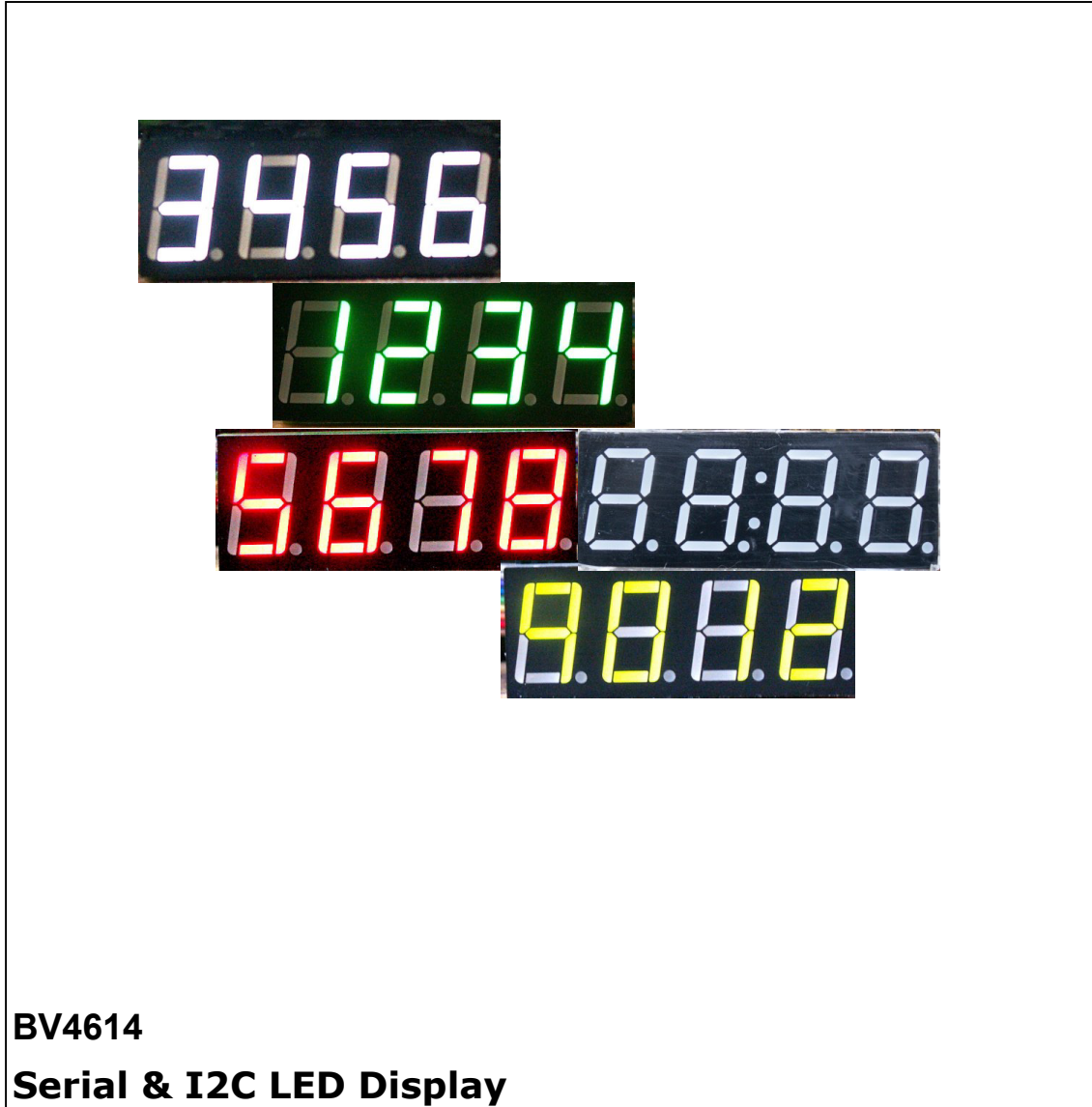

Serial & I2C LED Display

BV4614



Product specification

August 2013 V0.a

Serial & I2C LED Display

BV4614

Contents

1.	Introduction	3
2.	Features	3
3.	Electrical Specification	3
4.	Factory Reset	3
5.	Serial (SV3)	3
6.	Clock Type Display	3
7.	Serial Command set	3
7.1.	b	3
7.2.	i	3
7.3.	c	3
7.4.	s (see also n)	3
7.5.	n	3
7.6.	p	3
7.7.	e	3
7.8.	m	3
8.	I2C	3
9.	I2C Command set	3
9.1.	Command 1	3
9.2.	Command 2	3
9.3.	Command 3	3
9.4.	Command 4	3
9.5.	Command 5	3
9.6.	Command 6	3
9.7.	Command 7	3
9.8.	Command 8	3
10.	EEPROM Locations	3
10.1.1.	Changing the device Address	3
10.1.2.	Baud Rate	3
10.1.3.	Error Reporting	3
11.	Revisions SV3	3
12.	Introduction to SV3	3
13.	SV3 Electrical Interface	3
14.	Serial Connections	3
14.1.	TX Output	3
14.1.1.	Handshake	3
15.	Start Up	3
16.	Command Format	3
17.	Numbers	3
18.	Data Packet	3
19.	Receiving Data	3
20.	Non-Addressable Commands	3

Serial & I2C LED Display**BV4614**

20.1.	Command 3	3
21.	Addressable Commands.....	3
21.1.	Summary	3
21.2.	Write to EEPROM	3
21.2.1.	Address.....	3
21.2.2.	ACK character	3
21.2.3.	NACK character.....	3
21.2.4.	Turn off Error reporting	3
21.2.5.	CR Character	3
21.3.	Read EEPROM	3
21.4.	Device ID	3
21.5.	Toggle Inverted.....	3
21.6.	Reset	3
21.7.	Version.....	3
21.8.	Hello	3
22.	Error Codes.....	3
23.	Connection and Configuration.....	3
23.1.	Multiple Devices	3
24.	Restoring Factory Defaults	3
25.	I2C System Commands	3
25.1.	Command 0x90.....	3
25.2.	Command 0x91.....	3
25.3.	Command 0x95.....	3
25.4.	Command 0x96.....	3
25.5.	Command 0xA0.....	3
25.6.	Command 0xA1.....	3
26.	Hardware Reset.....	3
27.	Commands	3
27.1.	Sending a single command	3
27.2.	Sending a command with a parameter byte/s	3
27.3.	Receiving bytes from the salve.....	3
28.	System EEPROM	3
28.1.1.	Address.....	3
28.1.2.	ACK character	3
28.1.3.	NACK character.....	3
28.1.4.	Turn off Error reporting	3
