4 Channel USB Relay Module User Manual

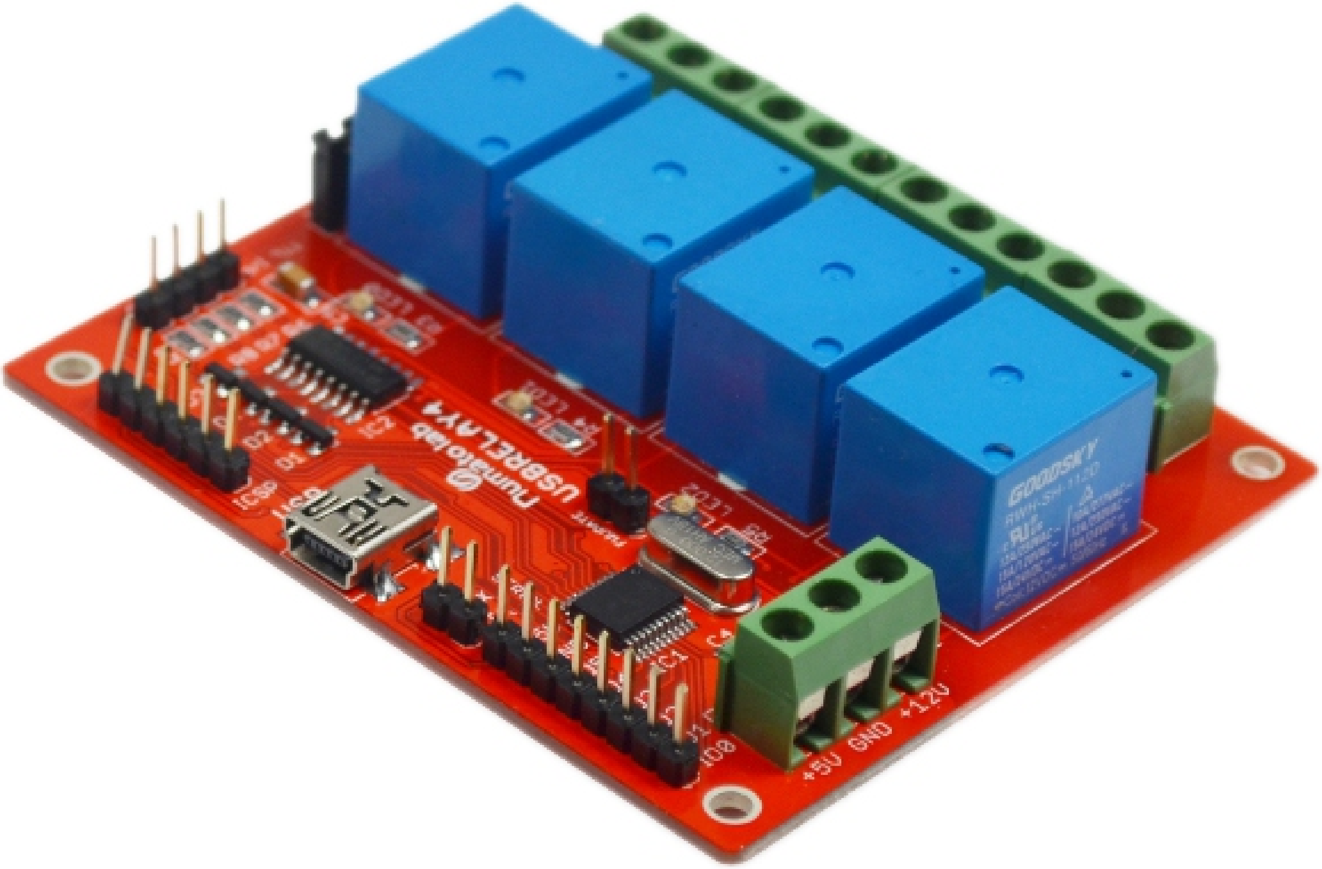


Table of Contents

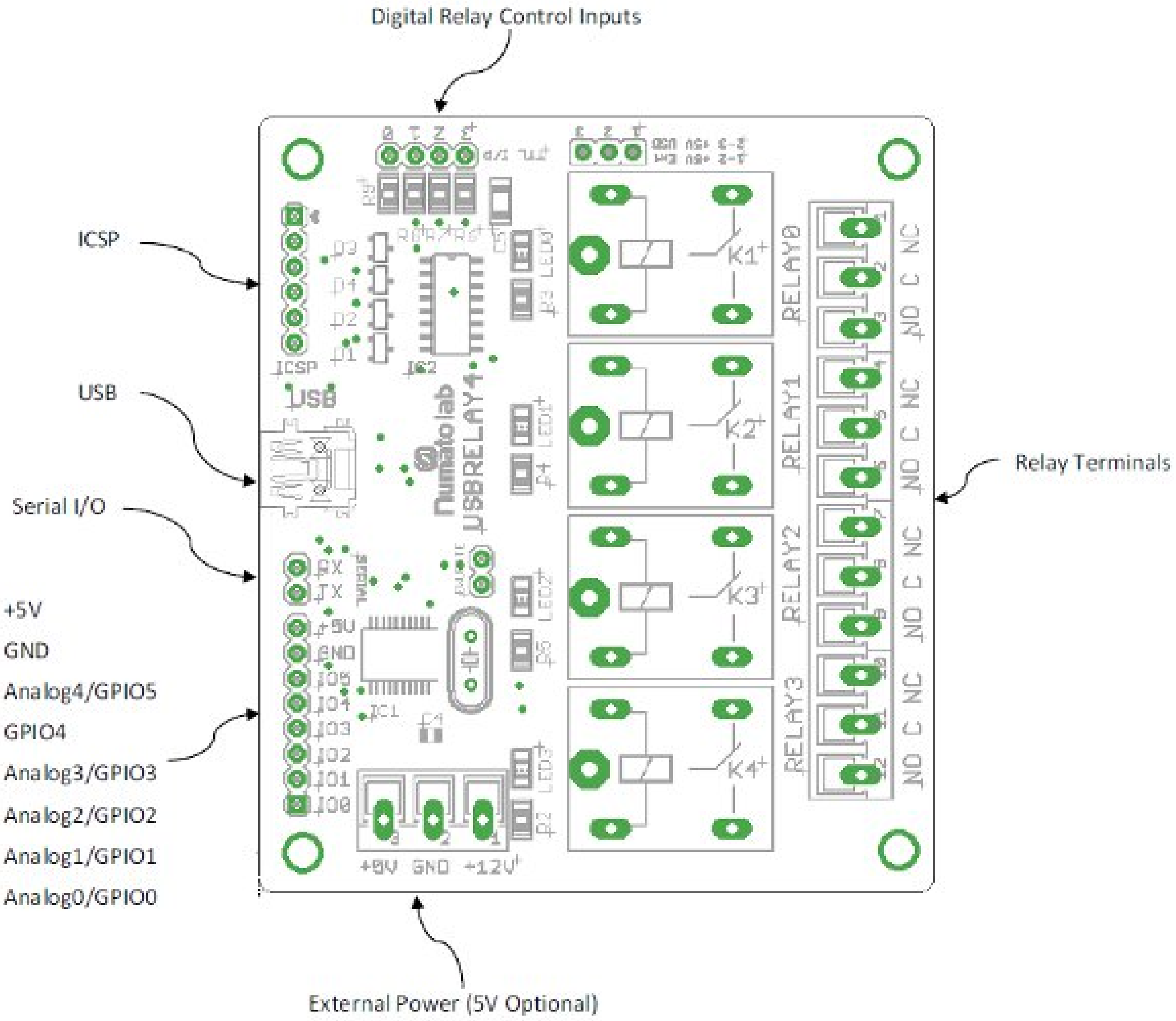
Introduction

Numato USB Relay Module is a versatile piece of hardware that is useful for development as well as production purposes. It comes with four on board relays and associated drivers. Apart from relays, the board has 4 General purpose I/Os, and 6 analog inputs (multiplexed with GPIOs). On board ICSP header and built-in bootloader makes it easy to use USB Relay module as a development system as well. The board communicates with host PC over high-speed USB link. When connected to PC, the board will show up as a normal serial port in Device Manager. A serial terminal emulation program like HyperTerminal can be used to control individual relays and I/Os.

