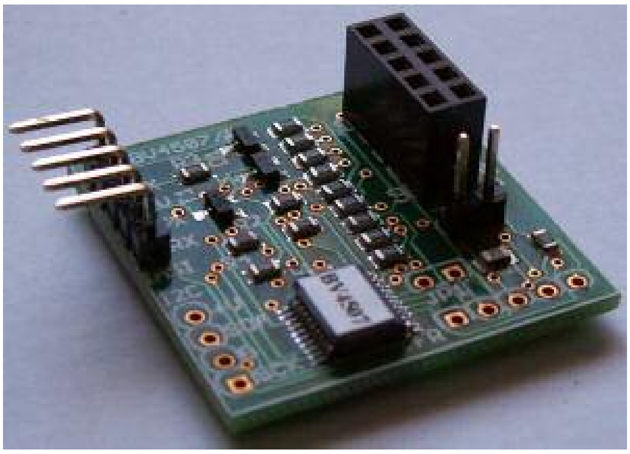
IASI2-9 Channel ADC BV4507



BV4507

IASI2-9 Channel ADC

Product specification January 2009 V0.a

Contents

Rev	Change
Jan 2009	Preliminary

1. Introduction

This is a 9 channel Analogue to Digital Convertor (ADC) that can be connected to a standard PC COM Port or directly to a microcontroller UART.

The 9 channels are capable of 10 bit resolution and there is a feature to allow automatic conversion to take place. The device is controlled by simple text commands using a serial interface.

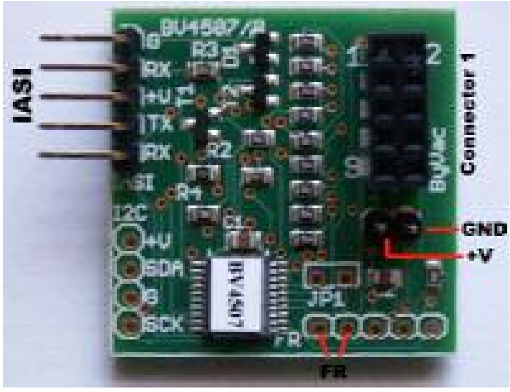
In addition the device has a user programmable address so that several of them can be connected to the same serial bus.

Default IASI address is 'b'

2. Features

- Easy to use asynchronous serial interface requiring only 4 connections.
- Command set based on simple commands
- Multiple devices can share the same data lines.
- Each device has it's own user configurable address, up to 26 devices
- No specialist hardware, can work from a PC Com port or from a microcontroller UART
- Automatic Baud rate detection up to 38.4K from a select set of Baud rates.
- Free Terminal software for Windows
- Works with RS232 standard voltages and +5V, no level translator needed for receiving data
- 9 channels with 10 bit resolution
- Automatic conversion mode
- Choice of voltage reference source
- Socket output for easy interfacing
- Size 32mm x 32mm x 12mm

3. Electrical Specification



Using simple commands the IASI2 interface controls the 9 channel ADC. The input is taken from connector 1.

The 5 pin connector is used for the serial interface, for connection and interface details refer to section 10.

3.1. JP1

A pull up resistor is used on the TX line (pin 2), this is connected to +V via JP1 which has a shorting track on the underside. Only one pull up resistor is required for each bus so should several IASI2 devices be connected to the same bus then the track shorting JP1 should be cut on all boards except one of them.

In practice this will not be required unless there are more than 4 boards on the same bus.

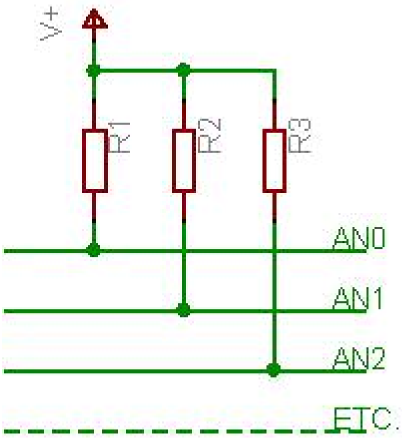
4. ADC Input

Pin	Function
1	AN0
2	AN2
3	AN3
4	AN4
5	AN5
6	AN6
7	AN7
8	AN8
9	AN9
10	AN1/Vref

Connector 1

At the bottom end of connector 1 there is also a pin connector that is connected to +V and Ground. This can be used to provide power for any external equipment.