28.1.5.	CR Character	3
29.	Trouble Shooting	3
29.1.	Pulse Stretching	3
29.2.	Last Read NACK	3
29.3.	Pull Up's	3

Serial & I2C LED Display

BV4614

Rev	Change
August 2013	Preliminary

1. Introduction

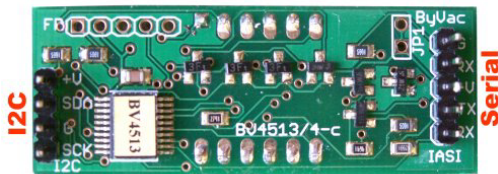
This is a 7 segment 4 digit display that has a serial AND an I2C interface for maximum flexibility.

The product is very simple to use and reduces the number of ports required from the host, the multiplexing and timing are all carried out on board so the user just has to supply the data.

2. Features

- Serial Interface
- I2C interface
- User definable on screen
- Individually set brightness
- Multiple devices can share the same data lines on the serial and the I2C interface
- Clear easy to see display
- Stackable
- 51 x 21 x 20mm including pins
- 2.5mA no displays on
- 10mA average all on

3. Electrical Specification



The back of the display has two connectors, and (Serial) connector and an I2C connector. By default the display is set to serial, when an I2C bus is applied to the I2C connector then the display will become an I2C display. See the I2C section.

The pin designations are marked on the PCB.

4. Factory Reset

The factory reset will restore the EEPROM device addresses. The two holes for this device are holes 1 and 2 of the pads marked FR above the I2C connector. Hole 1 is the square pad and hole 2 is the next one to it.

5. Serial (SV3)

The serial interface is set by leaving the I2C connector unconnected at start up. The Baud rate is initially set to 9600. By default (which can be changed by the user) the following will be displayed. This indicates that we are in serial mode.



The display is accessed by commands, preceded by an address. **The default address for this device is b (0x62)** To illuminate digit 0 with 3 for example the command **'bn0 3'** is sent

The detailed information about the serial command set is presented later.

