÜGT | AK | 1.02 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

4. DURCHFÜHRUNGSNACHWEIS

Für die korrekte Durchführung dieses Dokumentes zeichnet:

Andreas König
haber & koenig electronics GmbH

5. ANHANG – COPYRIGHTS

Die in diesem Dokument verwendeten Bezeichnungen für Erzeugnisse, die zugleich eingetragenes Warenzeichen darstellen, wurden nicht besonders gekennzeichnet. Das Fehlen der ® Markierung ist demzufolge nicht gleichbedeutend mit der Tatsache, dass die Bezeichnung als freier Warenname gilt. Ebenso wenig kann anhand der jeweils verwendeten Bezeichnung auf eventuell vorliegende Patente oder einen Gebrauchsmusterschutz geschlossen werden. Es sei ausdrücklich darauf verwiesen, dass die *hke gmbh* weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf den Gebrauch oder den Inhalt dieser Beschreibung zurückzuführen sind. Die in dieser Beschreibung enthaltenen Angaben können ohne vorherige Ankündigung geändert werden. Die *hke gmbh* geht damit keinerlei Verpflichtungen ein.

Ferner sei ausdrücklich darauf verwiesen, dass die *hke gmbh* weder Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf falschen Gebrauch oder falschen Einsatz der Hardware zurückzuführen sind. Ebenso können ohne vorherige Ankündigung Layout oder Design der Hardware geändert werden. Die *hke gmbh* geht damit keinerlei Verpflichtungen ein.

© Copyright *hke gmbh* – Alle Rechte vorbehalten.

Kein Teil dieses Datenblattes darf in irgendeiner Form ohne schriftliche Genehmigung der Firma *hke gmbh* unter Einsatz entsprechender Systeme reproduziert, vervielfältigt oder verarbeitet werden.

www.hkegmbh.com

haber & koenig electronics gmbh
A-5101 BERGHEIM bei SALZBURG

technology with the customer in mind