BV4626 User Guide

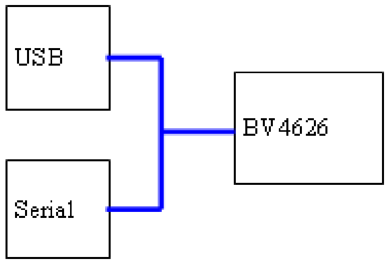
Revisions 3 Sept. 2010: Digital input impedance was wrongly stated at 10k.
Oct Minor corrections + USB information

The BV4626 is a general purpose input / output board with a Serial and I2C interface. This guide supplements the Datasheet and is intended to enable the user to become quickly familiar with the board. Various resources are available and are used in this guide these can be found at www.byvac.com under the BV4626 product.

** Please see the above website for the latest datasheet

Getting Started

There are three options available for connecting this device to other equipment and for simplicity of description the USB option will be used. The USB simply feeds into the serial interface and so anything that is relevant to this is relevant to the serial interface.



Connection to a microcontroller, serial or I2C is shown in the datasheet. When first connecting the USB, the FTDI drivers will be needed. The FDDI is so popular that there is a good chance you already have this installed and in fact the driver comes pre-installed in Linux. If not the driver can be obtained form here:

http://www.ftdichip.com/Drivers/VCP.htm

It is the Virtual Com Port (VCP) driver that is required. If you do not know how to install a driver the full instructions come with the driver zip file.

Terminal

The easiest way to understand how the BV4626 works is to use a simple terminal program, HyperTerminal will do but BV-COM is better and its free. It can be downloaded from the resource zip file at www.byvac.com under the BV4626 product or from here http://www.asi.byvac.com

BV-COM does not need to be installed, just place in a subdirectory along with its 'ini' file and double click on the 'exe'.

A useful feature of BV-COM is that it has a reset icon, which when pressed will reset the device.

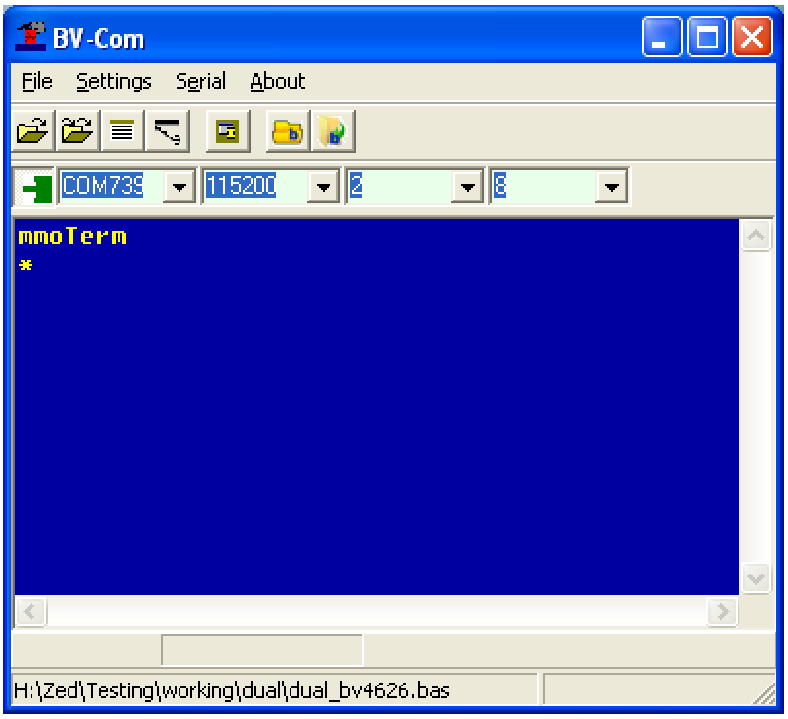
1. Connect the device to the USB
2. Select the correct COM port and a suitable Baud rate (as shown in FIG 1).
3. Connect to the port by using the red icon which should then turn green
4. Press <enter>. All being well a '*' will appear on the screen. If not press reset and then <enter> again.

Start up

It's worth mentioning here how the 'dual' devices start up by default:

1. At switch on, the device will wait for a carriage return (CR) byte, this is 0x0d or 13 and is sent when the user presses <enter> at the keyboard.