The input to all of the channels has a 100k pull up resistor as shown.



This does not greatly effect the input impedance but does prevent spurious readings for unconnected channels.

By default the reference voltage is taken from the +V supply but if required this can be taken from pin 10. This is done by sending a command to tell the BV4507 to use pin 10 as a reference.

5. Factory Reset

As the IASI address is stored in EEPROM and is fully configurable, it is possible that the user may forget what the address is. To restore factory defaults do the following:

1) remove power
2) Connect the square hole marked FR to the hole next to it, this will in fact connect the square hole to +V
3) Apply power
4) Remove power

The address and any other parameters that may be stored in the first half of the EEPROM will now be reset.

In general the first 16 bytes of EEPROM are normally reserved for system use but these areas can still be overwritten by the user.

In general the first 16 bytes of EEPROM are normally reserved for system use but these areas can still be overwritten by the user.

6. Operation

There are two modes of operation for this device. The normal mode and autoscan mode.

6.1. Normal Mode

The normal mode of operation is a 3 or 4 stage process:

1. Select channel
2. Do conversion
3. Check conversion has finished (optional)
4. Read conversion results

This mode of operation uses less power as conversion only takes place when requested. An alternative is to use the automatic or autoscan mode.

6.2. Autoscan Mode

This is enabled by issuing a command and once enabled each channel is scanned in turn and the result stored as a 16 bit value. The result can be read at any time. The rate at which scans are performed are set by the acquisition delay and is set by default to 10uS.

The only operation required is to read the result as steps 1 to 3 from the above list are carried out continuously.

Another advantage of using this mode is that because the results are stored, the differential of two reading can be obtained.

6.3. Differential Readings

The differential command is used in conjunction with the autoscan mode and returns the difference between 2 channels. This can effectively double the range that the ADC can produce.

The calculation is done internally and the result is returned as a signed number, the channel numbers used are as follows:

0 AN0 – AN1
1 AN2 – AN3
2 AN4 – AN5
3 AN6 – AN7
4 AN8 – AN9

As an example if AD0=75 and AS1=100 then the result form differential channel 0 would be 75-100 = -25.

7. IASI2 Command set

*NOTE there are two distinct command sets. The system command set, commands normally using upper case and the device command set. The communication command set is described in the introduction to IASI-2 (Intelligent Asynchronous Serial Interface) section **8**.*

*** Commands are case sensitive ***

Device default address is 'b'

Command	Device Command Set
c	Select channel
n	Do conversion
s	Conversion status
r	Fetch Result

e	Enable ADC
a	ADC autoscan
b	Read results from autoscan
d	Acquisition delay
x	Read results from autoscan – differential
j	Result justification
v	Voltage reference source
t	Conversion clock

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 as the IASI2 Protocol is case sensitive.

There are also another set of system commands that are in upper case, see section 17 for details of these.

7.1. Command 'c'

Name: **Select channel**

Command Parameters: **0 to 9**

Typical use **bc3**

Selects a channel for later conversion.

7.2. Command 'n'

Name: **Do conversion**

Command Parameters: **None**

Typical use **bn**

After selecting the channel this command will instruct the ADC to carry out the conversion. The conversion itself takes approximately 2uS but this will be influenced by the serial interface.

7.3. Command 's'

Name: **Conversion Status**

Command Parameters: **None**

Typical use **bs**

This command will return 1 if the conversion is still going on otherwise it will return 0.

7.4. Command 's'

Name: **Fetch Result**

Command Parameters: **None**

Typical use **br**

Returns the result of the ADC conversion for the selected channel as a decimal value in the range 0 to 1023.

7.5. Command 'e'

Name: **Enable ADC**

Command Parameters: **0 or 1**

Typical use **be0**

By default and at reset the ADC is enabled, however some applications may wish to turn off the ADC to save power.

7.6. Command 'a'

Name: **Enable Autoscan**

Command Parameters: **0 or 1**

Typical use **ba1**

By default autoscan is switched off, see the description of what autoscan does in section 6.2.

Using a parameter of 1 (ba1) will enable autoscan. Note that in this mode the normal scanning method of select, convert and read is disabled.

