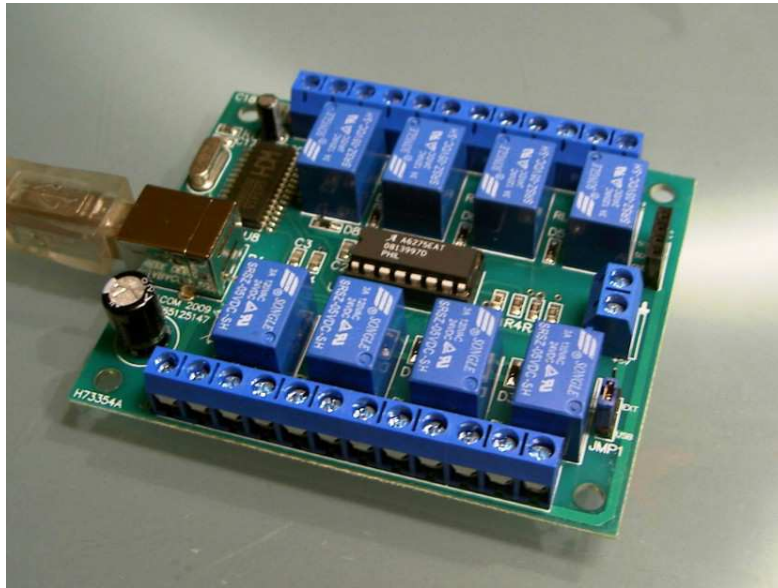


USB relay board LRB



Features

- Compact USB relay board
- Dimensions approx.. 70 x 85 x 20 mm
- Suitable for car installations, model building, measurement and control applications
- Power supply (+5V) either USB or external
- Low power consumption, typ.300 mA (all relays on)
- 8 relays, each with one switch-over contact, 1A / 24V DC or 1A / 40V AC
- Convenient screw-terminal connectors; **Max. terminal current 1A**
- USB chip set: CH341A
- Integrated I²C/TWI master interface
- System requirements XP, 2000, Vista, WIN 7, 32 / 64 Bit
- Suitable for ProfiLab 4.0.software
- Easy programming Interface with ActiveX control (OCX)
- Programming examples in C++, Delphi und Visual Basic.

Download link for manual, driver and examples

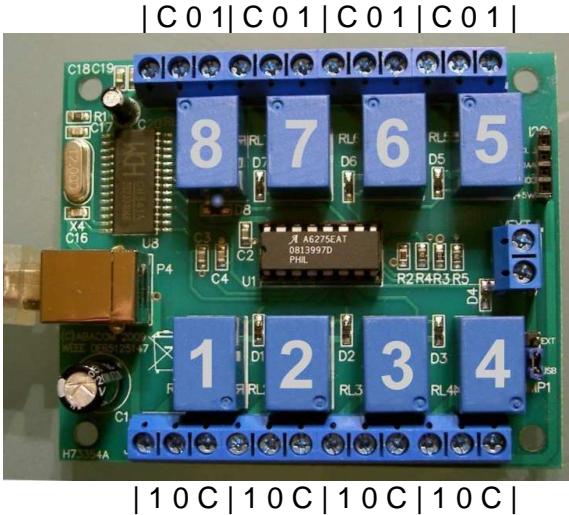
http://www.abacom-online.de/div/setup_usb_LRB.exe

Installation

- Before connecting the device to the USB download and install setup software please.
- After that connect the device to the USB and follow the Windows Plug & Play instructions.
- The device is now ready for use. Further information on how to use the board can be found in the manual installed with the software.

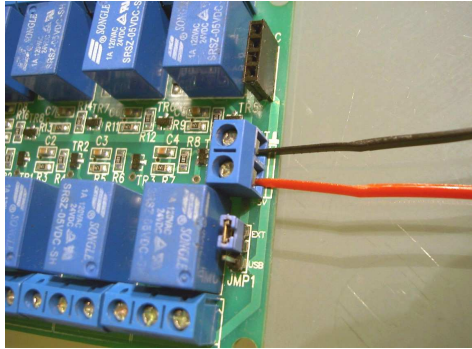
Relays and contacts

Each relay has one switch-over contact routed to a three-lead screw-terminal connector. While relay is in OFF state, terminals C and 0 are connected. In relay ON state terminals C and 1 are connected. C is the common switch over contact. Relay contact are isolated against any other electrical parts. Maximum contact current is 1A.

