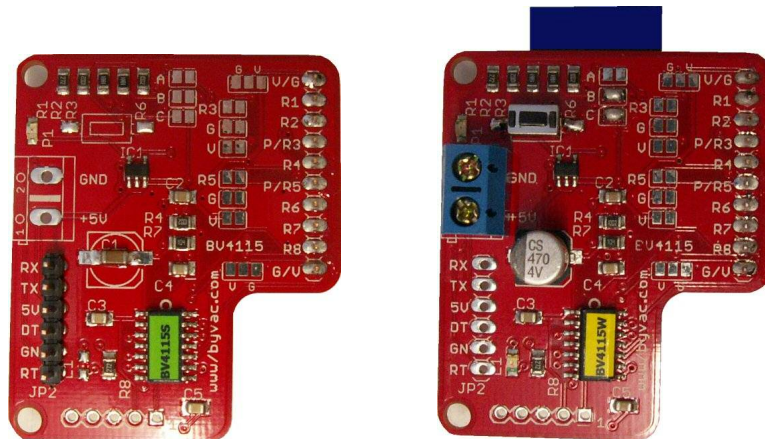


Serial (Wi-Fi) Relay Controller**BV4115****BV4115****Serial Relay Controller with Optional wifi**

Product specification

August 2015



Serial (Wi-Fi) Relay Controller

BV4115

Contents

1.	Introduction.....	3
2.	Features.....	3
3.	Physical Description	3
3.1.	Power Supply.....	3
4.	Serial Interface	3
4.1.	Hand Shaking	4
5.	Relay Interface.....	4
5.1.	2 Way Example.....	4
5.2.	4 Way Example.....	4
5.3.	8 Way Example.....	4
6.	Operation	4
7.	Wi-Fi Device (BV4115W)	4
8.	Device Parameters.....	4
8.1.	Address.....	5
8.2.	Polarity	5
8.3.	ACK character.....	5
8.4.	NACK character.....	5
8.5.	Baud Rate	5
8.6.	CR Character (EOL)	5
8.7.	Scale	5
9.	Commands	6
9.1.	Serial.....	6

Serial (Wi-Fi) Relay Controller

BV4115

Rev	Change
August 2015	Preliminary

1. Introduction

The BV4115 provides a serial interface to a standard relay board allowing on/off of individual relays and timed on and off.

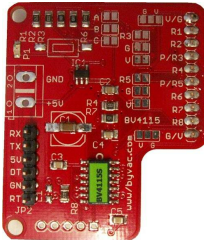
It can be connected directly to most 2, 4 and 8 way relays using the on board solder jumpers.

There are two versions. BV4115S for control via a serial port and BV4115W for control through a wifi socket.

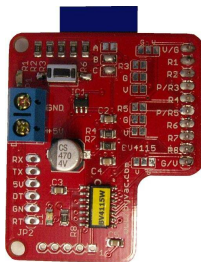
2. Features

- Wide voltage range 3.3V to 5.5V
- Up to 8 relays
- Timed on / off up to 48 days
- Addressable many devices can share a single serial bus
- Fixed Baud initial rate 115200, user changeable
- Use with high on or low on
- User configurable
- Wifi option

3. Physical Description

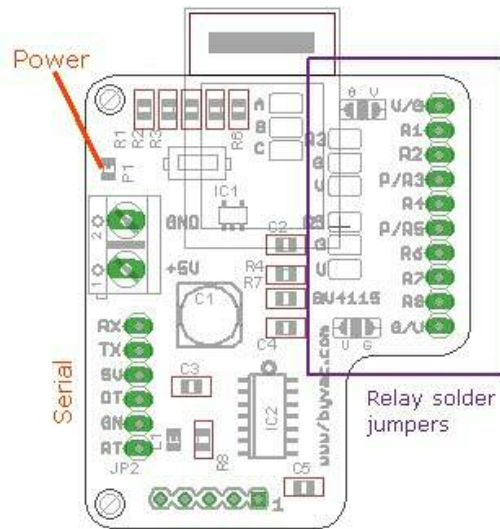


BV4115S uses the serial connection



BV4115W the serial connection is via the Wi-Fi link

Both the serial and Wi-Fi devices use the same serial commands



The interface has a 2.54mm spaced 2 rows by 5 pins.