7.7. Command 'b'

Name: **Read Autoscan Results**

Command Parameters: **0 to 9**

Typical use **bb0**

When autoscan is enabled the results in the internal RAM are constantly being updated, this command reads the results of a particular channel as represented by that RAM location.

The result is a 16 bit number ranging from 0 to 1024.

7.8. Command 'd'

Name: **Set Acquisition Delay**

Command Parameters: 1 **to 255**

Typical use **bd10**

As autoscan goes from one channel to the next a short delay is introduced. The delay is necessary for the ADC to be able to select, read and store the channel. This is set at 5uS by default and this command allows that time to be changed.

Each value represents 0.5uS, the default value is 10 which gives a delay of 5uS. The maximum value is 255 which represents a delay of 127.5uS. The minimum delay value is 1 which represents 0.5uS. If 0 is entered this will be converted to 1.

7.9. Command 'x'

Name: **Read Autoscan Results - Differential**

Command Parameters: **0 to 4**

Typical use **bx0**

See section 6.3 for a description of what the differential reading is.

This command does not check for a valid channel number so any number outside the above range will give unpredictable results.

7.10. Command 'j'

Name: **Result Justification**

Command Parameters: **0 or 1**

Typical use **bj0**

By default the 10 bit result is right justified in a 16 bit field, the higher bits (11 to 15) are cleared to 0. This gives a result range of 0 to 1023 in decimal.

This command can change the justification to be left of the 16 bit field occupying bits 15 to 6. This effectively multiplies the result by 64.

Parameter value:

1=right (default)

0=left.

7.11. Command 'v'

Name: **Voltage Reference Source**

Command Parameters: **0 or 1**

Typical use **bv0**

By default and at reset the voltage reference is taken from +V and pin 10 of connector 1 is channel 1. This will give a full scale reading (1025) when a channel = +V.

An alternative voltage reference can be supplied to pin 10, from a voltage reference for example, and this command is used to make use of it.

Parameter value:

1=Voltage reference taken from +V (default)

0=Voltage reference taken from pin 10

7.12. Command 't'

Name: **Conversion Clock**

Command Parameters: **0 to 2**

Typical use **bt1**

The ADC conversion speed can be slowed if required using this command

Parameter values:

0=2uS (default)

1=4uS

2=8uS

8. Revisions to the IASI-2 Section

Rev	Change
Oct 2008	Preliminary

9. Introduction to IASI-2

The Intelligent Asynchronous Serial Interface (IASI-2) 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 text command set and a flexible hardware and software interface. The 'Intelligent' aspect is derived from the fact that each particular IASI-2 knows about the connected hardware so a simple command can make the hardware perform a reasonably complex function.

When used in a microcontroller system this enables the controller and designer to concentrate on the important aspects of the design and control rather than the mundane job of controlling the hardware. It also means that the task of driving common peripherals is not being constantly re-invented.

10. IASI-2 Electrical Interface

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

The interface is specifically designed so that it can be connected to either a standard com port (on a PC for example) or directly to a microcontroller UART or even a microcontroller port pin with a software generated UART (Universal Asynchronous Receiver and Transmitter). A five pin connector is used with normally only 3 or four pins being connected at any one time.

There are **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 an 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 Baud rates: 9600, 14,400, 19,200 and 38,400.

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

11. Serial Connections

The device is designed to work in either of **two** modes: an INVETED mode for connecting directly to an RS232 port (factory default) or a NON-INVERTED mode for connecting to a microcontroller UART.

As previously described there are two inputs, one for each alternative interface. On the transmit side (output from the interface) there is only one pin that takes care of inverted and non-inverted logic, this is configured in software. The output is 0 to +5V only, rather than the RS232 specification requiring positive and negative signals.

On most RS232 specification interfaces this will work although it is not within the actual RS232 specification.

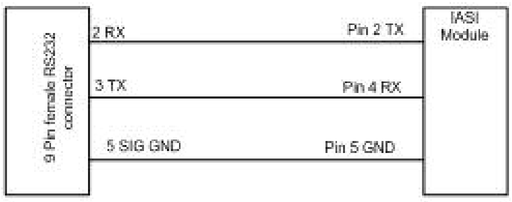


Figure 1 Connection to a PC

Figure shows the connections to a 9 pin D type

Pin	Name	Description
1	RX	Receive data in non-inverted form at +5V logic levels. Use this pin for connecting to MAX232 devices or directly to microcontrollers.
2	TX	Transmit (output) data. This is 0V and +5V, RS232 levels are not used. Devices will work without this connected but no feedback can be received. This pin is configurable in software to transmit either normal or inverted logic. (see multiple devices section 21.1)
3	+5V	Standard 5V power to the device
4	RX-Invert	Receive data (input) this will accept -12V to +15V volts in inverted logic as is normally available on a PC Com port. The format is RS232 1 start bit 8 data bits and 2 stop bits.
5	GND	Ground

Table E2 Serial Connection Details

connector found on most PC's.