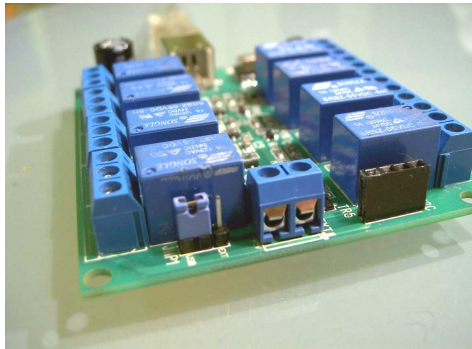


Relay drivers and power supply

Nevertheless in some cases the USB may not be strong enough to drive the board. In such a case the board can be supplied from an external **STABLE** 5V DC power supply. Therefore the jumper JUMP1 must change its position to EXT. After that the external power can be connected. Check polarity carefully! Otherwise device gets damaged. Never use cheap and simple DC adaptor plugs. These are not suitable to supply the board. The supply should be strong enough to drive 150 mA loads.



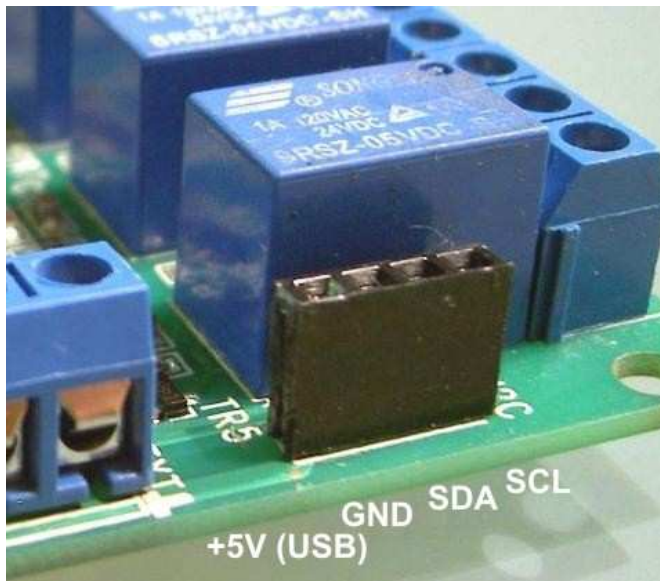
External power supply (red = +5V; black = Ground)



Internal power supply from USB

I²C master connector

The USB power (V_{cc}, GND) and the I²C signals (SDA and SCL) are available at a female pin connector. Make sure that the overall current from USB does not exceed 500 mA. The I²C pull-up resistors (usually 2K2) must be provided by the user circuitry.



Software

The software setup copies the following files to your hard-disc:

API

This directory contains files and examples made for some programming languages.

DOC

Contains some useful datasheets.

DRIVER:

This folder contains files necessary for Windows plug & play installation. Connecting the device to the USB these driver files for CH341A chipset installed. The device is then ready for use. In software applications the device is addressed with a device number. This is a consecutive number of available CH341A chips starting from zero. Applications open a data channel to the CH341A chip that will then execute the data transport to the relays. In most applications the device no. is adjustable.

TEST

A simple search & test program for the relay board can be found in this folder.

Functional test

For a simple device test and to find out the device no. use the USB_REL_TEST.EXE from the \TEST folder.

ProfiLab Expert

Examples for ProfiLab Expert 4.0 can be found in the directory \PROFILAB. Please make sure to have latest Internet update of your ProfiLab software installed. The I²C bus can also be controlled, using corresponding CH341A API functions.

Programming interface (API)

Programming examples for some languages can be found in the \API directory of the software installation.

The relays and the I²C interface is programmed using a ActiveX control (file ABACOM_USB_LRB.OCX). The ActiveX control is registered as "ABACOM USB REL" during software installation process and can be imported with any programming language that support ActiveX controls. Please refer to the manual of your programming language how ActiveX controls are handled.

