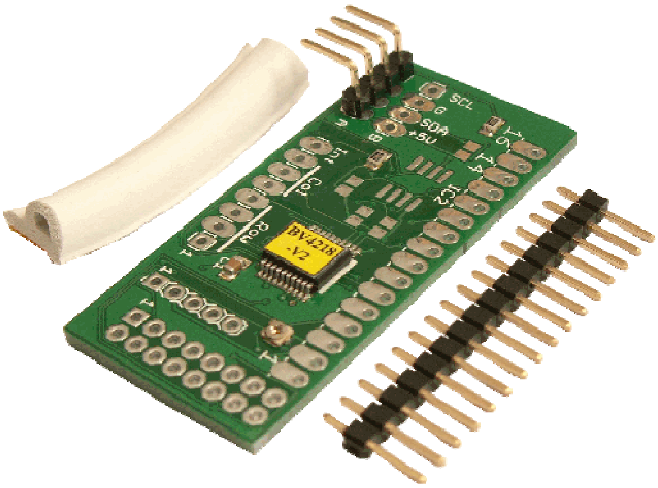


BV4218-V2

I2C-LCD & Keypad

Product specification

November 2010

Contents

Rev	Change
November 2010	Preliminary

1. Introduction

The BV4218-V2 is a small circuit board with an I2C serial interface and command set designed to fit at the back of an LCD display.

In addition to controlling the display it can also drive a standard 12 key keypad. A ready made input/output system is realised for small systems using just one I2C interface.

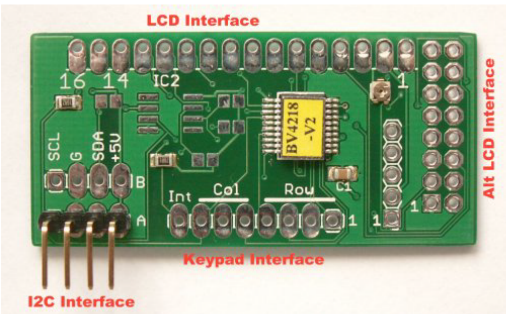
Version 1

This is FULLY backwards compatible with the previous version of the device. Version 2 has the optional addition of an I2C line driver. The firmware is exactly the same.

2. Features

- I2C up to 400kHz
- User configurable address
- Simple command set for direct interface to LCD module
- Back light output
- Contrast trimmer
- 1,2 and 4 line displays
- 12 way crosspoint keypad interface
- 16 key keypad buffer
- Extra general purpose pin or interrupt when key available
- Operating voltage 2.0 to 5.5V
- Current <1mA @ 5V

3. Electrical Specification



The board is designed for both buffered and unbuffered operation. Most boards supplied will be unbuffered, this is the normal I2C interface and will require pull up resistors. These resistors are usually on the master device.

The I2c Interface is as soun on the lower set of pins.

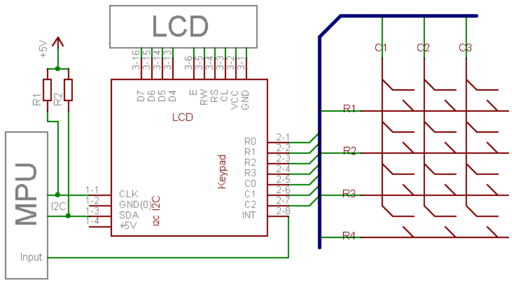


Figure 1 Wiring Diagram

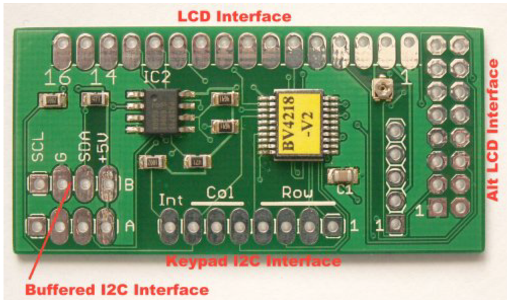
The device, when fully connected can be used for an LCD and a 12 way keypad with the typical wiring arrangement as shown.

It is not necessary to have both the LCD and the Keypad as the device will handle one or the other or both.

The default for the device is set for 2 display lines with a 5 x 7 dot matrix character set. This is determined by the commands issued to the display at start up, this of course can be changed when the I2C communication is established.

