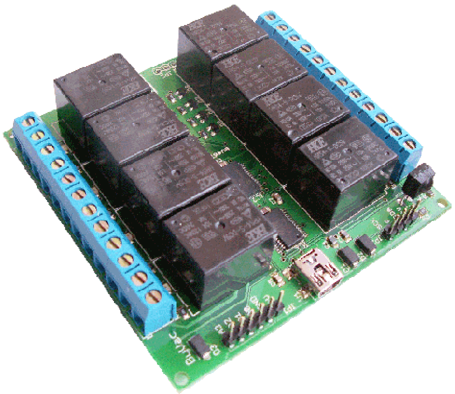
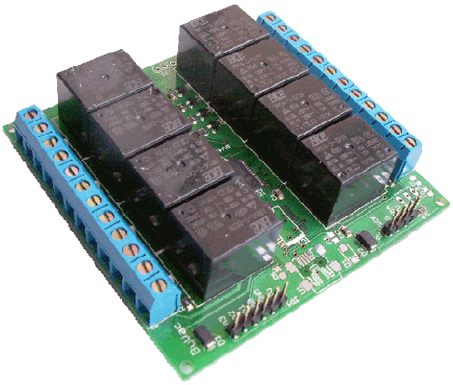




BV4627

Multi Interface 8 Way Relay

Product specification

December 2010 V0.a

Contents

Rev	Change
Jan 2010	Preliminary
April 2011	Added relay specification
June	Minor error correction to esc[<num>i command
April 2013	New PCB version BV4627-f

1. Introduction

This device contains 8 single pole double throw (SPDT) relays that can be operated in no less then FOUR ways. The actual attached relays are capable of switching 10A @24VDC and 10A @250VAC.

Each relay can be instantly switched on or off or commanded to switch at a set time of up to 1.3 hours (USB, I2C only).

There are two versions of this board. The full version and a cut down version that does not have the IR and USB interface. This data sheet describes the full version, if you have the cut down version then ignore the IR and USB sections.

1.1.1.New PCB

A new PCB version has been introduces from March/April 2013 onwards. The size and holes are the same (exact information is here http://doc.byvac.com/index.php5?title=Product_BV4627)

The main change is to the relay output, see the diagram in section 3.

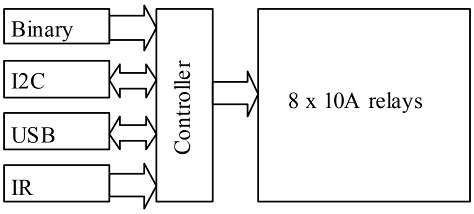
2. Features

- 8 relays 10A contact rating
- Binary interface
- I2C interface
- USB interface
- Infra Red Interface
- Delayed on/of timer*
- Momentary or toggle switch action**
- Voltage 5V***
- Current 1.5mA (all relays off)
- Size 83x80x18mm

* Feature available on USB & I2C interfaces only
** Feature available on IR interface only
*** Each relay when on consumes approximately 80mA.
The relays fitted will have the following minimum specification:

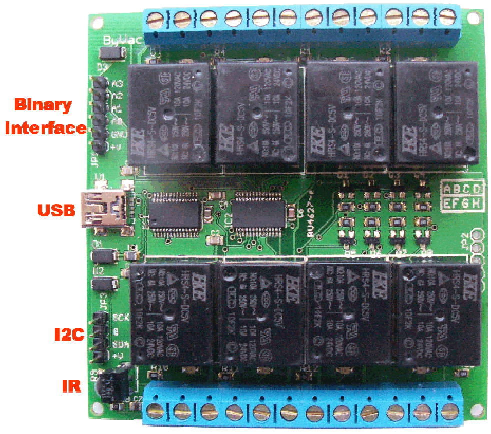
- Normally Open 10A @ 240V AC
- Normally Closed 6A @ 240V AC
- 10A @ 24V DC