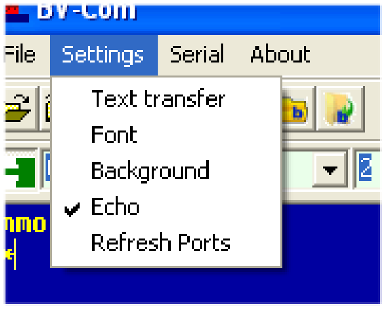
2. When the device receives CR it uses this to calculate the Baud rate of the host device. The Baud rate is selected from a set of allowable Baud rates. If something other then CR is sent then the device at this stage has know way of knowing that it is not CR and so may choose an incorrect Baud rate. It is important therefore that only CR is sent at this stage.
3. Once a Baud rate has been selected it will send '*'. This can be used by the host to verify communication has been established.

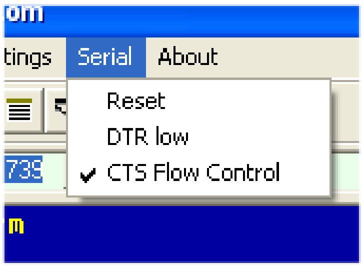


```
mmoTerm
*
```

FIG 1

To make things easier, make sure echo is on and CTS flow control.





Echo: will enable you to see what you are typing
CTS Flow Control will stop the host sending when the BV4626 is busy, but this will happen only when the BV4626 buffer becomes full which will only happen when sending a long file to the device.

The device is essentially a serial device which makes communication with the host device very easy and it is based on escape codes. Pressing the escape key will send 0x1b (27). This indicates to the device that a command sequence is about to happen, try:

<esc>[1A<esc>[1B<esc>[0A<esc>[0B

This <esc> is the escape key, you should here the relays click on and off, if not you have misunderstood the above sequence of characters, try again.

A fundamental principle is used on the Dual interface which may have been observed when typing in the above sequence and that is the command is actioned immediately following the last letter of the command sequence. This when turning relay A on with:

<esc>[1A

The relay was switched on as soon as 'A' was received. None command characters are simply ignored and so the above could have been entered as follows:

<esc>[1A

<esc>[1B

<esc>[0A

<esc>[0B

The additional <enter> keys make not difference.

NOTE: On PCB Version BV4626-c relay A is marked as RLY2 and relay B is marked as RLY1

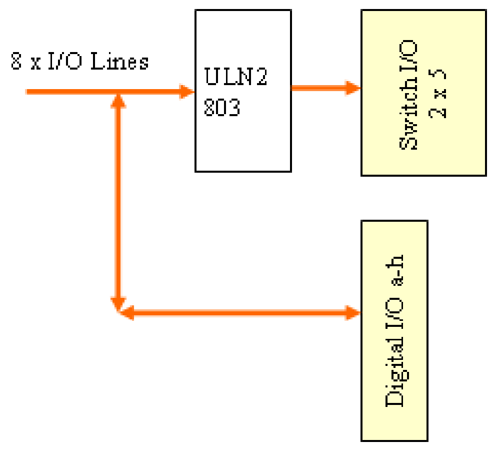
USB

When using the USB as the main controlling point it is important to note that when a USB is plugged into a host (the PC). The host will 'enumerate' (find out what's just been plugged in) and while it does this the power supply provided by the PC will fluctuate up and down. This does not effect the relays as they will remain off, however it will mean that the host software (your software) will need to take this into account. This happens with all USB devices. The easiest way to accommodate this is to delay a few seconds before sending commands to the device.

This would not normally be a problem if the device is permanently plugged into the USB as the host has long enumerated before the host software can be come active.

Digital I/O

The digital I/O section is a digital port that can be used for input and output. It also has connected to it a ULN2803 device which is a high power high voltage output device. This will enable switching of loads up to 500mA at 30+ volts.



The block diagram shows the arrangement. The 2 x 5, 10 way connector is output only and from the high power switch, whereas the 8 pin connector is input and output. Also observe that they share the same lines.