6. Clock Type Display

The clock type display has two extra LED's between the 2nd and 3rd digit. They are controlled by the second decimal point so when the second decimal point comes on so does the two extra LED's

This is how the display is manufactured and so for a non-clock type physical masking (paint tape) of the two extra digits, or the decimal point are required.

7. Serial Command set

The serial commands are given as text because it is easier to understand and see, however the device only receives binary data so sending the address 'c' to the device is just receiving 0x63. This important to understand when sending data from a host other than a terminal. Also **all serial commands are terminated with CR (0xd)**.

Where a number is required for example 20, then this needs to be converted to ASCII before sending. To give an example, changing the global brightness to say 150 would require the following to be typed at a terminal:

'bb150'

If a microcontroller is connected then for this command it should send:

0x62 0x62 0x31 0x35 0x30 0xd

The two crucial factors to not here is that the number 150 is not sent as a single value of 0x96 but as three values '1' '5' and '0' also the 0xd must be present at the end of every command.

In return the BV4614 will return 0x6 to acknowledge that the command has been received and dealt with, or 0x15 if not.

Device default address is 'b'

Command	Device Command Set
b	Global brightness
i	Individual brightness
c	Clear (blank) display

s	Send byte to digit
n	Send number to digit
p	Decimal point
e	Writes current brightness value to EEPROM
m	Fixes current display

Table 1 Device Command Set

Table 1 is a command summary of all of the device specific commands. Note that these are all in lower case.

There are also another set of system commands that are in upper case, they enable to configuration of the device such as the address. Details of these are given later in the text under "SV3 Protocol"

In all of the examples the default address of 'c' is used

7.1. b

Name: **Brightness**

Command Parameters: **<n>(0-250)**

Typical Use **bb10**

Sets the brightness of all of the digits on the display, 0 is dim and 250 is bright. Any value greater than 250 will default to 250.

NOTE: The brightness varies considerably with the colour of the display and so this value has been extended to cover all types of display. Care must be exercised not to overdo the brightness as this may cause excessive heating of the electronics and reduce its life.

The format is:

<address><brightness>

7.2. i

Name: **Individual Brightness**

Command Parameters: **<n>(0-250)**

Typical Use **bi0,40**

Sets the brightness of an individual digit, 0 is dim and 250 is bright. Any value greater than 250 will default to 250.

The format is:

<address><i><digit 0-3><brightness>

7.3. c

Name: **Clear display**

Command Parameters: **none**

Typical Use **bc**

Clears (blanks) all of the display.

7.4. s (see also n)

Name: **Send byte to digit**

Command Parameters: **<digit><byte>**

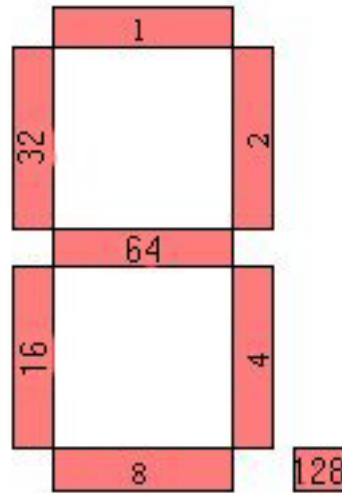
Typical Use **bs0,6**

This will place the byte at the given digit. Note there must be a space or comma between the digit number and the byte value.

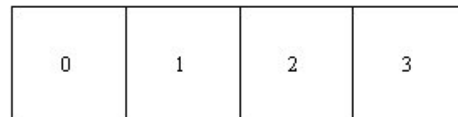
This command is used for making up special characters that are not in the rang 0 to 0xf (see the n command)

Digit is in the range 0-3

Byte is in the range 0-250



The byte vales are shown above and can be added together, so for example to display the number 3: 1+2+64+4+8 is added and so the result is 79. To display '3' onto the first digit the command **'bs0,79'** is used.



The digits are numbered from left to right as shown.

7.5. n

Name: **Send number to digit**

Command Parameters: **<digit><number>**

Typical Use **bn0,6**

There is a built in look up table and this command uses that in order to illuminate the correct segment to represent a number.

0-15 will display the corresponding number on the requested digit as follows:

0,1,2,3,4,5,6,7,8,9,A,b,C,d,E,F

16 will blank the digit.

7.6. p

Name: **Decimal point**

Command Parameters: **<digit><0 or 1>**

Typical Use **bp1,1**

The decimal points are treated separately from the other segments this is to enable a digit to be displayed with or without the decimal point. To illuminate the decimal point on the first digit 'bp1,1' is used to extinguish it 'bp1,0' is used.