The sign on message is 11 characters so should fit on a small display. This can be changed using a system command.

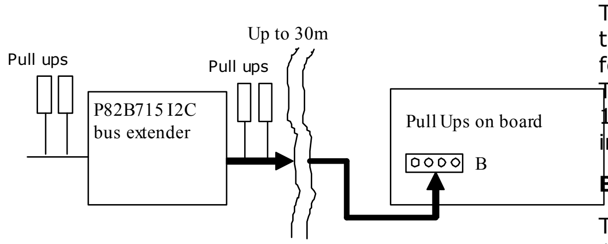
3.1. I2C Bus Extender Interface B



This version has the option of providing an I2C bus extender (NXP Part 82B715) that can enable the device to be operated at up to 30 metres away. This option can easily be seen by the presence or absence of the bus extender IC.

There are TWO sets of I2C pins one marked A and the other marked B. A is for normal use and B is for use with the buffer. So for parts without the buffered option fitted A should be used.

This option is used with another extender at the other end, no pull up resistors are required at the device end as they are already on board.



When using this option 2 x 470R resistors are on board connected to the SDA and SCL lines. These will need to be matched at the source end.

The angled pinhead is supplied but not soldered in so that the user can choose to use the unbuffered or buffered I2C. Either way with this board the pull up resistors are in place.

3.2. I2C Interface A

This is for normal use and does not go through the buffer (if fitted). If a buffer is fitted then pins B should be used.

There are FOUR sets of pins, one for the I2C connector, TWO for the LCD connector and one for the keypad. The pin out descriptions are given in the following tables.

Pin	Description I2C Pins
1	SCK
2	GND
3	SDA
4	+5V

Table 1 I2C Pin Description

The I2C interface is a standard 2 wire interface that must have somewhere along the bus two pull up resistors. A suitable value would be 5k6.

Pin	Description LCD Pins
1	GND
2	VCC
3	CL
4	RS
5	R/W (Gnd)
6	E
7	n/c
8	n/c
9	n/c
10	n/c
11	D4
12	D5
13	D6
14	D7
15	Back Light
16	GND

Table 2 LCD Pin Description

The LCD pin-out is repeated along the top of the circuit board and down the side to cater for top and side connected LCD displays. There are no physical connections to pins 7 to 10 as the device uses a 4 way command interface.

Back Light

The back light is normally presented on pins 15 and 16 with the cathode of the LED to pin 16 but this is not always the case. The BV4218 will supply and sink about 20mA on pin 15. There is no current limiting resistor so this may be needed depending on the display type, some have them built in.

Contrast

The contrast is provided by the small trimmer, fully ant-clockwise is full contrast. This is simply a potentiometer with a tap onto pin 3.

Pin	Description Keypad Pins
1	Row 0
2	Row 1
3	Row 2
4	Row 3
5	Column 0
6	Column 1
7	Column 2
8	Interrupt, active low, (factory)

Table 3 Keypad Pin Description

A standard cross point keypad can be directly connected to these pins. This will provide a keypad interface that can be interrogated using the I2C interface. The main advantage of this is that the MPU does not need to poll the keypad matrix thus leaving valuable time to do other things.

There is an internal 16 key buffer and an optional translation table that will translate the scan codes into user selectable values.

Interrupt

The interrupt output goes low whenever there are keys in the input buffer ready to be read. The MPU can either test this line or send an I2C command to query the key buffer.

There are also commands to set and clear this independently of the key buffer function.

Factory Reset

Pin 8 (table 3) is the factory reset pin and this has a resistor to hold it high at switch on. By holding this pin low at power on a factory reset will take place, see section 9.

4. 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.

There are three type of command, those referring to the LCD, those referring to the keypad and those referring to the system. The system commands enable changing of the device address etc.

Device default address is 0x42

All I2C commands have one of two formats. For writing to the device:

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

and for reading from the device:

<S-Addr><Command><RS-add><data..[NACK]><Stop>

S-Addr is a start condition followed by the device address. The address here will always be an even number, i.e. with bit 0 clear (0).

All command begin by writing to the device and the first thing that is written is the command number. What follows after this is dependant on the command.

Where the command requires information from the device (read) it is necessary to send a start condition again followed by the address + 1, this is shown by RS-Addr above.

Some read commands can be terminated by sending not-acknowledge (NACK) before the supply of data is exhausted.

