

---

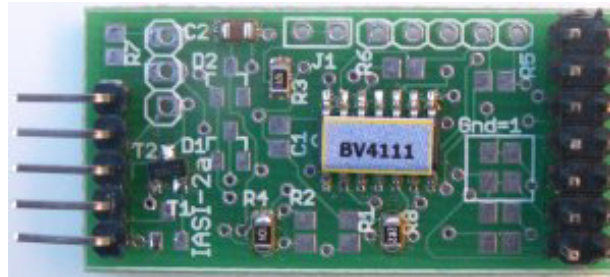
---

# SV3 Relay Controller

---

---

# BV4111



## BV4111 Serial Relay Controller

Product specification

Nov. 2012 V0.a

**SV3 Relay Controller****BV4111****Contents**

1.	Introduction .....	4
2.	Features .....	4
3.	Electrical interface .....	4
3.1.	Serial interface.....	4
3.2.	Relay Connector .....	4
3.3.	Jumper.....	4
3.4.	Factory.....	4
4.	Command Set .....	4
4.1.	Turn on / off relay .....	5
4.2.	Timer value .....	5
4.3.	All off.....	5
5.	EEPROM values for this device.....	5
5.1.	Scale (locations 12 & 13).....	5
5.2.	Invert (location 14).....	5
6.	Typical Relay.....	6
7.	Revisions SV3 .....	7
8.	Introduction to SV3.....	7
9.	SV3 Electrical Interface .....	7
10.	Serial Connections .....	7
11.	Start Up .....	8
12.	Command Format .....	8
13.	Numbers.....	8
14.	Non/Inverted Mode .....	8
15.	Data Packet .....	9
16.	Receiving Data .....	9
17.	Non-Addressable Commands.....	9
17.1.	Command 1 .....	9
17.2.	Command 2 .....	9
17.3.	Command 3 .....	9
18.	Addressable Commands.....	9
18.1.	Summary .....	9
18.2.	Write to EEPROM .....	10
18.2.1.	Address.....	10
18.2.2.	ACK character.....	10
18.2.3.	NACK character.....	10
18.2.4.	Turn off Error reporting.....	10
18.2.5.	CR Character .....	10
18.2.6.	Multi .....	10
18.3.	Read EEPROM .....	10
18.4.	Device Number.....	10
18.5.	Toggle Inverted.....	11

---

---

# SV3 Relay Controller

---

---

# BV4111

18.6.	Reset .....	11
18.7.	Version.....	11
19.	Error Codes.....	11
20.	Connection and Configuration.....	11
20.1.	Multiple Devices .....	11
21.	Restoring Factory Defaults .....	11

# SV3 Relay Controller

# BV4111

Rev	Change
Nov 2012	Preliminary

## 1. Introduction

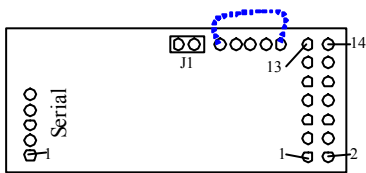
The BV4111 provides a serial interface to a standard relay board allowing on/off of individual relays and timed on and off.

## 2. Features

- Wide voltage range 2.5V to 5.5V
- Up to 8 relays
- Timed on / off up to 48 days
- Addressable many devices can share a single serial bus
- Automatic Baud rate detection up to 115200
- Use with high on or low on
- User configurable

## 3. Electrical interface

There are two main connections to the board. The serial interface and the relay interface.



### 3.1. Serial interface

Pin	Description
1	RX
2	TX
3	+V
4	n/c
5	Gnd

Pin number 1 is the nearest to the edge of the PCB

Most relays require 5V and so V+ will be 5V as this voltage can be conveniently taken from the 14 way connector at the other end of the PCB.

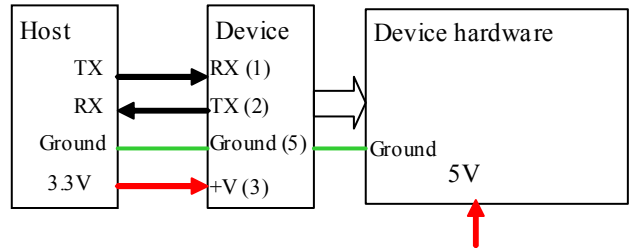
### 3.2. Relay Connector

Pin	Use	Use	Pin
13	RLY7 'g'	RLY8 'h'	14
11	RLY5 'e'	RLY6 'f'	12
9	n/c	n/c	10

7	RLY2 'b'	+V	8
5	GND	RLY3 'c'	6
3	RLY1 'a'	RLY4 'd'	4
1	n/c	n/c	2

n/c is no connection

The relay outputs are normally high by default and when active they go low.