12. Start Up

The interface will wait for a Carriage Return (#13) from either the inverted or non-inverted input in order for it to establish a Baud rate. The Baud rate is determined from a fixed range 9600, 14,400, 19,200 and 38,400.

No feedback is given and so it is possible wise to send more then one CR just in case. Once the Baud rate has been established the interface is ready to receive commands.

13. Command Format

All devices have an address which is one byte in the range 97 to 122 (0x61 to 0x7A), this corresponds to the printable ASCII characters 'a' to 'z'

The **default address is 'a'** 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 usually bytes that correspond to upper case characters 'A' – 'Z' and there are device specific commands using higher values that correspond to lower case characters 'a' – 'z'.

This section deals with the system commands.

14. Numbers

Some commands require an ASCII coded number and other commands require a byte, for example when specifying the brightness of the LED display the command is **aj4**.

'a' Is the address

'j' Is the brightness command and

'4' Is the value of the brightness.

This command is specified as an ASCII code so the actual bytes sent to the LED device is:

97 (a) 106 (j) 52 (4)

Note that the '4' is sent as byte 52 (0x34) and not the byte 4.

This is convenient when directly typing commands at a terminal but can cause confusion when using code. As a generalisation if a byte value is required then the code will be something like:

Send(#4)

But if an ASCII coded commend is required as in the above example, it would probably be sent as text:

Send("aj4");

15. Factory Configuration

When an IASIM (Intelligent Asynchronous Serial Interface Module) leaves the factory it is usually configured to address 'a'

Factory settings can be restored normally by shorting two connections with a piece of wire and cycling the power.

16. Non/Inverted Mode

As previously mentioned the device is capable of operating with a standard RS232 communication port (inverted) and a microcontroller (non-inverted). The device will accept either signal but will output only one and at reset this is inverted

17. 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.

When a command has successfully completed it will return the byte value 62 (0x3e) (displayed value '>') This can be detected by software as an acknowledgement (ACK).

There are a few special commands that enable discovery of the devices and system wide defaults.

17.1. Command 1

This is the discovery command and it is a byte with a value of 1 that needs to be sent to the device, this can be done on a terminal by (Ctrl-A). On receiving this, the device will send back its address. This however is done in a timely fashion with address 'a' being sent first and 'z' last. Each device has 30ms to send its address and will wait its turn, therefore the device with address 'z' will wait 26x30=780ms to send its address.

The address is sent along with the ACK '>' As an example if 3 devices were connected to the bus 'a', 'f' and 'p' the response to command 1 would be:

a>f>p>

17.2. 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.

17.3. Command 4

At reset the output from the device will be inverted, this command will set all devices to non-inverted. This command should be used if the devices are connected to a microcontroller. Or if a USB-TTL (BV101) type device is being used. The start up sequence for example would be:

CR
CR ; to establish baud rate
Command 4 ; to set non-invert

At this point a discovery command (1) could be sent to see if all of the expected devices are working.

18. Addressable Commands

The next block of commands are directed at a single device and so need an address before sending the command.

The default address of a device is 97 ('a') By convention these commands are in the range 65 to 90 giving a printable character of 'A' to 'Z', this makes it easier for text input if required.

18.1. Summary

Command	Description
A	Address
B	Write to EEPROM
C	Turn off ACK
D	Delay
E	Turn off error reporting
F	Factory reset
G	Read EEPROM
U	Unlock
M	Macro run at start up
N	Switch to non-inverted
P	Print contents of EEPROM
R	Reset device
V	Version
T	Test macro
Z	Create macro

Note that examples will use the default address of 'a' and the address and commands will be shown as their ASCII code because these can be entered directly from a terminal. The device however will only recognise the byte value so when 'aA' is entered the device will see two bytes 97 and 65

18.2. A (0x41)

Name: **Address**

Command Parameters: **byte 97-122**

Typical use **aAp**

This command is used to set the address of the device. The address is one byte with a value of between 97 and 122, giving 26 possible addresses. The range has been chosen because it renders the values as printable characters in the range 'a' to 'z'.

