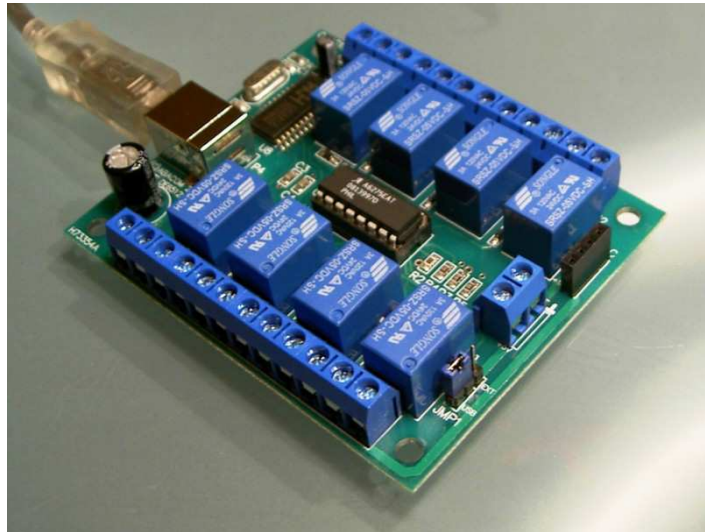


## USB-Relaiskarte LRB



### Features

- Kompakte USB-Relaiskarte in der Grösse einer Zigarettenschachtel
- Abmessungen ca. 70 x 85 x 20 mm
- Ideal für Modellbau, Kfz-Installationen, Mess- und Steuerungstechnik, etc.
- Betriebsspannung (+5V) wahlweise vom USB oder extern
- Geringe Stromaufnahme, typ.300 mA (alle Relais an)
- 8 Relais mit je einem Umschaltkontakt, 1A / 24V DC bzw. 1A / 40V AC
- komfortabler Anschluss über Schraubklemmen; **Maximaler Klemmenstrom 1A**
- USB-Chipsatz: CH341A
- Integrierte I<sup>2</sup>C/TWI-Master-Schnittstelle
- Systemvoraussetzungen: XP, 2000, Vista, WIN 7, 32 / 64 Bit
- Einsetzbar mit ProfiLab 4.0.
- Einfache Programmierung per ActiveX-Steuerelement (OCX)
- Programmbeispiele in C++, Delphi und Visual Basic.

### Download-Link für Anleitung, Treiber, Beispiele

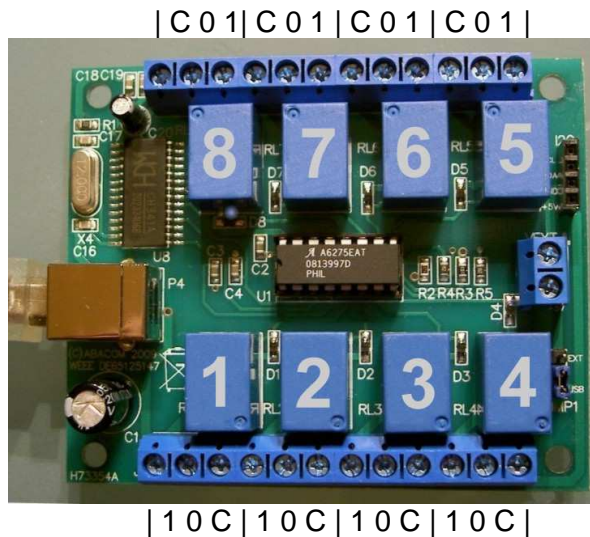
[http://www.abacom-online.de/div/setup\\_usb\\_LRB.exe](http://www.abacom-online.de/div/setup_usb_LRB.exe)

### Installation

- Bevor Sie die Karte mit dem USB verbinden, laden Sie bitte unbedingt die die zugehörige Software herunter und führen Sie das Installationsprogramm aus.
- Danach können Sie die Karte mit dem USB verbinden und die gewohnte Windows-Plug&Play-Installation durchführen.
- Nun ist die Karte einsatzbereit. Weitere Informationen zur Verwendung der Karte finden Sie jetzt in der Anleitung und in den weiteren Dateien, die das Installationsprogramm installiert hat.