All command strings whether read or write terminate with a stop condition.

Command	LCD Command Set
1	Send LCD Command
2	Send LCD Data
3	Back Light
4	Print Message
5	Display on
6	Message Read
Command	**Keypad**
0x10	Key Query
0x11	Get Key
0x12	Key map start
0x13	Key down
0x14	Clear key buffer
0x15	Get Key Scan Code
0x16	Clear Interrupt pin
0x17	Set Interrupt pin
0x18	Dump Key Buffer
Command	**System Command Set**
0x55	Test
0x90	Read EEPROM
0x91	Write EEPROM
0x92	Confirm Command Complete
0x93	End of EEPROM
0x98	Change Device Address Temporary
0x99	Change Device Address Permanent
0xa0	Return version

Table 4 LCD & System Command Set

Table 4 is a command summary of all of the LCD, keypad and system commands.

5. The LCD Command Set

The LCD command set is a direct control of the standard internal controller chip, typically a HD44780, SPLC780D that allows simple control over the LCD display.

There are just two basic inputs to the LCD display and that is one for controlling how the display works and the other for writing characters to the display. This is taken care of by commands 1 and 2 as described later.

A useful source of information on how to control the lCD display can be found at http://www.doc.ic.ac.uk/~ih/doc/lcd/ . This is one of many on the net.

The method of writing to the display 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 (0x42). 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..[NACK]><Stop>

The restart address will be one greater then the start address, thus if the start address is 0x42, the restart address will be 0x43. Again the data can be one or more bytes read from the device.

Each command will have it's own format and is described in the following text. A start condition and address is always followed by a command. The device will use clock stretching until it has finished with the command. Not all master devices can detect this. If this is the case delays in the host software will be required.

5.1. Command 1

Name: **Send LCD Command**

Format: <**S-addr**><**1**><**byte**>**<Stop>**

This command will send data directly to the LCD as a command, i.e. with the RS line low. Using this enables the display behaviour to be altered to suit the application.

Examples for **<byte>**:

1 = Home and clear display

4 = This will move the cursor from right to left when a character is displayed.

6 = Default move cursor to right

0x0c = Turns cursor off

0x0e = Turns cursor on

0x0F = Cursor blinks whole character

All of the above are determined by the display controller, command 1 simply sends a byte of information to the display with the RS line low.

5.2. Command 2

Name: **Send LCD Data**

Format: <**S-addr**><**2**><**byte...**>**<Stop>**

This will send data to be displayed on the display. The data should be ASCII coded text .

Examples for <**byte..**>

0x46 0x72 0x65 0x64

Writes **Fred** to the display.

5.3. Command 3

Name: **Back light**

Format: <**S-addr**><**3**><**1 or 0**>**<Stop>**

This command controls the back light pin as a digital output. It is the inverse of the number sent so a 1 would place a logic 0 on the pin and a 0 would place a logic 1 on the pin.

As this is connected to pin 15 which in turn is normally connected to the anode of the LED, sending a 0 would normally illuminate the back light.

5.4. Command 4

Name: **Print Message**

Format: <**S-addr**><**4**><**EEPROM Address**>**<Stop>**

This will print a string contained in the EEPROM to the display at the address given at EEPROM Address. The string can consist of commands and characters and follows this format:

0x0 Terminates the string

0x01 Next byte that follows is an LCD command

Anything else will be sent to the display as LCD data. See command 6 for creating a new sign-on message.

Example:

0x01 0x01 0x46 0x72 0x65 0x64 0x00

The first byte 0x01 tells the controller to interpret the next byte as a command, the command in byte 2 0x01 will be sent to the display as a command and thus it will clear the display and home the cursor. The sting "0x46 0x72 0x65 0x64" is ASCII for Fred and this is terminated with 0x00.

5.5. Command 5

Name: **Display on**

Format: <**S-addr**><5><0 or 1>**<Stop>**

This directly controls the contrast output and is the inverse. A one in this command will take the contrast pin low, thus enabling the display. A 0 on this command will take the contrast pin high and thus blank the display.

5.6. Command 6

Name: **Message Area**

Format: <**S-addr**><**6**><**R-Addr**><**data**>**<Stop>**

The value returned from this command is the EEPROM address of the sign on message. A new message can be placed here to customise the start up message.