By default, i.e. at switch on the port is set to input and the command:

<esc?[r

will return 0 (zero) (try it). This is because of the input resistance of the ULN2803 keeping the line low, this is 2k or so. Connect one of the pins to +V and repeat the command. +V can be found on either JP1(ADC) or JP2(DAC) connectors.

To use the port as an **output**, use the command <esc>[Ns, where 'N' is a number from 0 to 255 (0x0 to 0xff). The port is 8 bits wide and the number represents those bits, a 0 is output and a 1 is input. Here are some examples:

Channel	0x	de	

h	g	f	e	d	c	b	a			
1	1	1	1	1	1	1	0	FE	254	Set all channels to i/p except a which is o/p
0	0	0	0	1	1	1	1	0F	15	Set a-b to i/p, set e-f as o/p

The command for the first example would be **<esc>[254s**

The command for the second example **<esc>[15s**

High Power Output

The data sheet for the UNL2803 is in the resources zip file, and discrepancies between this document and the data sheet is a mistake and the data sheet should be assumed correct.

The logic diagram has been repeated here at FIG 2. The channels and pins have also been added so that channel 'a' is pin 1 of the 2 x 5 connector etc.

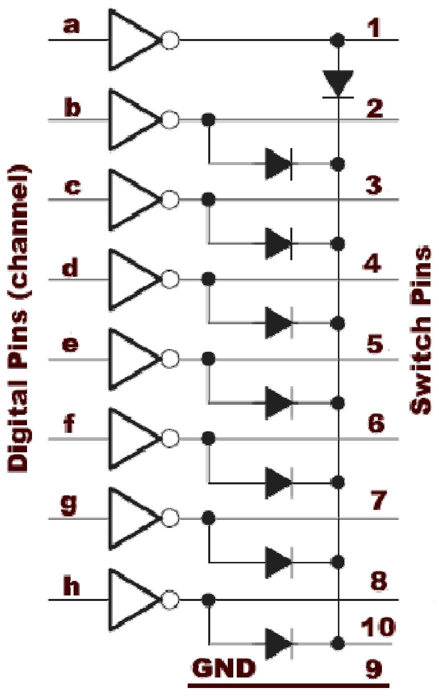


FIG 2

When driving none inductive loads pin 10 can be left disconnected thus:

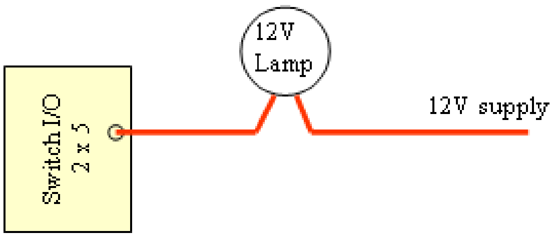


FIG 3

The device acts as a switch to ground and so setting the channel to output and then setting it high* would cause the lamp to illuminate. *High because the output is inverted.

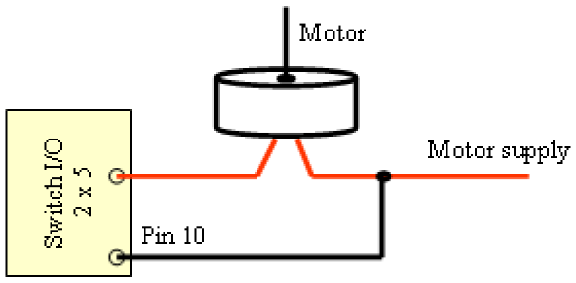


FIG 4

In this instance, because the motor is an inductive load (anything with a coil in it) pin 10 is connected to the power supply. Again the switch will pull the line to ground so sending the command to put the channel high(see above) will cause the motor to run.

PWM

Although the outputs are digital they use pulse width modulation and so in the above example, if the motor was connected to channel ‘a’:

<esc>[255a	would run the motor at full speed and
<esc>[128a	would run the motor at half speed

Well, half the power would be applied to the motor, this would not necessarily result in the motor running at half the speed.