7.7. e

Name: **Brightness**

Command Parameters: **None**

Typical Use **be**

The brightness value for each digit is stored in RAM but at switch on is retrieved from EEPROM and so by writing the current brightness to EEPROM with this command, it can be saved even after a reset.

WARNING: Do not use this command as part of your program as the EEPROM will only accept a certain number of write cycles.

7.8. m

Name: **Current Display**

Command Parameters: **None**

Typical Use **bm**

The current display, i.e. whatever is currently on the display at the time of using this command can be written to EEPROM. This will then be displayed at start up. The default for the serial interface is shown in the serial section so that on first use the user knows that the serial interface has been selected. The I2C default is different.

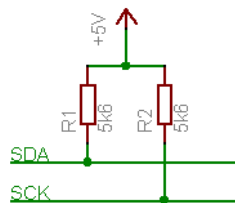
WARNING: Do not use this command as part of your program as the EEPROM will only accept a certain number of write cycles.

8. I2C

This display has two interface options. To use the I2C option the display must be connected to an I2C bus before power up. The pull up resistors on the I2C bus are detected and this will put the display in I2C mode. This is verified by the display showing the pattern as below.



The display shows this when in this mode. This can be changed by the user.



It is essential that the i2c interface is properly terminated otherwise the display will revert to the serial mode. This is only checked at start up.

When in I2C mode the **default 8 bit address is 0x62 the 7 bit address is 0x31**. For an explanation of 7 and 8 bit addresses see http://doc.byvac.com/index.php5?title=I2C_Troubleshooting under the Addressing section.

A full explanation of the system commands can be found later in the text

9. I2C Command set

The format used by this device consists of a command, this is a number, followed by other bytes depending on that command.

Device default address is 0x62 (0x31)

Command	I2C Command Set
1	Global brightness
2	Individual brightness
3	Clear (blank) display
4	Send byte to digit
5	send number to digit
6	Decimal point
7	Writes current brightness value to EEPROM
8	Writes current display to EEPROM

The method of writing to the device using the I2C protocol follows a consistent format, typically:

<S-Addr><Command><data..><Stop>

Where S-Addr is the start condition followed by the device address (0x62). Command is one of the commands given in the table. Data is one or more bytes and Stop is the stop condition.

Reading data requires a restart and this will be in the format:

<S-Addr><Command><R-Addr><data..><Stop>

The restart address will be one greater than the start address, thus if the start address is 0x62, the restart address will be 0x63. Again the data can be one or more bytes read from the device.

Where the host uses a 7 bit address then there will be an I2C command for writing and an I2C command for reading and the same 0x31 address will be used throughout.

9.1. Command 1

Name: **Brightness**

Format: <S-addr><1><0 to 250><Stop>

Sets the brightness of all of the digits on the display, 0 is dim and 250 is bright. Any value greater than 250 will default to 250.

NOTE: The brightness varies considerably with the colour of the display and so this value has been extended to cover all types of display. Care must be exercised not to overdo the brightness as this may cause excessive heating of the electronics and reduce its life.

9.2. Command 2

Name: **Individual brightness**

Format: <S-addr><2><0 to 3><0 to 250><Stop>

Sets the brightness of an individual digit on the display, 0 is dim and 250 is bright. Any value greater than 250 will default to 250.

9.3. Command 3

Name: **Clear display**

Format: <S-addr><2><Stop>

Clears or blanks the display

9.4. Command 4

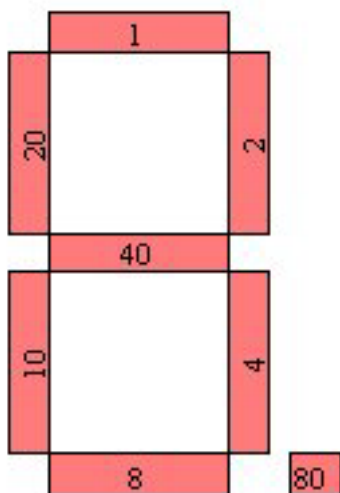
Name: **Send Byte to digit**

Format: <S-addr><4><digit><byte><Stop>

This will place the byte at the given digit.

