

BV4108 User Guide

Rev	Change
Jan 2011	Preliminary
Jan 2011	Updated BSerial

Resources (zip file) can be found on www.byvac.com/bv3 under the description of this device and also at: http://www.asi.byvac.com/da_data.php

Introduction

The BV4180 is an LCD controller that has an ASI interface. This means that it can connect directly to a COM port if required and more than one device can share the same bus this is because each device has its own unique address. The address is settable by the user and stored on internal EEPROM.

A full description of the protocol and connection information can be found here:

http://www.asi.byvac.com/pr_asi.php

This guide mainly deals with the software aspects. There are two libraries provided one for the PIC32-Basic and the other for the Arduino.

Software

Common to all devices is the ASI commands section, two 'libraries' are available one for the Arduino and one for the PIC32-Basic.

PIC32-Basic (version 1.47)

The files here are not quite complete and are included for reference mainly. It should easily be possible to 'hack' the code for your own purposes.

```
asi.bas      // main utilities
iasi_lcd.bas // sample test routine for display
```

Arduino BV4108 Library

NOTE: see http://www.asi.byvac.com/da_data.php for the ASI library guide as this class inherits all of the methods from those classes. Details of these methods can be found in the ASI guide.

Methods inherited from BSerial:

```
baud(rate)
handshake(uint8_t rtsPin, uint8_t ctsPin)
flush()
putch(char c)
unsigned char puts(char *s)
unsigned char buffer()
char getch()
```

Methods inherited from ASI:

```
begin();
resetall(char *dev, char max)
```

```
devices(char *dev, char max)
char wait(char *inp, char term, char max);
char wait();
changeaddress(char DEVadr, char newadr)
eepromwrite(char DEVadr, char addr, char *s)
ackoff(char DEVadr)
erroroff(char DEVadr)
eepromread(char DEVadr, char addr, char bytes, char *read, char max);
deviceid(char DEVadr, char *id)
macro(char DEVadr, char mon)
reset(char DEVadr)
version(char *s)
```

BV4108(rxPin, txPin)

This is the constructor and mainly passes the information over to the serial library.

NOTE to initialise the ASI device begin() must be used.

putdata(char DEVadr, char data)

Puts a byte to the LCD display as data, there is no ACK returned for this command and so some delay will be required after use. This will probably be very small sub 1mS and also there is an input buffer that will take a few characters. For full information on how to use this see the data sheet for the HD44780 (or similar) controller.

It is unlikely that this command will be needed but as an example:

```
putdata('a', 65);
```

will display 'A' on the LCD if the BV4108 has a device address of 'a'

putcmd(char DEVadr, cmd)

Sends a byte to the display as a command. For full information on how to use this see the data sheet for the HD44780 (or similar) controller.

cls(char DEVadr)

Clear the display, sends command 1. e.g. putcmd('a',1);

text(char DEVadr, char *s)

Send a string to the display.

rowcol(char DEVadr, char row, char col)

Place the cursor on a specific line and column on the display. This is based on:

```
#define CMD_LINE1 0x80
#define CMD_LINE2 0xc0
#define CMD_LINE3 0x94
#define CMD_LINE4 0xd4
```

Which works for most displays up to 4 lines. These are defined in the header file so can be changes to suit.

bl(char DEVadr, char blon)

Turn the back light on or off depending on what is supplied for 'blon' and also the display. 1 will either switch the BL on or off, 0 will do the opposite.

custom(char DEVadr, char chno, char *buf8)

This is for defining a custom character, place 8 bytes in *buf8 and choose where to store the character 0 to 7 in chno. Up to 8 characters can be defined. for a description of how to define them see the HD44780 data sheet. There are also a few design tool on the net.

Example pde

The supplied example in the resources zip file will produce an output as shown in the picture when using a 4 line display. It runs once in loop() and is stopped from repeating by while(1); at the end of the loop.



NOTES and FAQ

Using two or more ASI devices

It is possible and quite likely that two or more ASI devies will be running from the same Arduino. One way to do this is to include both libraries, for example

```
BV4108 disp(rxPin, txPin);  
BV4501 key(rxPin,txPin);
```

This works but there may be resource issues as both classes share and inherit the same base classes namely BSerial and ASI. I do not know if this will take up double resources or the compiler will optimise the extra instances out. If I find out (dug@bayvac.com) I will update this note.

If this does take up resources then simply put the required code in the main sketch, most of the work is carried out by the ASI class anyway.