## Relais und Schaltkontakte

Jedes der acht Relais verfügt über je einen Umschaltkontakt. Somit ist jedem Relais eine dreipolige Schraubklemme zugeordnet. Ist ein Relais ausgeschaltet so ist der Kontakt zwischen seinen Klemmen C und 0 geschlossen. Im eingeschalteten Zustand ist der Kontakt zwischen C und 1 geschlossen. C (Common) kennzeichnet also den gemeinsamen Wechselkontakt, der je nach Relaisstatus den Kontakt zur Klemme 0 (Relais aus) oder Klemme 1 (Relais an) herstellt. Alle Kontakte sind ohne externe Beschaltung potenzialfrei. Der maximale Klemmenstrom beträgt 1A.



Die Ansteuerung der Relais erfolgt intern über einen USB-Interface-Chip (CH341A) und ein Schieberegister mit Ausgangslatch, dessen Treiberausgänge direkt die Relais ansteuern. Dieses verhindert wirksam das Auslösen ungewollter Schaltvorgänge.

---

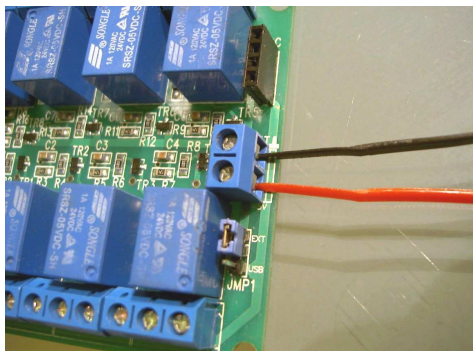
*Bitte beachten Sie beim Anschluss von Verbrauchern, dass Schaltfunken an den Relaiskontakten die empfindliche Elektronik der Karte stören können. Dieses gilt insbesondere beim Schalten induktiver und kapazitiver Lasten, wie z.B. Motoren, Spulen, Schütze, etc. In diesen Fällen sollten die Relaiskontakte in geeigneter Weise entstört werden. In vielen Fällen genügt zur Funkenlöschung bereits ein einfacher keramischer Kondensator mit 100nF, der zwischen C und 0 bzw. C und 1 angeschlossen wird.*

---

## Relaistreiber und Spannungsversorgung

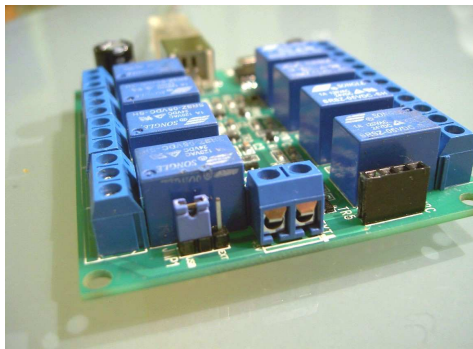
Das Relaiskarte ist für die Spannungsversorgung per USB ausgelegt. Die Stromaufnahme beträgt typisch ca. 300 mA. Da der USB max. 500 mA liefert, ist bei Verwendung weiterer Geräte am USB eventuell ein AKTIVER USB-HUB (Verteiler) notwendig.

In Fällen in denen keine ausreichende Stromversorgung per USB gewährleistet ist, kann eine externe Spannungsversorgung der Relais erfolgen. Dazu muss der Jumper JMP1 in die Position EXT umgesteckt werden. Über die Schraubklemmen VEXT kann nun eine **STABILISIERTE 5V-Versorgungsspannung** eingespeist werden. Achten Sie dabei unbedingt auf die richtige Polung und die richtige Spannung (5V), da andernfalls Schäden am Gerät entstehen können. Einfache Steckernetzteile sind für diesen Zweck in der Regel NICHT geeignet. Die externe Stromversorgung sollte wenigsten 350 mA liefern können.