3. Physical Specification

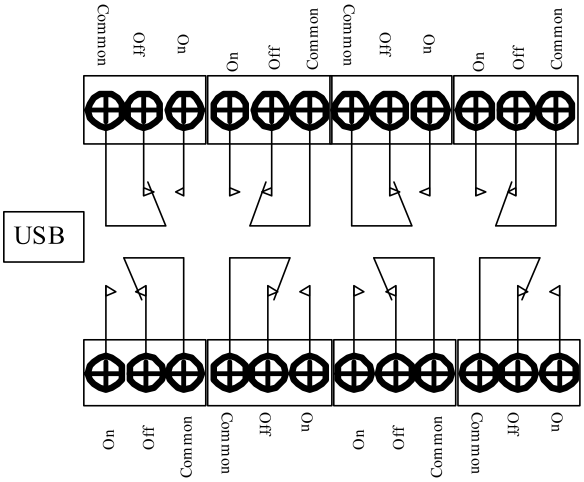


Block Diagram

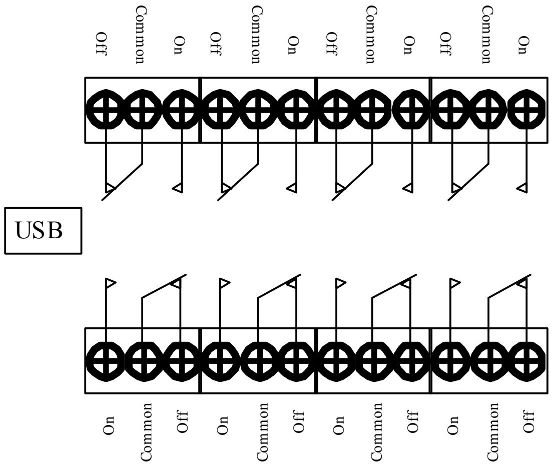
The device has four interface options each are mutually exclusive, that is to say only one interface can be used at any one time. The device is powered from either the USB, I2C or binary interfaces. There is no on board regulator and so when not used with USB requires a power supply of 5V + or – 10%. The processor will work down to 3.3V but the IR detector and relays require 5V.



Relay Connections



PCB version BV4627-e and before



PCB Version BV4627-f (after March 2013)

4. Interface Options

There are 4 interface options for this device and each have their priority as follows:

1. Binary interface
2. I2C interface
3. USB Interface
4. IR Interface

The selection of the interface is determined at reset or power on and it will remain with the selected interface until the next reset. In general only one interface can be used at any one time.

Options available for each interface

Interface	Delayed on/off	Toggle
Binary	NO	NO
I2C	YES	NO
USB	YES	NO
IR	NO	YES

When all other interfaces are disconnected the IR interface will be active, this however needs to be set up first using the USB interface.

NOTE: To allow the detection of the interface there is a 1 second delay after reset or switch on before commands can be sent.

5. Binary Interface

The purpose of this interface is to give a quick and easy method of interfacing to a microcontroller without the need for 8 individual logic lines. It is very simple to use and should be easily implemented using just 4 output lines.

5.1. Selection

The interface is selected by taking A0, the clock pin low at reset or power on. This is the idle state for this pin. When the binary interface is selected no other interfaces are active. When disconnected this pin is held high by an internal resistor.

5.2. Pins

The interface consists of 6 pins at jumper 1 as follows:

Pin	Function
A3	I/P
A2	I/P
A1	I/P
A0	Clock I/P
GND	
+V	+5V

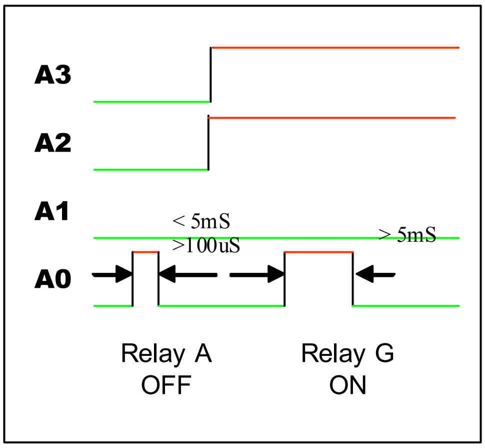
The device can also be powered through this interface via +V and GND.