Pin	Description
1	RX
2	TX
3	5V (up to 6V acceptable)
4	GT – reset, pulse low
5	GN Ground
6	RT(see wifi)

Serial Connector

3.1. Power Supply

The device works on 3.3V but there is an on board 3.3V regulator and so it can operate from 3.3V or up to 6V.

The relays and serial interface are 3.3V logic. This will work for 5V logic systems and most Opto isolated relay boards.

5V relays should be used and they will be supplied directly from the 5V pin or the screw terminal.

4. Serial Interface

The serial interface is via the TX and RX pins, By default the Baud rate is set at 115200. This can be changed by altering a value in the EEPROM via the EEPROM write command.

The protocol follows the standard 1 start bit, 8 data bits and 1 or 2 stop bits. All data (with the odd exception) is ASCII coded so that is the number 75 is sent via the serial interface then this will be TWO bytes '7' and '5', the actual value of the bytes will be 55 and 53, that being the ASCII codes for 7 and 5.

Serial (Wi-Fi) Relay Controller

BV4115

4.1. Hand Shaking

This has been avoided by the use of ACK. A serial command consist of a packet <address><command and data><EOL>

All packets are less then the buffer size and so the device will not respond until a full packet is received. When the device receives a packet it carries out the command and THEN send the ACK back to the host. The host should not send any commands until the ACK is received. This method of communication avoids the need for a hardware handshake that is the cause of so many serial problems.

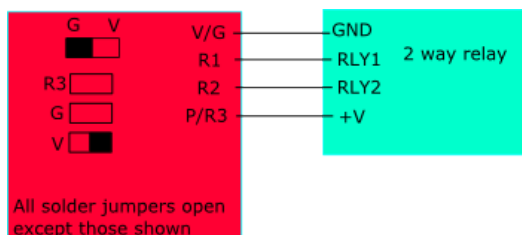
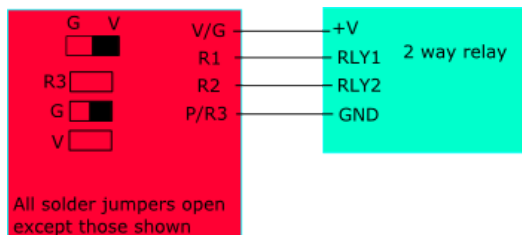
In addition, because the device is addressed more than one device can use the same serial bus.

5. Relay Interface

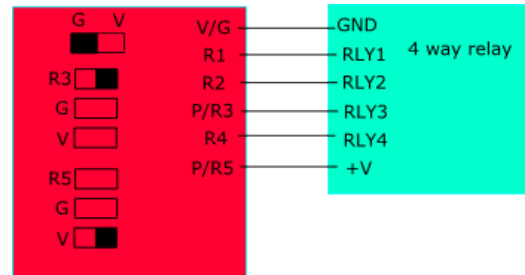
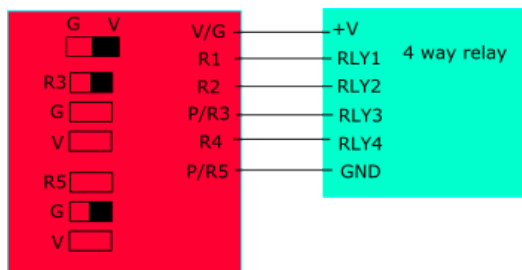
There are various solder jumper options to cater for the many variations in relay board. This will allow in most cases a direct connection from this device to the relay board.

If this is not possible then wiring can be used.