Digit is in the range 0-3

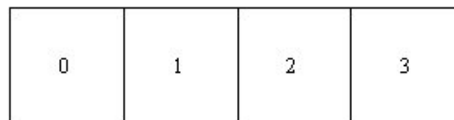
Byte is in the range 0-0xff



The byte vales are shown above and can be added together, so for example to display the number 3: 1+2+40+4+8 is added, these are HEX values and so the result is 0x4f. To display '3' onto the first digit the command

s 3 0 4f p

is used, where s is the start condition followed by the address and p is the stop condition



The digits are numbered from left to right as shown.

9.5. Command 5

Name: **Send number to digit**

Format: <S-addr><5><digit><byte><Stop>

There is a built in look up table and this command uses that in order to illuminate the correct segment to represent a number.

0-15 will display the corresponding number on the requested digit as follows:

0,1,2,3,4,5,6,7,8,9,A,b,C,d,E,F

16 will blank the digit.

9.6. Command 6

Name: **Decimal point off / off**

Format: <S-addr><6><digit><0or1><Stop>

The decimal points are treated separately from the other segments this is to enable a digit to be displayed with or without the decimal point. 1 will illuminate the DP 0 will turn it off.

9.7. Command 7

Name: **Write brightness values to EEPROM**

Format: <S-addr><7><Stop>

The brightness value for each digits is stored in RAM but at switch on is retrieved from EEPROM and so by writing the current brightness to EEPROM with this command, it can be saved even after a reset.

WARNING: Do not use this command as part of your program as the EEPROM will only accept a certain number of write cycles.

9.8. Command 8

Name: **Write current display to EEPROM**

Format: <S-addr><8><Stop>

The current display, i.e. whatever is currently on the display at the time of using this command can be written to EEPROM. This will then be displayed at start up. The default for the serial interface is shown in the serial

section so that on first use the user knows that the serial interface has been selected. The I2C default is different.

WARNING: Do not use this command as part of your program as the EEPROM will only accept a certain number of write cycles.

10. EEPROM Locations

The initial conditions and set up are controlled by EEPROM values and can be changed by the user with the appropriate commands.

System		
Location	Default Value	Description
0	0	Factory
1	98	Device address
2	6	ACK value
3	21	NACK
4	3	Baud Rate
5	1	Error reporting
6	13	Command end
14	98	Device Address
250	98	Device Address

All of these values are restored on a factory reset as some wrong choices here would make it impossible to communicate with the device.

The device address applies to both I2C and serial, hence the default serial address is 'b' and the default I2C address is 0x62. This is also stored in 3 locations and checked on start up as it is important to retain the integrity of the address.

10.1.1. Changing the device Address

The user can change the address by changing **all three locations** before reset. If only one location is changed this will be detected and set to whatever the other two are set to.

10.1.2. Baud Rate

This can be changed to another value by changing the Baud rate EEPROM location. The Baud rate code

Value	Rate
1	2400
2	4800
3	9600
4	14400
5	19200
6	38400

7	57600
8	115200

If a value is used other than those above is used then it will revert to 9600.

10.1.3. Error Reporting

This applies to Serial only, setting this value to 0 will suppress the words "Error n" but the device will still return the NACK value on error.

Error Codes

1. Not used
2. Unknown command
3. Bad device address
4. Number out of range
5. Incomplete command
6. Command locked
7. Autobaud not available

Command end only applies to serial communication and by default is set to CR (Carriage Return) or 13.

The following EEPROM locations are specific to this device.

Device		
Location	Default Value	Description
16	50	Serial brightness digit 0
17	50	Serial brightness digit 1
18	50	Serial brightness digit 2
19	50	Serial brightness digit 3
20	64	Serial value digit 0
21	64	Serial value digit 1
22	64	Serial value digit 2
23	64	Serial value digit 3
24	50	I2C brightness digit 0
25	50	I2C brightness digit 1
26	50	I2C brightness digit 2
27	50	I2C brightness digit 3
28	4	I2C value digit 0
29	4	I2C value digit 1
30	4	I2C value digit 2
31	4	I2C value digit 3

The brightness default value may vary.

A factory reset will not restore these values. There are commands available to change these values and so the user does not need to write to the EPROM directly for this.

Serial SV3 Protocol Description & System Commands

11. Revisions SV3

Rev	Change
August 2013	Version 2.3

12. Introduction to SV3

Serial Version 3 is simple serial protocol that will allow multiple devices to be used on the same serial bus similar to I2C but much easier to use.

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.

All SV3 version 3 devices have an open collector TX output that enables several devices to be tied together.