The relay is selected using A1 to A3 and the clock is pulsed to switch the relay on or off. A long pulse is on and a short pulse is off.

Relay selection; the following table shows the values required to select a particular relay.

Rly>	A	B	B	D	E	F	G	H
A1	0	1	0	1	0	1	0	1
A2	0	0	1	1	0	0	1	1
A3	0	0	0	0	1	1	1	1

The clock input in its idle state is low, a long pulse, exceeding 5mS will turn on the selected relay and a short pulse, less then 5mS will turn off the selected relay.



The diagram shows how to switch on or off a particular relay. The relay is switched on the falling edge of the clock.

5.3. To Switch a relay on

1. Select the relay using lines A1-A3
2. Pull line A0 high and keep it high for more then 5mS
3. Return A0 to low, when A0 returns low the relay will be switched.

5.4. To Switch a relay off

1. Select the relay using lines A1-A3
2. Pull line A0 high and leave high for 100uS but less then 5mS.
3. Return A0 low within 5mS of pulling it high.

6. I2C Interface

This is a standard interface with an 8 bit **address of 0x64** (7 bit 0x32). The commands are shown in the command table and all numbers in the table are in decimal. The address can be changed and this will be stored semi permanently in EEPROM.

6.1. Selection

The I2C interface is selected by holding the SDA line high at switch on or reset. This will occur automatically when connected to an I2C bus as there are pull up resistors on the bus so simply connecting to an I2C bus will select this interface.

When no I2C is connected the SDA line is held low by a resistor. When this interface is connected all other interfaces are disabled.

Pin	Function
1	+V
2	SDA
3	GND
4	SCL

The device can also be powered through this interface via the +V and GND lines. Details of what each command does is given in the command table but in general the following algorithms are used.

6.2. Sending Data

1. send start condition
2. send 0x64
3. send command(s)
4. send stop condition

6.3. Receiving Data

1. send command as above (6.2)
2. send start condition
3. send 0x65
4. receive byte(s)
5. send stop condition

The command table uses a shorthand notation that is used in the BV4221 IC2 Terminal device (http://www.byvac.com).

s sends the start condition followed by the device address;

p sends the stop condition;

r sends the restart condition followed by the device address+1;

g-n gets n number of bytes.

Using this notation to immediately turn on relay A would be:

s 10 1 0 1 p

To get the device ID requires that command 85 be sent before receiving 2 bytes, thus:

s 83 r g-2 p

7. USB Interface

The USB interface presents itself as a COM port via the use of a virtual COMM port IC (http://www.ftdichip.com/). The latest drivers for this can be found at the specified site. There are also drivers in the resource pack which can be found at http://www.byvac.com It is a very popular device and so it may already be installed on your machine.

When the COM port is established communication is via text commands as in the table at section 9.1.

All commands begin with an escape character (0x1b) and are followed by other text characters. For example to turn on relay A the following sequence would be sent:

esc[1A

In byte this would be:

0x1b 0x5b 0x32 0x41

The command is activated on receiving the last byte ‘A’ in this case. To turn on relays A and B:

esc[1Aesc[1B

7.1. Selection

The interface is selected upon inserting the USB cable, it uses the TX line to detect the presence of the USB. When this interface is in use the other interfaces are locked out. However it is possible to activate the IR interface using the esc[?32t command.

7.2. Handshake and ACK

The ideal interface would implement the hardware handshaking mechanism using RTS/CTS. This is where the devices RTS is connected to the host CTS, when the device is busy, it raises the RTS line and the host temporarily stops transmitting.