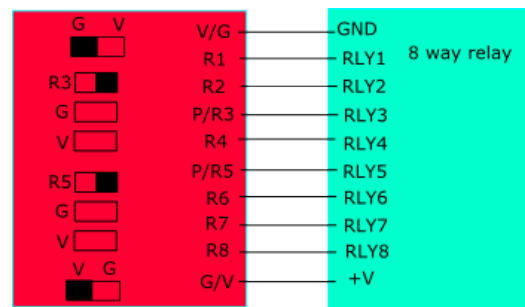
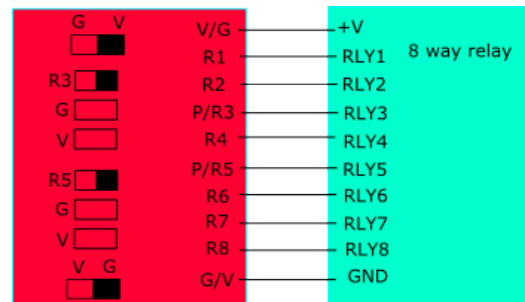
5.1. 2 Way Example



5.2. 4 Way Example



5.3. 8 Way Example



6. Operation

When connected to a serial interface at 115200 Baud the device will initially output '*'. If connected to a terminal then type dV<ent> and this will return the firmware version.

All operations follow this format, for example to activate relay 1 this would be da1,0<ent> for more information about this see the back of the text.

7. Wi-Fi Device (BV4115W)

The device is still essentially a serial device, when used with wifi the wifi device will act as the serial link and so all serial commands apply; just through the wifi instead of a COM port.

See:

<http://www.byvac.com/index.php/BV4115>

and the quick start guide for initial set up using a browser.

8. Device Parameters

The EEPROM contains values that control the way the device behaves. All of the values can be changed by the user using the serial interface.

Serial (Wi-Fi) Relay Controller

BV4115

The EEPROM consists of 255 bytes and in general the first 16 bytes are used by the system

Adr	Default Value	Description
0	0	System Use
1	100	Device address 'd'
2	1	Polarity
3	255	not used
4	6	ACK
5	21	NACK
6	255	Not used
7	3	Baud rate Code
8	13	End Of Line (EOL)
9	1	Scale low byte
10	0	Scale high byte
14	106	Device address copy
250	106	Device address copy

Table 1 System EEPROM use

8.1. Address

These EEPROM locations contains the device address. By convention the address is set to values between the values 97 to 122, no checking is made by the device so setting values outside this range may or may not work.

For security the address is stored in three places and to change the address of the device at least two of the locations need to be set otherwise the device will detect the anomaly at start up and revert to the majority value.

Normally to change the address of a device locations 1 and 14 are both changed. The device will detect this at start up and change the address in location 250 to match.

8.2. Polarity

Some relay modules require a logic 1 to activate others require a logic 0. This is set to the active value of the relay. If set to 1 then the relay will be activated by a logic 1, if set to 0 then it will be activated by 0.

8.3. ACK character

By default this is 6 but can be changed using the EEPROM Write command. The effect will not be implemented until the device is reset.

8.4. NACK character

By default this is 21 but can be changed using the EEPROM Write command. The effect will not be implemented until the device is reset.

8.5. Baud Rate

The Baud rate has the following values:

0. no valid
1. Baud rate is fixed at 2400
2. Baud rate is fixed at 4800
3. Baud rate is fixed at 9600
4. Baud rate is fixed at 14400
5. Baud rate is fixed at 19200
6. Baud rate is fixed at 38400
7. Baud rate is fixed at 57600
8. Baud rate is fixed at 115200 (default*)

8.6. CR Character (EOL)

By default this is 13 which is the standard ASCII CR and the whole serial protocol relies on this being at the end of every command. It may be that this is unsuitable in some systems and so this can be changed.

8.7. Scale

By default the timing value used by the device is milliseconds. This value can be scaled using a setting in the EEPROM.

The scale value is a 16 bit word, thus if it is set to 1000 then the timing value will be 1 second, if set to 60000 then the timing value will be 1 minute. As the maximum value a 16 bit word can hold is 65000 then the scale maximum is 65 seconds (approximately). So when a timer value of 65000 is given this gives a delay of about 48 days.

The timings are approximate and will vary. If accurate timings are needed then tests should be carried out.

Some EEPROM values for various scales

Scale	(low)	(high)
10	10	0
100	100	0
1000	232	3
60000	96	234

The formula is: device the scale value by 256, that is the high byte, the remainder is the low byte.