The above shows how to drive 5V relays from 3.3V logic as the serial interface is isolated from the 5V supply all logic levels are 3.3V

### 3.3. Jumper

This is used if more than about 3 devices are connected to the same serial bus then the PCB track on this jumper (J1) may need cutting for more details on multiple devices see the SVD Description document.

### 3.4. Factory

There are a set of 5 holes to the top of the PCB, the left most one has a square pad. For factory reset connect the square pad to the right most hole as indicated by the blue dotted line. For details of the procedure refer to the SV3 Description document.

## 4. Command Set

*NOTE there are two distinct command sets. The system command set and this command set. For the system command set refer to the SV3 Description document, at the back of this text.*

### Default address 100 ('d')

Command Set	
'a' to 'h'	Turn on / off relay
'r'	Read relay timer
'o'	Turn off all relays

**Table 1 Command Set**

All of the above commands require a device address to be specified as described in the SV3 description document and command examples are shown using the default address of 100.

# SV3 Relay Controller

# BV4111

The timer value is 1mS and can be scaled by setting two values in EEPROM. Details of this are at the end of this text.

## 4.1. Turn on / off relay

There are 8 commands 'a' through 'h' (or 97 through 104) that correspond to the relay output pin as shown in the connector table.

Each relay can be turned on / off immediately or after a period of time. The command format is:

<address><'a' to 'h'><0 or 1>,<0 or n><EOL>

<0 or 1> is 0 for off and 1 for on.

<0 or n> is 0 for immediately and n is a number from 1 to 65500, this is the time for the action (on or off) in mS. Observe that the numbers are specified as ASCII strings and so "1" is in fact 49 not 1

### Examples:

- Turn on relay 'c' immediately:

```
100 99 49 44 48 13
```

On a terminal this would look like:

```
dc1,0<return>
```

- Turn on relay 'a' in 20 seconds:

```
100 97 49 44 51 48 48 48 13
```

On a terminal this would look like:

```
da1,30000
```

## 4.2. Timer value

Gets the current timer value for the relay. It may be useful for the host to know when or just how far into the action the relay has got. Some appropriate action may not be performed until the relay has energised for example. In this case the relay is specified as a number from 1 to 8

The command format is:

<address><'r'><relay number 1 to 8><13>

As an example to get the relay timer value for relay 'c' on a terminal would be:

```
dr3<return>
```

This would be the following bytes:

```
100 114 51 13
```

The value returned is an ASCII coded decimal number. For example if the value returned was 251 then the output would be:

```
50 53 49 6
```

The ending 6 is the ACK.

## 4.3. All off

This is just a convenient way of turning all of the relays off with one command.

Example on a terminal:

```
do<return>
```

This is bytes:

```
100 111 13
```

## 5. EEPROM values for this device

This device has the following values set in EEPROM that will affect the way it behaves by default.

When changing a value the device will need resetting for the changes to become effective.

### 5.1. Scale (locations 12 & 13)

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

EEPROM Address	Value
12	Low byte
13	High byte

EEPROM byte locations.

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 is 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	12 (low)	13 (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.

### 5.2. Invert (location 14)

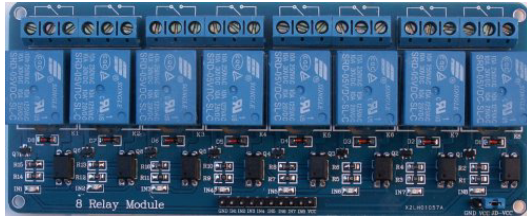
Under normal circumstances this will be set to 1. This means that when the relay is off the output is high and it is brought low to activate the relay. This matches the type of relay board shown in the next section.

For relay boards that work the opposite way. this value can be set to 0.

# SV3 Relay Controller

# BV4111

## 6. Typical Relay



This is an 8 way relay with 10 connections, two are for the supply and the other 8 are for connecting to a controller. When left unconnected or high the relays are off, when a relay pin is connected to ground the relay comes on.