This address up to the end of EEPROM can be used for any purpose including other custom messages. The end of the EEPROM can be determined by system command 0x93

5.6.1. Changing the sign on message

The sign on message is stored in EEPROM so that it can be easily changed by the user. The following gives an example of changing the message using a BV4221 device and the default address of this device as 0x42.

Step 1 discover where the address of the message starts:

BV4221 Example

0x42>s 6 r g-1 p

(returned value is 28)

Step 2 use this address for writing to EEPROM. The message MUST end in 0.

BV4221 Example

0x42>s 91 28 48 65 6c 6c 6f 0 p

Restart the device and the new message will be "Hello"

6. Keypad Commands

Command	Keypad

0x10	Key Query
0x11	Get Key
0x12	Key map start
0x13	Key down
0x14	Clear key buffer
0x15	Get Key Scan Code
0x16	Clear Interrupt pin
0x17	Set Interrupt pin
0x18	Dump Key Buffer

Keypad commands are mainly of the read type, that is they return one or more values.

6.1. Command 0x10

Name: **Key Query**

Format: **<S-addd><x010><RS-add><dada><P>**

This command will return one byte that represents the number of keys in the keypad buffer. Zero will be returned if there are no keys in the buffer. Each time a key is pressed this value will be incremented by one. When a key is read using the Get Key command this number will be decremented by one.

In practice this command can be used to poll the keypad to check if a key has been pressed. There is also an interrupt line that goes low if there are any keys in the key buffer.

NOTE the key buffer is 16 bytes

6.2. Command 0x11

Name: **Get Key**

Format: **<S-addd><x011><RS-add><data...NACK><P>**

This command will get a key from the key buffer and reduce the buffer count by one. The value returned is not the scan code but the value in the EPROM that is addressed by the scan code. One or more bytes can be read with this command, a NACK should be sent after reading the last byte and prior to sending the stop condition. ** See also Get Key Scan Code

6.3. Command 0x12

Name: **Key Map Start**

Format: **<S-add><0x12><RS-add><data><P>**

This command will return the address of the keymap start address in EEPROM. When a key is pressed a scan code is created and stored in the key buffer

Address	**EEPROM MAP**							
0x0	0x 42				0x 28	0x 0e	0x 06	0x01
KEYMAP	0	1	2	3	4	5	6	7
	8	9	10	12				DEF
MSG								
								END

Table 5 System EPROM Area

Key Mapping

KM* 0 1 2 3 4
5 6 7 8 9
10 11

The above values are stored in the EEPROM after a factory reset, the values can be used to determine the address of the EPROM that the particular key returns. This is because the values are the same as the address using KM as an offset.

If for example key 1 is pressed on the keypad and the resulting value of 5 is returned when using the get Key command the value of 1 needs storing at location 5 so a subsequent press of 1 will return the contents at 5 which will be a 1.

KM is determined by command 12. This value is used when writing to the EEPROM using command 0x91. So in the above example when 5 was returned after pressing key 1 the address to write a value of 1 to is **KM** + 5.

KM + 16 is the default value, this is returned when there are no keys in the buffer to return. This value can be used instead of using the key query commend. The factory setting of this is 0xFF.

6.4. Command 0x13

Name: **Key Down**

Format: **<S-add><0x13><RS-add><data><P>**

This command returns one byte and will indicate if a key is actually being pressed at the time the command was issued. A value of 1 indicated that a key is being held down, 0 indicates otherwise.

Using this command the microcontroller can implement a repeat key scheme, a volume control for example or a quick type system as implemented on a PC.

6.5. Command 0x14

Name: **Clear Key Buffer**

Format: <**S-addr**><**0x14**>**<Stop>**

Issuing this command will flush the buffer and set the key count to 0.

6.6. Command 0x15

Name: **Get Key Scan Code**

Format: **<S-addd><x015><RS-add><data...NACK><P>**

This works in the same way as Get Key except this will return the key scan code in its 'raw' format. The code will not be translated via the EEPROM look up table.

If no keys are in the buffer, the output from this will be meaningless.

6.7. Command 0x16

Name: **Clear Interrupt Pin**

Format: **<S-addr><0x16><P>**

This will set the interrupt pin to 0.

6.8. Command 0x17

Name: **Set Interrupt Pin**

Format: **<S-addr><0x17><P>**

This will set the interrupt pin to 1.