### Externe Spannungsversorgung

Rot = +5V  
Schwarz = (PC-)Masse  
Position JMP1 = EXT

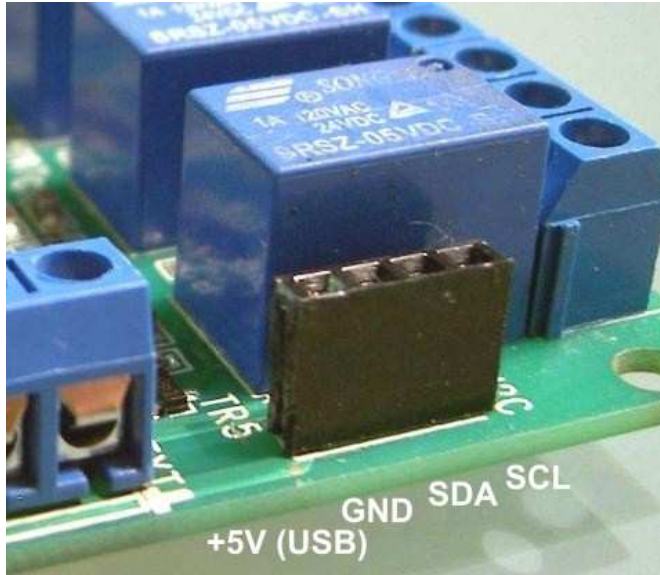


### Interne Spannungsversorgung per USB

Position JMP1 = USB

## I<sup>2</sup>C-Master-Anschluss

Eine vierpolige Buchsenleiste stellt die I<sup>2</sup>C-Signale und die USB-Versorgungsspannung für eigene Schaltungserweiterungen bereit. Dabei darf die Gesamtstromentnahme aus dem USB-Port 500mA nicht übersteigen. Die I<sup>2</sup>C-Pullup-Widerstände (meist 2K2) müssen in der Anwenderschaltung vorgesehen werden.



## **Software**

Laden Sie die Software von unseren Internetseiten herunter. Das Setup-Programm kopiert folgende Ordner auf Ihre Festplatte:

### **API**

Dieser Ordner enthält Dateien und Beispiele für die eigene Programmierung der Karte mit verschiedenen Programmiersprachen.

### **DOC**

Enthält einige Datenblätter und evtl. nützliche Zusatzinformationen.

### **DRIVER:**

Dieser Ordner enthält die USB-Treiber die Sie für die Windows Plug&Play-Installation benötigen.

Nach dem Anstecken am USB muss die Treiberinstallation für den CH341A-Interfacechip erfolgen. Den Treiber finden Sie im Ordner DRIVER. Danach ist das Gerät einsetzbar. Die Adressierung der Karte erfolgt über eine fortlaufende Gerätenummer des CH341A-Interface-Chips. Alle am System angeschlossenen CH341A-Chips erhalten automatisch eine fortlaufende Nummer beginnend mit 0.

Programme die Daten an das Gerät senden wollen, öffnen zunächst anhand der Gerätenummer einen Datenkanal zum CH341A-Chip, über den dann die Datenübertragung zu den Relais erfolgt. In der Anwendungssoftware muss in der Regel nur die richtige Gerätenummer eingestellt werden, um die Relais anzusprechen.

### **TEST**

enthält ein einfaches Testprogramm für einen ersten Funktionstest.

## **Funktionstest**

Für einen ersten Funktionstest und zum Ermitteln der richtigen Gerätenummer finden Sie ein Testprogramm im Ordner \TEST. Starten Sie das Testprogramm (USB\_REL\_TEST.EXE).

## **ProfiLab Expert**

Beispielprojekte für ProfiLab Expert 4.0 finden Sie im Ordner \PROFILAB. ProfiLab stellt eine entsprechende Hardware-Komponente zur Steuerung der Relais bereit. Bitte stellen Sie sicher, dass Sie das neueste Internet-Update Ihrer ProfiLab 4.0-Software verwenden. Auch die Ansteuerung des I<sup>2</sup>C erfolgt in ProfiLab über die CH341A-API-Funktionen

## Programmierschnittstelle

Programmierbeispiele für verschiedene Programmiersprachen finden Sie im Ordner \API der Softwareinstallation.