Electrical Specification

System Supply Voltage	**- 5V , USB powered or Externally powered**
Relay Supply Voltage	**- 12V External**
Microcontroller	**- PIC18F14K50**
IO Voltages	**- 5V Max**
Analog Input Resolution	**- 10 Bits**
Max Relay load Current	**- 7A Max at 250V / 12A Max at 120V**

Wiring Details



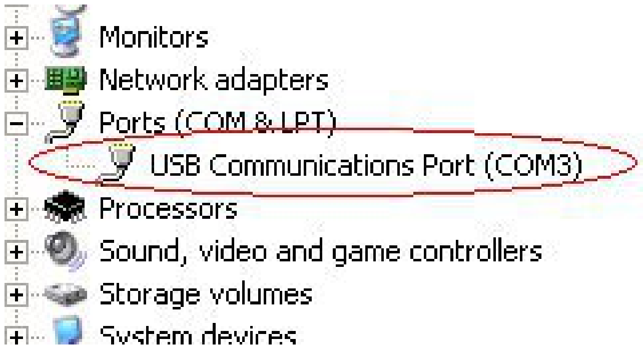
[Using the Module]

The primary functionality of this module is to control the on-board relays from a PC through USB link. Besides controlling relays, there are digital and analog I/Os available which can be individually read or written according to your requirements. The board's digital circuitry can be powered either from USB or from an external power supply. By default the board is shipped with USB powered configuration (Jumper JP1 pins 2 and 3 connected together). If the board needs to be powered from an external source, connect pin no. 3 of screw terminal X3 to a +5V supply and move jumper on JP1 to position 1-2. Relay drive circuitry always run on externally supplied +12V supply (Screw Terminal X3 Pin No. 1). It is very important that +12v supply is connected in order for the relays to work. If relays are not used, this supply is not necessary.

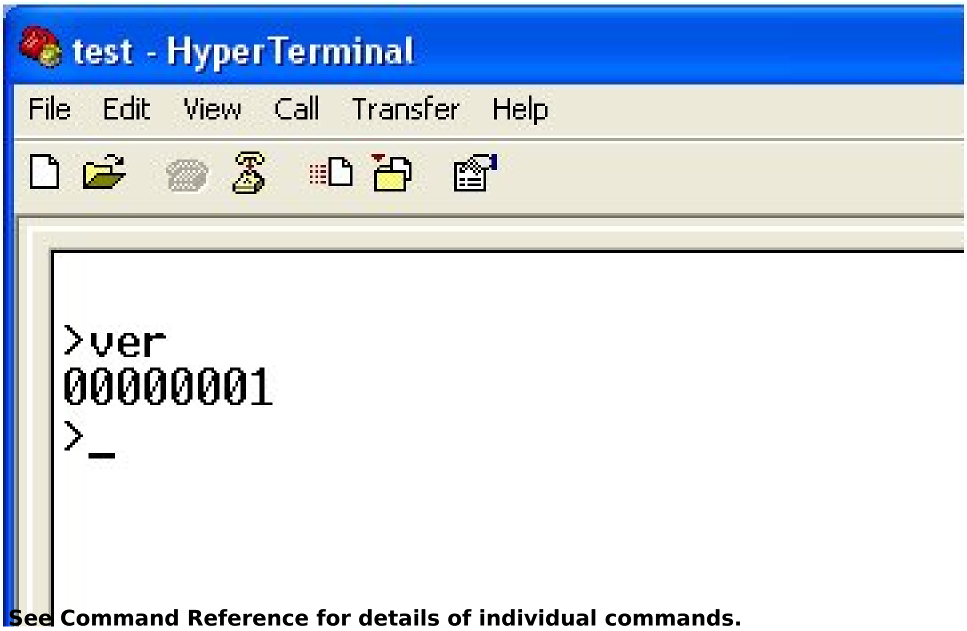
Screw Terminal X2 gives access to all relay terminals. It is recommended to coat a layer of solder on all tracks that connect relay terminals to corresponding screw terminals (solder mask is kept open on all these tracks) if the board is used to switch more than 3A current.