13. SV3 Electrical Interface

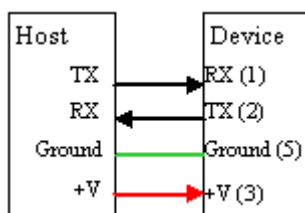
The actual physical interface may vary from the pinout shown. If it does it will be stated in the data sheet for the device.

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	Device specific
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.

14. 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 output pin will be the same as the +V pin 3.

14.1. TX Output

This is an 'open' collector output but normally there is a resistor in the device. In an ideal situation there should be only one resistor on the system. However in practice at least three devices can be connected together without any loss of performance.

If 'many' devices are needed to be connected together then the collector resistor of all of the devices except one should be removed.

14.1.1. Handshake

It is not necessary to implement a hardware handshake at the protocol expects a CR to initiate communications and the device will return ACK at the end of communication.

This considerably eases the burden on the host device and makes it very easy to debug, particularly for first time users.

15. Start Up

The device by default is 9600 Baud and will be ready as soon as power is applied. The device will only respond to its own address so this must be the first byte send.

As an example if the device address is 0x63 then if the device is connected to a terminal:

cD<CR>

will return the device ID as 'c' has a byte value of 0c63. It is recommended that a terminal be used to become familiar with the device.

16. Command Format

All devices have an address which is one byte and can be any value within the range 32 to 254 however by convention the range 97 to 122 is used. The user can alter the default 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 (one) 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 (upper case letters) and those that are device specific which are described in the devices datasheet.

NOTE: By convention the commands common to all devices are in the range 65 to 90. This corresponds to the ASCII codes for upper case letters. The range used for the individual device commands are in the range 97 to 122, this corresponds to lower case letters. Using this convention makes it easy to debug a device using a text terminal.

This section deals with the system (upper case) commands.

The command format in general is therefore:

`<address>{commands}<EOL>`

Serial SV3 Protocol Description & System Commands

The device will always (reset command is an exception) respond with ACK which by default is 6. EOL is the end of line character which by default is 13 ("r")

17. 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>
```

Terminal Example cR12,120

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
```

Terminal Example cP12,120

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.

18. 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. 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 it is important that the host waits for the ACK.

19. 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.

20. 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.

20.1. Command 3

This will reset all devices as if they had just been powered up.

21. 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 EEPROM for example would simply be:

```
aR0,6<CR>
```

The device will then output readable text.

21.1. Summary

Command	Description
'W'(87)	Write to EEPROM
'R'(82)	Read EEPROM
'I'(73)	Toggle inverted
'C'(67)	Reset device
'V'(86)	Version
'D'(68)	Device ID
'H'(72)	Hello

Note that examples will use the default address of 97 (a).

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	Depends on device	Device address *
2	6	ACK character
3	21	NACK character
4	3	Baud rate [1]
5	1	Error reporting 0 is off
6	13	Default CR or end of line character

Serial SV3 Protocol Description & System Commands

Bytes up 15 reserved for device or future use.
*Some devices store the device address in more than one location for security. When changing the device address all of the locations holding the address have to be changed.

The Baud rate has the following values:

1. Baud rate is fixed at 2400
2. Baud rate is fixed at 4800
3. Baud rate is fixed at 9600 (default)
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

A value other than the above will revert to 9600 Baud.

21.2. 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 this part of the EEPROM back to its default settings.

As an example to change the Baud rate from 9600 to 115200 the following bytes will be sent by the host:

```
97 87 52 44 56 13
```

```
aW4,8<CR>
```

A comma (44) is used as a delimiter to separate the EEPROM address from the value. EEPROM address 4 contains the Baud rate location.

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

21.2.1. Address

This EEPROM location 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.

Some devices store the address in more than one place. This is for address security as in poor electrical environments it is possible for an EEPROM location to randomly change. If this is the case the locations will be stated in the data sheet and all of the locations need to be changed before a reset. Values 0 and 0xff will be considered invalid.

At start up the system will check all address locations for agreement and reset any deviant address.

21.2.2. 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.

21.2.3. 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.

21.2.4. 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.

21.2.5. 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.