Die Programmierung der Relais und der I<sup>2</sup>C-Schnittstelle erfolgt mit Hilfe eines ActiveX-Steuerlements. Dieses befindet sich in der Datei ABACOM\_USB\_LRB.OCX. Das Steuerlement wird bei der Installation der Software automatisch unter dem Namen "ABACOM USB REL" registriert und kann mit allen Programmiersprachen verwendet werden, die den Import von ActiveX-Steuerlementen unterstützen. Grundlegende Informationen zum Import von ActiveX-Steuerlementen entnehmen Sie bitte der Anleitung zu Ihrer Programmiersprache. Dies nachstehenden Programmauszüge stellen keinen lauffähigen Programmcode dar. Sie dienen nur zur Erläuterung des Funktionsprinzips. Bitte beachten Sie auch die mitgelieferten Programmbeispiele.

Für die Programmierung erzeugt man zunächst ein "ABACOM USB REL" – Objekt vom Typ USBLRBX. Dies kann entweder dynamisch zur Laufzeit oder mit Hilfe des Formular-Designers der Programmiersprache geschehen.

```
TUSBLRBX *REL ;  
  
REL = new TUSBLRBX(this);  
REL->Parent = this;  
REL->Left = 24;  
REL->Top = 40;
```

Neben den gewohnten Eigenschaften zur Verwaltung des Steuerlements, wie .z.B. LEFT, TOP, etc. verfügt das Steuerlement über Methoden und Eigenschaften, die speziell der Programmierung der Relais und der I<sup>2</sup>C-Schnittstelle dienen, wie z.B.

```
REL->REL_Status = ...
```

oder

```
REL->I2C_Write(...)
```

Diese Namen dieser Eigenschaften und Methoden beginnen entsprechend Ihrer Verwendung mit "REL\_" bzw. "I2C\_".

## Programmierung der Relais

Den acht Relais bzw. dem Schaltzustand sind Eigenschaften (Properties) REL\_1 ...REL\_8 zugeordnet, die es erlauben den Relaiszustand einzelner Relais gezielt zu setzen:

```
REL->REL_1 = 1; // Relais 1 einschalten  
REL->REL_3 = 0; // Relais 3 ausschalten
```

Ein gleichzeitiges Schalten aller Relais erfolgt durch Setzen der Eigenschaft REL\_Status. Dabei ist den einzelnen Bits 0..7 des Statusbytes jeweils ein Relais 1..8 zugeordnet:

```
REL->REL_Status = 3; //Bit 0 und Bit 1 gesetzt => REL1+REL2 EIN  
REL->REL_Status = 0; // Alle aus  
REL->REL_Status = 255; // Alle an
```

Für die Verwendung mehrerer Karten legt man ggf. mehrere Relaiskarten-Objekte an und setzt die Gerätenummer REL\_DeviceNo entsprechend:

```
REL1->REL_DeviceNo = 0; //erste Karte  
REL2->REL_DeviceNo = 1; //zweite Karte  
usw.  
RELX->REL_DeviceNo = -1 // Disabled device
```

## Programmierung der I<sup>2</sup>C/TWI-Schnittstelle

Das Karte besitzt eine I<sup>2</sup>C-Master-Schnittstelle, die es erlaubt zusätzliche I<sup>2</sup>C-Peripherie-Chips am PC zu betreiben. Dazu sind Kenntnisse der I<sup>2</sup>C-Technik und das Verständnis der Datenblätter der verwendeten Chips unerlässlich.

Die Programmbeispiele beziehen sich auf eine I<sup>2</sup>C-Demonstrationsschaltung von MicroChip(TM), die anschlussfertig als Zubehör bei uns erworben werden kann.  
[http://www.electronic-software-shop.com/product\\_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2\\_13](http://www.electronic-software-shop.com/product_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2_13)

Ebenso lassen sich - bei entsprechender Programmierung - natürlich auch andere I<sup>2</sup>C-Schaltungen an der Schnittstelle betreiben.