The following will connect it to the controller:

Controller		Relay
Pin	Name	Name
1	n/c	
2	n/c	
3	RLY1	IN1
4	RLY4	IN4
5	GND	DND
6	RLY3	IN3
7	RLY2	IN2
8	+V	[1] VCC
9	n/c	
10	n/c	
11	RLY5	IN5
12	RLY6	IN6
13	RLY7	IN7
14	RLY8	IN8

[1] If using a 5V logic host then this will go to the relay VCC pin as the relays require 5V to operate.

If using a 3.3V host then this pin will not be used and the relays will be connected to a separate 5V supply.

# SV3 Relay Controller

# BV4111

## 7. Revisions SV3

Rev	Change
Nov 2012	Update from IASI
Nov 2012	Version 1.0

## 8. Introduction to SV3

Serial Version 3 is an enhancement of the original IASI protocol that allows smaller more compact packets and thus increasing the efficiency. It is backward compatible but some of the superfluous commands have been removed. The default now is to connect to a microcontroller rather than RS232 and some devices do not support RS232 voltage levels any more.

The SV3 is a common standard that makes it much easier to control and use hardware from either a standard communication interface (terminal) or a microcontroller.

It is based on a very simple command set that does not require hardware handshaking and is therefore very easy to set up.

All of the transfers (unless otherwise stated) to and from the host are in text. This makes it easier to interface to common programming languages such as VB or Python.

## 9. SV3 Electrical Interface

The device has very simple requirements. A power supply, transmit and receive lines as shown in table E1.

The interface will always interface to 5V logic and may also have a provision for receiving 12V RS232 signals. A five pin connector is used with normally only 3 or four pins being connected at any one time.

There **may** be two receive lines, pin 1 receive line will accept normal 5V logic as presented by a microcontroller pin or UART and pin 4 will accept positive and negative voltages up to 15V that are normally present on a standard RS232 interface. Pin 4 will also invert the logic which is also normal for this interface.

The Baud rate is automatically detected at start up on the first or second receipt of Carriage Return (#13). The detection is from a fixed set of standard Baud rates normally: 9600, 14,400, 19,200 and 38,400 and sometimes 115200.

The transmit pin has an open collector output that has a pull-up resistor on board connected through a jumper. This allows more than one device to be used on the same serial line, only one jumper should be shorted. See the section on multiple devices for further information.

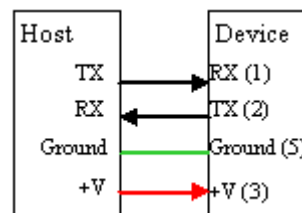
The data byte is represented by 1 start bit, 8 data bits and 1 stop bit. The idle state is normally high so a stop bit begins by pulling the serial line low.

Pin	Description
1	RX receives data from the host
2	TX sends data to the host
3	+V 2.5 to 5.5V (but see data sheet for that device)
4	RX input, can accept $\pm 15V$ and $-15V$ will also invert the data- <b>not implemented on all devices</b>
5	Ground

### 5 Pin connector

The +V pin is in the centre and so if the plug is inadvertently connected backwards there is no consequence.

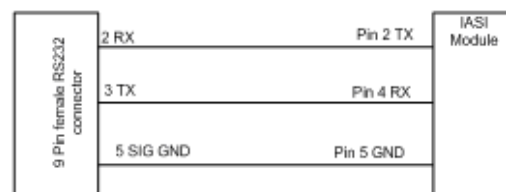
## 10. Serial Connections



The device is designed to work with a microcontroller or similar device that outputs logic level signals, in turn the device will also output a logic level signal.

The power supply to the device depends on the device, the electronics will work from 2.5 to 5.5V, however it may be connected to a peripheral that requires more voltage, relays and some LCD displays require 5V for example. The logic level of the putput pin 2 will be the same as the +V pin 5.

There is an alternative input provided on some devices for the RS232 standard on pin 4. This is just the input, the output on pin 2 will remain at logic levels. The input on pin 4 can accept up to + and - 15V. It will also invert the input as this is what is needed for RS232.