Where it is not possible to do this and also where software can be simplified, the ACK mechanism is available. This is switched off by default and so must be enabled using the appropriate command.

Because the device takes a finite time to carry out a command, say turning on a relay, it is useful to have an indication of when that has happened and more accurately when the device is ready to accept another command. This is what the ACK mechanism is for. When enabled and when the device has completed a command and is ready to accept another command it will send an ACK character. The actual character (value from 1 to 255) can be specified by the user.

Not all commands will send an ACK where this is so it is indicated on the command table.

8. IR Interface

The IR interface is designed to be as flexible as possible and should, with a bit of patience, be made to work with most remote control handsets. This is NOT a grantee though and will depend on the handset itself and how much time will be devoted to testing the various codes.

It has for example been tried with a £1.00 all-in-one remote from a pound shop and works well with the Philips code 1009.

The IR interface is implemented using the IR detector mounted on the corner of the PCB. This type works on 36KHz which should be okay for most IR remote transmitters.

By default it is not set to any particular controller so before it is used it must be set up using the USB interface. For full set up instructions see the BV4627 User Guide.

When using the IR remote the device can be powered from the Binary interface pins marked +V and GND or the I2C power pins. when setting up the device this will of course be powered by the USB.

This interface offers the opportunity of being able to turn on a relay when the remote is pressed and it goes off when released (like a volume control). This is called momentary or single mode. The alternative mode is toggle; this switches the relay each time the button is pressed. Modes can be set for each individual relay.

9. Relays

There are eight relays that can be individually turned on or off. Each relay can switch up to 10A at 277V.

WARNING: This device must NOT be connected to AC Mains voltages without the proper advice from a qualified electrician. Mains voltages are Lethal you have been warned.

9.1. Implemented Escape Codes & I2C commands

In the table columns indicate:

S is the VT100 standard, Y is standard, P is partially standard, N is non-standard

Code is the escape sequence to invoke the command

DEF column shows the default values from the factory, and F in this column will indicate that this value will be stored when writing the system to block 0

I2C column is the I2C command when in this mode ** Note the command values are decimal ** The notation used is described in the text

S	Code	DEF	I2C	Name	Description
	Relay				
N	esc[<num>A esc[<num>,<time>A		10 <on><16 bit time>	Relay A	Turns on or off relay, when <num> is 0 the relay is turned off, when it is 1 the relay is turned on. The alternative function gives timed on or off. The value of time can range from between 1 to 65000. A value of 0 will be ignored. The value gives approximately 5 seconds when time is set to 68, thus the range can vary from 74mS for a count of 1 to 1.3 Hours. The time is approximate and will slow down if there is a lot of serial activity. **NOTE: for no delay use 1 NOT 0** Example: To turn the relay on in 1 minute form issuing the command: esc[1,810A To turn the relay off in 5 minutes time: esc[0,4054A
N				I2C Note	There is no alternative commands for I2C, this is specified by FOUR bytes, the command, on or off, and two bytes to specify the delay. High byte first. Taking the above timing examples: To turn the relay on in 1 minute form issuing the command: s 10 1 3 42 p (3 and 42 are high and low representations of 810) To turn the relay off in 5 minutes time: s 10 0 15 214 p
N	esc[<num>B esc[<num>,<time>B		11 <on><16 bit time>	Relay B	As A