Für die Kommunikation mit I<sup>2</sup>C-Chips stehen folgende Funktionen zur Verfügung:

### Byteweises Schreiben und Lesen Chip-Registern

Die Methode

```
REL->I2C_Write(0x20,0x09,LEDStatus)
```

schreibt ein Byte (hier LEDStatus) an die Chip-(Slave)-Adresse 0x20 in ein Chip-Register (hier 0x09).

```
REL->I2C_Read(0x20,0x09,InData)
```

Liest ein Register (hier 0x09) eines I<sup>2</sup>C-Chips, dessen Chip-Adresse 0x20 ist. Der Rückgabewert InData enthält den gelesenen Byte-Wert als lesbare Zeichenkette (HEX-Dump; z.B "A7")

### Direktes Schreiben und/oder Lesen von Busdaten

Ein direktes Schreiben und Lesen von Datenbytes über den I<sup>2</sup>C-Bus ist mit der Funktion STREAM-Funktion möglich:

```
OutData = "92 00";  
REL->I2C_Stream(outData, 2, inData);
```

Die Datenübergabe (inData; outData) erfolgt als lesbare Zeichenkette (Hex-Dump). Das Beispiel schreibt zwei Bytes 0x92 und 0x00 auf den Bus. Die Anzahl der zu schreibenden Bytes wird aus der Länge des Hex-Dumps (OutData) implizit ermittelt.

Die Anzahl der zu lesenden Bytes muss explizit angegeben werden (hier zwei). Der Rückgabewert inData liefert wiederum eine lesbare Zeichenkette der eingelesenen Bytes, wie z.B. "FE B6".

Die Funktion schreibt/liest die Daten wie gesagt – ohne jede eigene Logik – direkt über den I<sup>2</sup>C-Bus und ist damit sehr universell einsetzbar.

### EEPROM

Das Schreiben und Lesen eines Standard-EEPROMS wird durch die folgende Funktionen ermöglicht:

```
REL->I2C_WriteEEPROM(eprom24C02, 0, "FF FE");  
REL->I2C_ReadEEPROM(eprom24C02, 0, 7, Data);
```



Der Funktion wird zum Schreiben der EEPROM-Typ, die Speicher-(Start-)Adresse und die Daten als lesbarer Hex-Dump übergeben. Zum Lesen muss zusätzlich die Anzahl der zu lesenden Bytes (hier: sieben) übergeben werden.

### Taktfrequenz

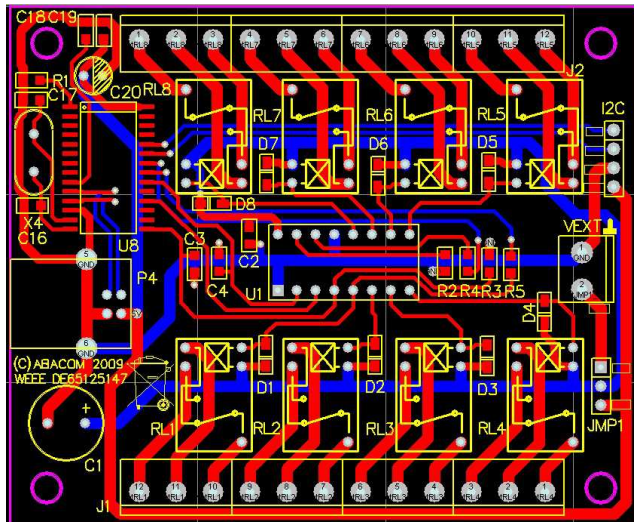
Die I<sup>2</sup>C-Übertragungsgeschwindigkeit, d.h. die Taktfrequenz ist in vier Stufen einstellbar:

```
REL->I2C_Speed = slow_20kHz  
REL->I2C_Speed = normal_100kHz  
REL->I2C_Speed = fast_400kHz  
REL->I2C_Speed = high_750kHz
```

Hohe Taktraten sind nicht unbedingt für alle I<sup>2</sup>C-Chips geeignet.

### Abmessungen

70 x 85 x 20mm



© ABACOM-Ingenieurgesellschaft