When the Relay Controller Board is connected to a PC for the first time, a window will pop up asking for the drivers. Please follow the wizard and load the necessary driver from the 'Driver' folder in the USBRELAY4 zip package. If everything goes fine, you should be able to see a new serial port in Device manager as in the

picture below (The port number can be different on your machine).



At this point, the board can be accessed by opening the corresponding serial port using a serial terminal program like HyperTerminal. When configuring the port make sure that flow control is set to “none”. Rest of the settings can be left as defaults. Pressing the ‘Enter’ key should show a command prompt sign (>) where you can enter the commands. Enter ‘ver ‘ to see the firmware version. The version should be printed on the console as shown in the picture below.



```
>ver
00000001
>_
```

See Command Reference for details of individual commands.

[Serial Interface]

This board supports a serial interface which can be used to control relays using a microcontroller or a PC serial port over a standard RS232 data interface. The Serial Interface pins do support only 5V logic. So to connect to a standard PC Serial Port, a level translator such as MAX232 is required. This interface supports a single byte command which can be used to read or write relay status. A response byte will be sent by the board only for read requests. To confirm if a write command was successful, the status of the relay should be queried. Please see the table below for command details.

Command

Bit	**Definition**
7	**0 – Write**

	1 - Read
6	Value to be written during a write command. 0 – Turn off relay 1 – Turn on relay Reserved during Read command.
5	Reserved
4 - 0	Relay index

Response

Bit	Definition
7	Reserved
6	Relay Status 0 – Relay Off 1 – Relay On
5	Reserved
4 - 0	Relay index

Digital Relay Control Inputs

There four digital relay control inputs available which can be used to control the relays by simply inputting 5V/0V logic. It is recommended not to use these inputs when the board is acrively being controlled by USB or Serial interface.

Updating Firmware

The board comes with bootloader pre programmed. Firmware can be changed/updated using Microchip HID bootloader application which is a part of Microchip Application Libraries. Microchip Application Libraries can be downloaded from http://ww1.microchip.com/downloads/en/DeviceDoc/MCHP_App_Lib_v2010_10_19_Installer.zip.

Command Reference

ver

Command : ver

Arguments : None

Usage : ver

Example : ver

Description : Displays version of the firmware running on the board.

relay

Command : relay

Arguments : on/off/read, relay number

Usage : relay on RelaNum , relay off RelaNum, relay read RelayNum

Example : relay on 0, relay off 0, relay read 0

Description : Control individual relays on the board.

reset

Command : reset

Arguments : None

Usage : reset

Example : reset

Description : Resets all relays to off state. However, GPIOs won’t be affected by this command.

adc

Command : adc

Arguments : read, channel

Usage : adc read channel

Example : adc read 0, adc read 1

Description : Reads voltage on the specified analog input.

gpio

Command : gpio

Arguments : set/clear/read, Gpio Number

Usage : gpio set gpionum, gpio clear gpionum, gpio read gpionum

Example : gpio set 0, gpio clear 0, gpio read 0

Description : Reads or writes a logic state from/to an individual GPIO pin.

Frequently Asked Questions

1. Can I program this board with my own firmware?

Yes, as long as you know what you are doing. You can write your own firmware and use it with this board. This board has a PIC18F14K50 microcontroller, so make sure you are building the code for this processor. Use ICSP connector to program the microcontroller.

2. I see green mask is open on some traces on the bottom side, why is it so?

Those tracks are kept open intentionally so that user can coat a layer of solder on the track to increase current handling capacity.

3. Can I control the relays from my own application instead of using HyperTerminal or other terminal emulators?

Yes. It is possible to control the board with different programming languages/scripts. All that we need to make sure is that the language you choose supports serial port programming. Some possible Tools/Languages are,

Visual Basic

Visual C++

Perl

Python

VBA (Word, Excel etc...)

And more...

4. Can I power relays from USB instead of external +12V supply?

No. It may work sometimes, but not guaranteed to work every time. Moreover, when all relays are energized at the same time, it can draw too much current and cause power surge on the USB port and can even kill your USB port.

Source URL: http://numato.com/4-channel-usb-relay-module-user-manual