**Figure 1 Connection to a PC**

# SV3 Relay Controller

# BV4111

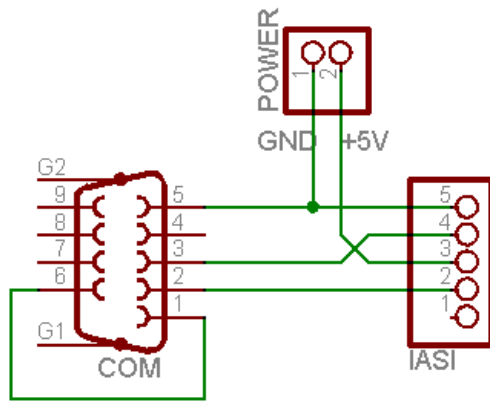


Figure shows the connections to a 9 pin D type connector found on some PC's. This is the RS23 connection. The RX line cannot be used for multiple devices in this configuration.

## 11. Start Up

The Start up procedure will establish the Baud rate of the host. The Baud rate will be selected from a table of standard Baud rates as follows:

1. 2400
2. 4800
3. 9600
4. 14400
5. 19200
6. 38400
7. 57600 [1]
8. 115200 [1]

[1] These two rates are not available on all devices.

The device will match the host baud rate by using the following procedure:

Send CR with a delay or 50ms between until the device sends back `\*`. This will normally take 3 CR's

Pseudo code:

```
for j = 1 to 5
```

```
  putc("\r")
```

```
  delay_ms(50)
```

```
  if getc() = '*' break
```

```
next
```

CR is 13 or 0x0d or "\r"

`\*' is 42 or 0x2a

The auto detect can be overridden with a fixed Baud rate by setting the appropriate value on the devices EEPROM. If multi (see eeprom) is set to 0 then any time CR is sent on its own, the device will respond with `\*`.

## 12. Command Format

All devices have an address which is one byte and can be any value within the range 97 to 122. The user specifies the address and so can be set up for multiple devices.

The **default address is given in the data sheet for the individual device** and all devices must be addressed although there are some global commands that address all of the devices at once.

There are basically two sets of commands, those which are common to all devices, these are described here and normally be in the range 65 to 90 and those that are device specific which are described in the devices datasheet.

This section deals with the system commands.

The command format in general is therefore:

```
<address>{commands}<EOL>
```

The device will always (unless reset) respond with ACK which by default is 6. EOL is the end of line character which by default is 13

## 13. Numbers

Some commands are followed by data. Where this is the case and a number is required it is sent as text. Numbers are also received as text.

As an example if a command required two numbers say 12 and 120, the command format may look something like below.

```
<address><command><"12"><delimiter><"120"><EOL>
```

The address would be a single byte corresponding to the devices address. The command would also normally be a single byte. The "12" would be 2 bytes, and "120", 3 bytes.

The delimiter between the numbers is normally a `,` but can be a space or any non digit, just as long as it separates the two numbers.

If the address was 99 and the command say 50 then the actual bytes sent to the device would be:

```
99 50 49 50 44 49 50 48 13
```

Just to re-iterate, the "12" is sent as text so this is 2 bytes with values of 49 and 50 that correspond to the ASCII values `1` and `2`.

This is the general `rule` however for efficiency this may be overridden by the individual device, if that is the case it will be clearly explained in the data sheet for that device.

## 14. Non/Inverted Mode

As previously mentioned the device is capable of operating with a standard RS232 communication port (inverted) and a microcontroller (non-inverted - default). The device will accept either signal as it has two inputs but it only has one output.



# SV3 Relay Controller

# BV4111

The output can be set as inverted to comply with RS232 but doing so will prevent any other device from working on the same bus. The inverted mode is not the normal way of working as this protocol allows many devices to share a serial bus.

The recommended method if an RS232 9 pin connection has to be used is to interface through a conversion device to make the RS232 behave as a microcontroller word. A MX232 for example.

## 15. Data Packet

A packet consists of a series of bytes followed by byte 13 (Carriage Return CR). It follows then that all commands must end with CR for the device to accept the command. The host should then wait for the device to return ACK (6) or NACK (21) before proceeding.

Using this protocol means that no hardware handshaking is required, for a microcontroller this means two less lines are required. There are some implications however and that is that the device must wait until the full command is received before acting on it. The buffer size default for a device is 64 bytes but this may vary. The buffer cannot be exceeded so will restrict some forms of communication.