N	esc[<num>C esc[<num>,<time>C		12 <on><16 bit time>	Relay C	As A
N	esc[<num>D esc[<num>,<time>D		13 <on><16 bit time>	Relay D	As A
N	esc[<num>E		14 <on><16 bit time>	Relay E	As A
N	esc[<num>F		15 <on><16 bit time>	Relay F	As A
N	esc[<num>G		16 <on><16 bit time>	Relay G	As A
N	esc[<num>H		17 <on><16 bit time>	Relay H	As A
N	esc[<num>V		30	Return delay value	This will return the delay as set by the relay commands. It may be useful to test this to see how long it will be before switching occurs. A value of 0 means that the relay is not about to switch. The <num> refers to which relay: 1=A, 2=B, 3=C, 4=D, 5=E, 6=F, 7=G, 8=H See command esc[<num>A for a description of what value is placed on the number.
				I2C Notes	The number is returned as 2 bytes, this is a 16bit number with the high byte being sent first. s 30 <n> r g-2 p (see text for use on this notation), <n> is the relay to get the delay value from.
N	esc[n		18	All off	Turns all relays off
N	esc[o		19	All on	Turns all relays on
N	esc[<num>N		20, num	Binary on	Turns on, one or more relays depending on the number specified. The number is specified in decimal but the binary position is the relay that will be switched on or off. As an example to

					switch on relays A,C and G: H,G,F,E,D,C,B,A 0,1,0,0,0,1,0,1 The above table represents all of the 8 relays, to switch on the specified relays place a 1 under the letter and this will give the 8 bit binary number 01000101 which is 69 decimal. using the command esc[69N will switch on relays A,C and G and turn off all of the other relays.
				I2C Notes	Send a byte after the command, in the example given: s 20 69 p (see text for use on this notation)
	System Settings				
N	esc[<num>e	Off	n/a	ACK	Sets and activates the ACK character as an alternative to hardware handshake. Some commands will send a character to the receiving device, this can be used to determine when the command has finished. The host can wait for this character before sending the next command. To set up the ACK to send 'X for example the following is used: esc[88E 88 Is the ASCII code for 'X'. To turn off the ACK system use esc[0E (esc[zeroE).
N	esc[?27D		80	Writes defaults to EEPROM	Any variable indicated by an 'F' in the 'DEF' column of this table will be written to the EEPROM. This means that when the device is reset or switched on the values set by the user will take effect (whatever the user has set them to) rather then the factory defaults. This will also apply to the Baud rate which will change from being automatically detected to the Baud rate that is set when this command is invoked.
N	esc[?27F		81	Resets EEPROM back to factory defaults	Using this command will write the defaults back to the EEPROM. This is the only way that the Baud rate can be reset back to automatic. The device will reset after this command.
N	esc[<num>i		82 <addr>	Set I2C Address	This will set the I2C address and must be followed by EEPROM write. The new address will not take effect until reset. The default address is 0x64 (100). As an example to set the address to 0x42 (66) : <esc>[66i<esc>[?27D Both of the commands must be specified although the write to EEPROM command does not need to directly follow this command. The new address will take effect when the device has been restarted. IMPORTANT An even number MUST be specified to give a valid 8 bit address, the device will

					ignore odd addresses.
				I2C Notes	When used with I2C, there is no need to follow this with command 80, the device does however need a reset for the new address to be used.
N	esc[?31d		83	Device ID	Returns device ID
				I2C Notes	The device ID is returned as 2 bytes, this is a 16bit number with the high byte being sent first. s 85 r g-2 p (see text for use on this notation)
N	esc[?31y	On F	n/a	Com Flag (0n)	When communication is established with the device, i.e. at switch on or when the auto Baud mechanism has selected the correct Baud rate, ASCII code 42 '*' is sent to indicate this has happened. This commands switches this on, it is on by default
N	esc[?31n		n/a	Com Flag (off)	Switches the flag off see esc[?31y
N	esc[?31f		84	Firmware version	Returns firmware version, this is two numbers separated by a period
				I2C Notes	The firmware version is returned as 2 bytes, major version followed by minor version s 84 r g-2 p (see text for use on this notation)
P	escc		85	Reset device	Carries out a software reset
P	esc[?32m		n/a	IR Setup	This command displays the IR set up menu
P	esc[?32t		n/a	IR Test	The device will go into the IR mode. Its main purpose is for testing the IR set up but could be used if the IR is used in conjunction with the USB. The only exit from this command is to reset the device by using the DTR line or cycling the power.

Notes