Analog to Digital Input

There are four analogue to digital channels that are being continuously monitored by the firmware. To obtain the value on channel 0 for example the command would be:

<esc>[0D

If nothing is connected this will return 1023 which represents the maximum value. The reason for this is that all four channels are connected to +V via 100k resistors. If you happen to have a 100k resistor and connect this to ground and pin 1 (channel 0) then a value of around half of the above will be obtained. The actual value will be higher than half, probably above 600. The reason for this is the voltage reference, which by default is 4.096V and given that the +V supply is nominally 5V the voltage applied to the ADC channel 0 is 2.5V.

If 1023 represents 4.096V then each value represents 4.096/1023 = 0.004V So 2.5V / 0.004 is 625. This will be close to the value that is observed.

The voltage reference is a precision reference and can be adjusted to one of three values via a command. This is very useful for example if you have a device that say only varies a small amount on a lower voltage. To give an example the LM35 temperature sensor IC gives out 250mV at 25 degrees C and 200mV at 20 degrees C. Using a voltage reference of 4.096 would give a value of 50 and 62 respectively for 20C and 25C. This does not give much resolution (12 points between the two readings). Setting the voltage reference to 1.024 however gives 200 and 250 which is a 50 point difference making the potential for accuracy greater.

Digital to Analogue

The DAC is facilitated by two 5k digital potentiometers. Each side of the potentiometer is connected to ground and +V and the wiper can vary, under command, between +V and ground.

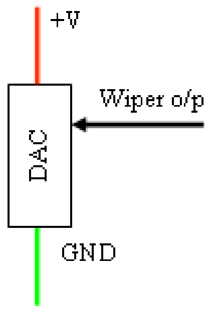


FIG 5

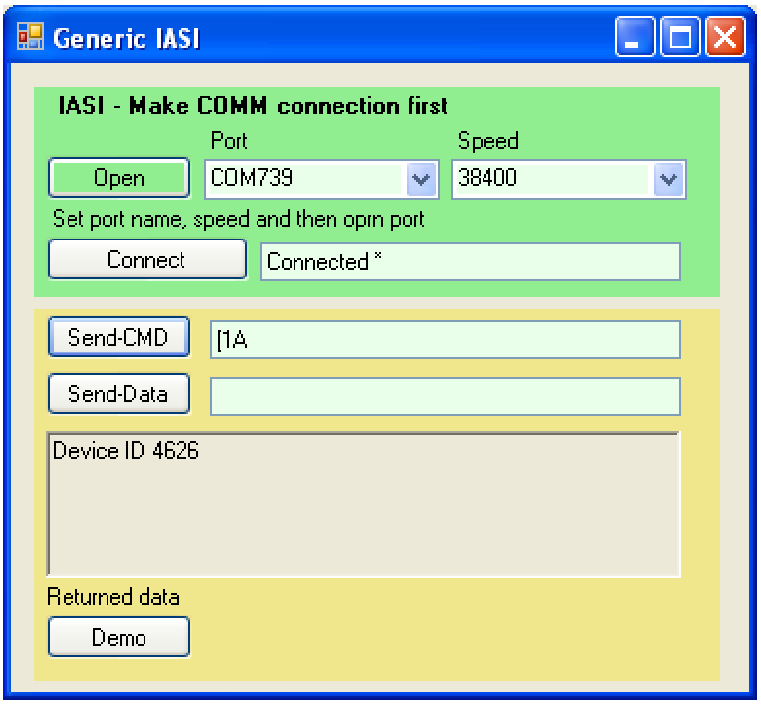
The device used is the MCP4011, 64 step digital potentiometer and the data sheet for this device is included in the resource zip file. The command will 'turn' the wiper to the desired position so for example the command:

<esc>[32X

will set the voltage on pin 2 of JP2 to approximately 2.5 volts representing a potentiometer that has been turned half way.

Visual Basic Demo

This demonstration (on the resource zip) is not a complete program as such but is simply meant as an illustration as to how the device may be used. In fact it is a general purpose program that can be used with any 'Dual' type device.



Port: COM739
Speed: 38400
Set port name, speed and then oprn port
Connected *
[1A
Device ID 4626

FIG 6

To use, run the exe file in bin\debug directory, or load into Visual Basic Express, you may need this anyway. Choose a suitable COM port, connect; this should show the device name in the grey memo box and send a command. The command above turned the rely on. No need for the <esc> key when sending a command.