## 16. Receiving Data

Some commands will return information. Where this happens the ACK or NACK will be received at the end of the received data. The ACK will instruct the host that it is okay to collect the data. See the Version command for a simple example of this behaviour.

## 17. Non-Addressable Commands

The interface is completely software driven, all commands and configuration are done through a serial interface. The only exception to this is the hardware factory default restore.

The following commands do not require a device address. These are 'global' in that all devices on the bus will respond or take action.

### 17.1. Command 1

This is the discovery command its purpose is to find out what devices are on the bus in an automated way. It can also be used to check that the devices exist as expected. The command requires two bytes to be sent:

1 13

Following this each device will place its address on the bus in turn using the address value in a delay calculation.

Each device has 2mS to place its address on the bus, the lowest address 97 will send its address immediately where as a device with address 122 will take  $(122 - 97) * 2mS = 50mS$ .

Example, two devices are on the bus:

1 13

99 102 <output from devices>

No ACK is given for this command so the host must wait at least 50ms for all the devices to respond.

### 17.2. Command 2

Toggle inverted. The output signal will be inverted, this is mainly to cater for devices that are connected to RS232 that requires an inverted signal.

### 17.3. Command 3

This will reset all devices as if they had just been powered up. Following this command one or two CR is required to establish the Baud rate.

## 18. Addressable Commands

The following commands require the device address to be sent first, only the device with that address will respond.

The command byte values have been chosen so that they are in the printable range. This makes it easier to debug using a simple terminal emulator. To read the first 6 bytes of the EERPOM for example would simply be:

aR0,6<CR>

The device will then output readable text.

### 18.1. Summary

Command	Description
'W'(87)	Write to EEPROM
'R'(82)	Read EEPROM
	Reserved
'I'(73)	Toggle inverted
'C'(67)	Reset device
'V'(86)	Version
'D'(68)	Device ID

Note that examples will use the default address of 97.

The first few bytes of the EEPROM contain system information and can be changed with the above commands. The system details along with the EEPROM address is as follows:

EEPROM Address	Default Value	Description
0	0	0xff causes factory reset
1	97	Device address
2	6	ACK character
3	21	NACK character

# SV3 Relay Controller

# BV4111

4	0	Baud rate [1]
5	1	Error reporting 0 is off
6	13 or 0xfc	Default CR or end of line character
7	0	Multi
8		
9		
10		

Bytes up 10 reserved for device or future use.

[1] The Baud rate has the following values:

0. This is the automatic Baud rate. The actual Baud rate will be determined by the host when 13 is first sent.
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 115200

The latter Baud rate may not be available on all devices.

## 18.2. Write to EEPROM

This device has an internal EEPROM with an address range 0 to 255. The user can use this as general non-volatile storage but should refrain from using addresses below 10 as they may be used for the system.

The command has the following format:

```
<address><"eeprom
address"><delim><"value"><EOL>
```

If things do go wrong with any of the system values then a hardware factory reset can be performed to restore the EEPROM back to its default settings.

As an example to change the address from 97 to 102 the following bytes will be sent by the host:

```
97 87 49 44 49 48 50 13
```

A comma (44) is used as a delimiter to separate the EEPROM address from the value. EEPROM address 1 contains the device address.

**NOTE: When altering a system EEPROM setting a reset is required for it to take effect.**

### 18.2.1. Address

This EEPROM location contains the device address. It is important to set the address

between the values 97 to 122, no checking is made by the device.

### 18.2.2. ACK character

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

### 18.2.3. NACK character

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

### 18.2.4. Turn off Error reporting

By default error reporting is enabled and this will be reported and an output prefixed by Error, for example **'Error 2'**. This may get in the way of the program trying to control the device and so it can be disabled with this command. The effect will not be implemented until the device is reset.

### 18.2.5. CR Character

By default this is 13 which is the standard ASCII CR and the whole 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.

**Important: This does not apply to the automatic Baud rate detection that must be 13**

### 18.2.6. Multi

Where there is only one device, sending EOL (CR) on its own will get a '\*' response back. This is very reassuring when testing and using the device. However in a multi-device environment. This behaviour can have a detrimental effect if all of the devices try to do it at the same time. Setting this value to 1 will turn off this behaviour.