21.3. 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>
```

```
aR0,16<CR>
```

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 EEPROM would be:

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

21.4. Device ID

Returns a number representing the device product number as a string

```
97 68 13<host>
```

```
aD<CR>
```

```
"4111"<ACK> Returned by device
```

21.5. 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 to any unusual equipment. **Warning.** in inverted mode only one device will be able to use the TX line as it will negate the effect of having an open collector output.

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.

Serial SV3 Protocol Description & System Commands

97 73 13 – output is now inverted

aI<CR>

97 73 13 – output is now non-inverted

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

aI<CR>

NOTE: inverted output cannot be used with more than one device connected to the bus.

21.6. 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.

aC<CR>

21.7. Version

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

An example of the transaction would be:

97 86 13 <host>

aV<CR>

49 16 19 <ACK>

21.8. Hello

This command is used to check what devices are on the bus. It simply returns ACK but where there is more than one device on the bus the following pseudo code will list them:

```
for j = 97 to 122
  Send(chr$(j)+"H\r")
  if ack received then
    print device j found
```

If a device is found then the other attributes such as device ID can be obtained. Also user information could be stored in the devices EEPROM and retrieved.

22. 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
7	Auto Baud rate not available on this device

23. Connection and Configuration

23.1. 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 most SV3 devices there is a built in pull up resistor this may be connected to a jumper that has is shorted out by a PCB track (check with the device data sheet). In practice around three devices can be connected to the same bus without regard to this jumper or pull up. Later SV3 devices use a 10k pull up resistor and so more than 3 devices may be used. If there is a problem then the resistors can be removed and just one left in place.

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 bus down 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.

24. Restoring Factory Defaults

The configuration of an SV3 device is contained in the first few bytes of 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 system part of 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.

Serial SV3 Protocol Description & System Commands

I2C System Commands

Rev	System Section Changes
Oct 2008	Preliminary
Dec 2008	Removed command 96
Aug 2013	Updated

25. I2C System Commands

The following section deals with system commands that are common to all I2C devices. Note that not all of these commands are available for all devices.

Command	System Command Set
0x90	Read EEPROM
0x91	Write EEPROM
0x95	Reset
0x96	Send ACK
0xA0	Firmware Version
0xA1	Returns device ID

Most but not all devices contain an EEPROM that can store data when the power is off. The first 16 bytes of the memory is reserved for system use and should not be changed by using these commands.

If the contents of the first 16 bytes are changed then, depending on the device unpredictable results may occur. A factory reset will put the contents back to normal. In some devices not all 16 bytes are used.

The rest of the EEPROM can be used by the user for any purpose.

25.1. Command 0x90

Name: **Read EEPROM**

Format: <S-addr><0x90><EE-Address><R-Addr><data...><Stop>

BV4221 Example

```
0x42>s 90 0 r g-3 p
```

The above will fetch 3 bytes from the EEPROM addresses 0, 1 and 2

This command will allow a single or several bytes to be read from a specified EEPROM address.

25.2. Command 0x91

Name: **Write EEPROM**

Format: <S-addr><0x91><EE-Address><data...><Stop>

BV4221 Example

```
0x42>s 91 10 1 2 3 p
```

The above write 1,2 and 3 to EEPROM addresses 0x10, 0x11 & 0x12

This command will write one or more, up to a maximum of 30 bytes at any one time, to be written to the EEPROM.

25.3. Command 0x95

Name: **Reset**

Format: <S-addr><0x95><Stop>

BV4221 Example

```
0x42>s 95 p
```

Resets the device, this is a soft reset and may not fully reset the device as recycling the power would.

25.4. Command 0x96

Name: **Send ACK**

Format: <S-addr><0x96><Stop>

BV4221 Example

```
0x42>s 96 r g-1 p
```

Request the device to send ACK and read it.

Simply requests the device to send an ACK. This can be useful to see if the device is still alive or some long commands may require this.

25.5. Command 0xA0

Name: **Firmware version**

Format: <S-addr><a0><R-Addr><byte><byte><Stop>

BV4221 Example

```
0x42>s a0 r g-2 p
```

This will return the two firmware bytes.

This simply returns two bytes that represents the firmware version.

25.6. Command 0xA1

Name: **Device ID**

Format: <S-addr><a0><R-Addr><byte><byte><Stop>

BV4221 Example