6.9. Command 0x18

Name: **Dump Key Buffer**

Format: **<S-addd><x018><RS-add><data...NACK><P>**

This will display the contents of the key buffer regardless of whether there are any keys in there or not. Data can be up to 16 bytes.

7. System Commands

Command	System Command Set
0x55	Test
0x90	Read EEPROM
0x91	Write EEPROM
0x92	Confirm Command Complete
0x93	End of EEPROM
0x98	Change Device Address Temporary
0x99	Change Device Address Permanent

The system commands enable the user to change the way the device works and this mainly consists of writing to the built in EEPROM. The EEPROM map with the default values is shown in Table 5.

7.1. Command 0x55

Name: **Test**

Format: < **S-addr**><**0x55**><**start**><**R-Addr**><**Value**><**NACK**>**<Stop>**

This command simply returns an incrementing value until NACK is sent by the master prior to stop. This can be useful for testing that the interface is working.

7.2. Command 0x90

Name: **Read EEPROM**

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

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

7.3. Command 0x91

Name: **Write EEPROM**

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

This command will write one or more, up to a maximum of 30 bytes at any one time, to be written to the EEPROM. Address 0 of the EEPROM is the device address and this cannot be written to by this command. A special command 0x99 is used for this purpose.

7.4. Command 0x92

Name: **Confirm Write to EEPROM**

Format: <**S-addr**><**0x92**><**R-Addr**><**data**>**<Stop>**

EEPROM write operations are not instantaneous. This command will return a value of 1 if the EEPROM is still in the progress of writing otherwise it will return 0.

This command should be issued before command 0x91 is used.

7.5. Command 0x93

Name: **End of EEPROM**

Format: <**S-addr**><**0x93**><**R-Addr**><**data><Stop>**

The system only uses a small portion of the first part of the EEPROM, the rest of the EEPROM can be used for user data or other purposes depending on the device. This command returns a single byte that will determine the last writeable address of EEPROM.

7.6. Command 0x98

Name: **Change Device Address Temporary**

Format: <**S-addr**><**0x98**><**New-Addr**>**<Stop>**

This will change the device address with immediate effect and so the next command will use the new address. The address must be a write address (even number) Odd numbers will simply be ignored. The effect will last as long as the device is switched on. Resetting the device will restore the address to its original value. The address is stored in EEPROM location 0.

7.7. Command 0x99

Name: **Change Device Address Permenant**

Format: <**S-addr**><**0x98**><**New-Addr**>**<0x55><0xaa><Current-Addr><Stop>**

This command changes the address immediately (the next command will need to use the new address) and permanently (see hardware factory reset). The address must be a write address (even number) and follow the sequence exactly.

Permanent in this case means that the device will retain this address after power down, i.e. it is stored in EEPROM. Should anything go wrong the default address can be restored by using a factory reset.

7.8. Command 0xa0

Name: **Read Version**

Format: <**S-addr**><**0xa0**><**EE-Address**><**R-Addr**><**big,small**>**<Stop>**

This command returns the current firmware version as two bytes, the first byte is a number from 0 to 255 and the second byte is an ASCII value representing the minor version identifier.

As an example if the version was 3.j the two bytes returned would be 3 and 106.

8. EEPROM

The EEPROM is normally 256 bytes and most of it can be used for messaging or other purposes. The size of the EEPROM may vary from device to device so this should be checked by the system command 0x93. There is a small area at the beginning of the EEPROM reserved for system use and is designated according to the following table:

Address	Default	Function
0	0x42	Device address
4	0x28	LCD Function Mode [1]
5	0x0e	Cursor direction [1]
6	0x06	Entry Mode [1]
7	0x01	Home cursor [1]

[1] During the initialisation of the display these four commands are sent to the display to set it up in the default mode. The bytes can be changed so that the display initialises in a different way if required.

9. Hardware Reset

A hardware reset sequence has been provided should the device address be changed to some unknown value or the system area is written to and this stops the device working.

There is a hardware reset pin that is normally high on start up, the relevant device will specify the pin. The sequence is as follows.

1. Power down the device
2. Connect the hw-reset pin to ground For the bv4218 this is the INT, pin 8 on the keypad connector)
3. Power up the device of a minimum of 100 ms
4. Power down the device and remove the connection to ground.

The function of the hardware reset will re-initialise all of the EEPROM values from address 0 to about address 0x40, setting the values to the factory defaults.