## 18.3. Read EEPROM

The EEPROM values can be read with this command given a starting address and the number of bytes to read.

```
<address><read eeprom><"start"><"#bytes">
97 82 48 44 49 54 13<host>
```

The output from the device will commence after receiving 13 and will consist of a string of data terminated with ACK.

The sting will be in the form of text delimited by ',' and all of the values will be decimal. An example of output for the first 5 bytes of EERPOM would be:

```
"0,97,6,21,0"<ACK>
```

## 18.4. Device Number

Returns a number representing the device product number as a string

# SV3 Relay Controller

# BV4111

97 68 13<host>

"4111"<ACK> Returned by device

## 18.5. Toggle Inverted

Pin 2 on the electrical interface that supplies the output information (Tx line), can be supplied inverted or non-inverted (at reset, start up). Inverted is used if the device is connected directly to an RS232 PC Com (where the interface is available) port and non-inverted is used when the device goes through a converter (BV201, BV101) or is connected to a microcontroller.

At reset the device is always in the non-inverted mode. Some device do not support this command in which case an error will be generated.

97 73 13 – output is now inverted

97 73 13 – output is now non-inverted

Just ACK will be returned by the device, but this will be an inverted ACK.

## 18.6. Reset

Resets an individual device. The baud rate will need establishing again after this command is used.

This is similar to command 3 but works on a single device. A soft reset will normally be the same as a reset at start-up but this may not always be the case. Obviously no ACK will be returned by this command.

## 18.7. Version

Returns the firmware version as a string in the format "H.L"

An example of the transaction would be:

97 86 13 <host>

49 16 19 <ACK>

## 19. Error Codes

Error codes will be displayed if they have not been switched off.

An error code is output as **text** followed by NACK, as an example if error 2 occurred then the output would be:

0x45 0x72 0x72 0x6f 0x72 0x32 0x15

Code	Description
2	Unknown command, the command issued is not in the command table for this device.
3	Bad device address, the address specified is outside the address range.
4	Bad number, out of range for the command specified

5	Incomplete command, usually because not enough bytes have been sent for the specified command.
6	reserved

## 20. Connection and Configuration

### 20.1. Multiple Devices

The output of the device is an open collector, this means that many devices can be connected to the same serial bus. The only proviso is that there needs to be a pull up resistor somewhere on the bus.

On all SV3 devices there is a built in pull up resistor connected to a jumper that has is shorted out by a PCB track. In practice around three devices can be connected to the same bus without regard to this jumper or pull up.

If more devices need to be connected then the cumulative impedance of all of the individual pull up resistors will be too great for the serial bus to overcome. In this instance the track between the jumper pins needs cutting except on one device so as to retain at least one pull up.

The above of course applies to the output from the device (input to the host). If this is not required then it doesn't matter about the pull ups.

As the normal idle state is high (+V) the pull up system works well as only one device at a time will pull the bud sown to create a signal. This however does not apply to the inverted mode where the idle state is low. In this mode only one device can be connected to the host input. This is why this mode is not recommended.

## 21. Restoring Factory Defaults

The configuration of an SV3 device is contained in the EEPROM. As the user has full access to this it is possible that it may render the device unreachable. If the end of line character has been accidentally changed to some unknown value it would take up to 256 attempts to find out what it was.

The default values for the eeprom can be reset by the following procedure.

1. Power down the device.
2. Use a shorting link (bit of wire – paper clip etc.) on the appropriate pins.
3. Power up the device, this will restore the factory settings.
4. Power down the device.
5. Remove the shorting link.

The shorting link position varies from device to device. See the data sheet for where it should go.

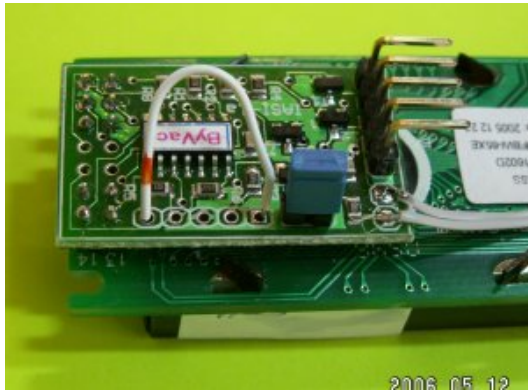
---

# SV3 Relay Controller

---

# BV4111

---



**Figure 2 Example Shorting link**