```
0x42>s a1 r g-2 p
```

This returns two bytes that represent the device ID. This is a later addition to the command sent and so may not be available on all devices.

26. Hardware Reset

A hardware reset has been provided should the device address be changed to some unknown value.

The method of restoring the factory defaults and thus the default device address is as follows:

- 1) Remove power

I2C System Commands

- 2) Hold the designated pin low or high or connect two pins together. The actual pins are device dependant and will be referenced in the sections above this text.
- 3) Apply power
- 4) Remove power
- 5) Remove shorting link

When power is now restored the device will have the default I2C address, normally 0x42.

27. Commands

To further explain the format of the commands this section has been provided. The design of the interface has been purposely kept simple and so there are only a few standard sequences required.

Key

Master

Slave

S Start condition

P Stop Condition

A Acknowledge = 1

N Not acknowledge = 0

27.1. Sending a single command

This is designated in the list as:

<S-addr><cmd><Stop>

This sequence is used for simple functions where no data is involved. The I2C sequence, using the default address is:

S 0 1 0 0 0 0 1 0 A Command A P

27.2. Sending a command with a parameter byte/s

This is designated in the list as:

<S-addr><cmd><data...><Stop>

Some commands expect a parameter after the command. In this case the bytes are sent one after the other up to the maximum of 31 bytes. The stop command tells the slave that there is a command ready to be executed.

S 0 1 0 0 0 0 1 0 A Command A →

Byte A Byte... A P

27.3. Receiving bytes from the slave

<S-addr><cmd-Addr><byte><Stop>

When receiving one or a number of bytes from the device a restart is required.

The command is sent with an even address (the R/W bit 0). When the slave acknowledges the command another start condition is sent with the R/W bit set to 1. This is called a restart. After this restart the slave will continue to send bytes until the master sends a not acknowledge (N or NACK).

If the master does not send a NACK at the last byte the slave will be expecting another byte to be requested and this may cause either unpredictable results or the I2C bus to lock.

28. System EEPROM

This may vary from device to device but in general will be as follows.

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	Depends on device	Device address *
2	6	ACK character
3	21	NACK character
4	3	Baud rate [1]
5	1	Error reporting 0 is off
6	13	Default CR or end of line character
7		
8		
9		
10		

Bytes up 10 reserved for device or future use. *Some devices store the device address in more than one location for security. When changing the device address all of the locations holding the address have to be changed.

The Baud rate has the following values:

9. Baud rate is fixed at 2400
10. Baud rate is fixed at 4800
11. Baud rate is fixed at 9600 (default)
12. Baud rate is fixed at 14400
13. Baud rate is fixed at 19200
14. Baud rate is fixed at 38400
15. Baud rate is fixed at 57600
16. Baud rate is fixed at 115200

A value other than the above will revert to 9600 Baud.

I2C System Commands

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

28.1.1. Address

This EEPROM location 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.

Some devices store the address in more than one place. This is for address security as in poor electrical environments it is possible for an EEPROM location to randomly change. If this is the case the locations will be stated in the data sheet and all of the locations need to be changed before a reset. Values 0 and 0xff will be considered invalid.

At start up the system will check all address locations for agreement and reset any deviant address.

28.1.2. 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.

28.1.3. 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.

28.1.4. Turn off Error reporting

** Applies to serial devices only **

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.

28.1.5. CR Character

** Applies to serial devices only **

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.

29. Trouble Shooting

This section has been added to answer frequently asked questions. The problems are usually caused because the master device has not had the I2C specification fully implemented.

29.1. Pulse Stretching

This is a method of I2C handshaking which is used in BV slave devices but it is not always supported by the master system. The symptoms are erratic behaviour, some commands will be accepted and others will not.

To explain: when the slave device is busy it holds the clock line low (normally only the master controls the clock line), the master should check that the clock line is high before sending the start condition. If it is low the master should wait until it is free.

Quite a few slave devices do not use pulse stretching and so this not being implemented in the master does not show up. However when dealing with relatively slow hardware, an LCD display for example (i.e. BV4219), this will become a problem. The work round is to make sure that the master recognises pulse stretching properly or introduce delays after each command.

29.2. Last Read NACK

When optionally multiple reads of a slave is required (the 0x55 command is a good example) the last read should send a NACK rather than a ACK. This informs the slave that no more reads from that command are required.

It has been found that some master implementations do not send a NACK on the last read. This causes the BV slave to remain in the (multi read) command effectively blocking any other commands.

The typical symptoms are that when the 0x55 command is implemented no other commands will work after that.

29.3. Pull Up's

The most common problem when trying to get a new device going is to forget to put the pull up resistors somewhere on the bus. BV Slave devices do not have pull up resistor on board so they must be provided by the master (the BV4221 has pull up's) or provided externally. A value of around 5k is okay but this is not usually critical.