Serial (Wi-Fi) Relay Controller

BV4115

9. Commands

9.1. Serial

All serial commands start with the address, for convenience only the command values have been chosen to be in the printable range. This makes debugging and experimentation easier.

A serial transaction is a packet that has the following elements:

<address><command and data><EOT>

The address for this device is the same as the I2C address by default 100 ('d'). All devices connected to the bus listen out for their address as the first byte of a packet.

<command and data> The next byte will be a command as indicated in the table below followed by any necessary data. There is no separator for the first byte after a command but subsequent data items should be separated by a comma or space unless the command says otherwise.

For commands that require more than one byte for example write to EEPROM then a comma is used, in this example: dW5,21 the value of 21 is written to address 5.

Serial	range	Default Value	EEPROM Location	Description																		
a to h	1,65535			Turns a relay on or off <address><'a' to 'h'><0 or 1>,<0 or n><EOL> The letter 'a' to 'h' refers to the relay, this is followed by '1' to turn the relay on or '0' to turn it off. The next value is the action time or when the action is required. da1,1000 Example relay 1 on in 1 second db0,0 Example relay 2 off immediately The above examples are using the default scale of 1mS																		
r				Gets relay timer value <address><'r'><relay number 1 to 8><13> This is for when a long timer value has been set, the host may want to know how long to go before action commences. dr3 Example get timer value for relay c																		
o				Turn off all relays Turns off all relays with a single command do																		
i	0-16			Relay Status This will return a byte representing the status of the relay port. Bit 1 is on and bit 0 is off: <table border="1"><tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>Rly</td><td>h</td><td>g</td><td>f</td><td>e</td><td>d</td><td>c</td><td>b</td><td>a</td></tr></table> As an example if just relay b and d were on and the rest of the relays were off then the following byte would be returned: 00001010 which is 0xa or 10 in decimal. The value is the status of the port that is driving the relays. This may not necessarily be the actual relay itself.	Bit	7	6	5	4	3	2	1	0	Rly	h	g	f	e	d	c	b	a
Bit	7	6	5	4	3	2	1	0														
Rly	h	g	f	e	d	c	b	a														

Serial (Wi-Fi) Relay Controller**BV4115**

System				
W	n=0-255 m=0-255			<p>Write to EEPROM</p> <p>This will write a single byte to an EEPROM location</p> <p>I2C Example write 23 to location 7</p> <p>s 0x91 7 23 p</p> <p>or</p> <p>bus.i2c(0x34,[0x91,7,23],0)</p> <p>See www.pichips.co.uk and 'notsmb' for an explanation of the above nomenclature.</p>
R	n=0-255 m=0-255			<p>Read from EEPROM</p> <p>Reads a single EEPROM values from a given address.</p> <p>Example</p> <p>To read from location 3:</p> <p>s 0x90 3 r g-1 p</p> <p>or</p> <p>bus.i2c(0x34,[0x90,3]1)</p> <p>See www.pichips.co.uk and 'notsmb' for an explanation of the above nomenclature.</p>
D				<p>Device ID</p> <p>Returns two bytes representing a 16 bit number, high byte first</p> <p>s 0xa1 r g-2 p</p> <p>or</p> <p>bus.i2c(0x34,[0xa1],2)</p> <p>See www.pichips.co.uk and 'notsmb' for an explanation of the above nomenclature.</p>
C				<p>Reset</p> <p>Resets an individual device. This is a soft reset.</p> <p>A soft reset will normally be the same as a reset at start-up but this may not always be the case.</p> <p>Example</p> <p>s 0x95 p</p> <p>or</p> <p>bus.i2c(0x34,[0x95],0)</p> <p>See www.pichips.co.uk and 'notsmb' for an explanation of the above nomenclature.</p>
E				<p>EEPROM reset</p> <p>Will reset the EEPROM back to the default values.</p> <p>bus.i2c(0x3d,[20],0)</p>
V				<p>Version</p> <p>Returns the firmware version as two bytes</p> <p>value = bus.i2c(0x3d[0xa0],2)</p>