The following code snippets are meant as illustration for the functional principles and may fail to compile. Remember the code examples.

To program the board an object "ABACOM USB REL" (USBLRBX) must be created. This could be done dynamically at run-time or with the IDE's form designer.

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

The REL object defines some properties and methods for general control administration (left, top, etc.) and specialized ones for relay and I²C control., like

```
REL->REL1_STATUS = ...
```

or

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

Methods and Properties are named with leading "REL_" and "I2C_".corresponding to their purpose.

Relay programming

Each of the eight relays has a corresponding property REL_1 ... REL_8. Setting these properties will switch relays individually:

```
REL->REL_1 = 1; //relay 1 = ON  
REL->REL_5 = 0; //relay 5 = OFF
```

All relays can be set simultaneously writing the REL_Status property, where each relay status is represented by one of the bits 0..7 in the control byte.

```
REL->REL_Status = 3; //bit 0 and Bit 1 set => REL1+REL2 ON  
REL->REL_Status = 0; // All off  
REL->REL_Status = 255; // All on
```

In the need to use more than one board, additional REL objects must be created. To address a device the device number must be set:

```
REL1->REL_DeviceNo = 0; //1st device  
REL2->REL_DeviceNo = 1; //second device  
etc.
```

```
RELX->REL_DeviceNo = -1 //Device disabled (default)
```

I²C (TWI) programming

The device is equipped with an integrated I²C(TWI) master interface, which makes it possible to connect additional peripheral chips and control them with PC software. Some basic knowledge in I²C technology and understanding of chips datasheets is necessary for that.

The programming examples are based on a ready-made I²C demonstration board from MicroChip (TM), which can be purchased separately from our web-shop. (http://www.electronic-software-shop.com/product_info.php?pName=microchip-ic-board-p-17&cName=hardware-ic-twi-c-2_13)

Instead of that, any other I²C circuitry could be connected and programmed.

The following I²C methods are usable for that purpose:

Read/write bytes from/to chip registers

The method

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

Writes a byte (here LEDStatus) to the chip (slave) address 0x20 into a chip register (here 0x09).

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

Read a register (here 0x09) of a chip, with chip address 0x20. The return value InData contains a readable ASCII string (HEX dump; e.g. "A7")

Direct read/write data over i²C bus

The following stream function allows direct read and/or write over the I²C bus:

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

The parameters (inData; outData) are readable ASCII strings (Hex dumps). The example writes two bytes 0x92 and 0x00 out on the bus. The number of bytes to write is implied with the the length OutData hex dump.

The number of bytes to read must be given in the second parameter (here: two). The return value is a readable hex dump string, like "FE B6" of the read bytes.

This function writes/reads bytes – without certain protocol – over the I²C bus and is therefore usable with almost any peripheral chip type.

EEPROM

Read/write to EEPROM can be done with

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

For write operation the EEPROM type, the memory (start) address and the write data must be fed. For read operations the number of bytes to read must be given in addition (here: seven). Read and write data are readable hex dump strings again.

Clock frequency

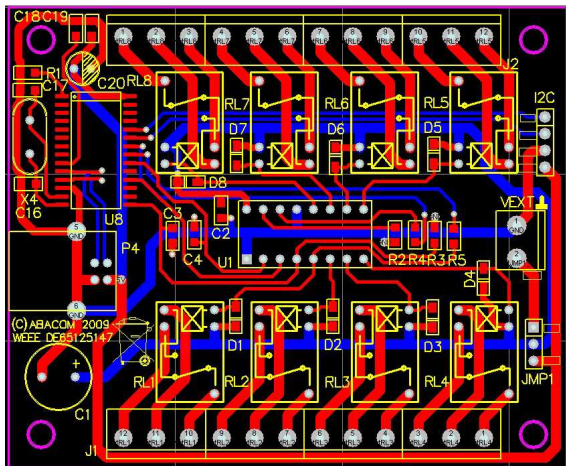
The I²C speed can be adjusted in four steps:

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

Fast and high clock rates may not be supported by all I²C chips.

Dimensions

70 x 85 x 20mm



© ABACOM-Ingenieurgesellschaft