To set a device from its default address to address 'p':

aU

aAp

If this is successful the device will return byte value 0x3e which is the ASCII code for '>' Note that this command requires an unlock (aU) before it can be issued, this is a safeguard to prevent the device from unexpectedly changing the address.

The address is stored at location 0 of the EERPOM – see command B.

When a device is used with other devices on the same bus the addresses must be set up individually before placing them on the same bus.

18.3. B (0x42)

Name: **Write to EEPROM as text**

Command Parameters: <address><space>'text'

Typical use **aB10 'Hello'**

This device has an internal EEPROM with an address range 0 to 255. Some of the addresses are used for system and macro storage so care must be taken where this text goes.

No check is made that the system or macro area is being overwritten, the first 16 bytes are reserved for system use so overwriting this may necessitate a hardware factory reset.

The macro area starts at 0xB0 so if there is a macro defined then this should be avoided.

The command format is
aB<address><space>'text'

Where <address> is the starting address of the EEPROM between 0 and 256. There must be a space between the starting EEPROM address and the single text quote. Note that this is a single quote (0x27). The text is written and the command appends a 0 onto the end of it so it will occupy one extra EEPROM space. Hello for example would be stored as:

0x48,0x65,0x6c,0x6c,0x6f,0x0

This is 6 bytes not 5 as may be expected.

18.4. C

Name: **Turn off ACK**

Command Parameters: **None**

Typical use **aC**

Some devices may be adversely effected by the ACK command or the controlling software may not require the ACK #67 byte. This command will suppress the ACK.

The device must be reset to turn it back on.

18.5. D

Name: **Delay**

Command Parameters: **1-255**

Typical use **aD50**

Pauses the device for a number of milliseconds. Some devices may require a small delay between commands particularly when used with the macro facility.

The delay is only approximate and should not be used for timing purposes.

As an example the LCD display required a delay after clearing and cursor home, so the macro would look like this:

aZac1;aD50;at'Hello';

18.6. E

Name: **Turn off error reporting**

Command Parameters: **none**

Typical Use **aE**

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 only way to enable it again is by resetting the device.

Example: aE.

18.7. F

Name: **Factory reset**

Command Parameters: **YeS**

Typical Use **aFYeS**

Sets the device back to the factory defaults, the command must be followed by bytes 0x59, 0x65 and 0x53 which is the ASCII codes for 'Y' 'e' 'S'

This will prompt on completion and will not require re-initialisation, the address of course will now be 'a'.

18.8. G

Name: **Read EEPROM**

Command Parameters: **aGss nn**

Typical Use **aG0 3**

The EEPROM values can be read with this command.

ss is the start address of the EEPROM in **hex**

nn is the number of bytes to read in **hex**

This command will accept ss and nn as number text values, this means that for the command:

aG10 3

The actual bytes sent to the device are:

0x61,0x47,0x31,0x30,0x20,0x33

Note how the 10 for the start address of the EEROM is specified as 0x31,0x30 which is the ASCII code for 10.

In a similar way the command returns the values as text.

Example:

aG0 3

Will typically return:

610DFF>

18.9. U

Name: **Unlock**

Command Parameters: **none**

Typical Use **aU**

The unlock command is required for certain other commands that may change the way the device works. It is a safeguard form accidentally issuing a command, change of address for example.

18.10. M

Name: **Run macro at start up**

Command Parameters: **1 [0 or nothing]**

Typical Use **aM1**

Macro commands are stored at 0xB0 onwards on the EEPROM. This command will set a flag in EEPROM that will be detected by the start up procedure and run the macro.

The macro will be run before the auto Baud detection. Once activated the command will always be run so care should be taken to test the macro (command T) before using this command otherwise a hardware factory reset may be required.

To activate macro at start up issue **aM1**, to turn off macro at start up issue **aM0** or just **aM**. Note the 1 and 0 are text numbers, i.e 0x31 or 0x30

18.11. N

Name: **Non-Inverted output**

Command Parameters: **none**

Typical Use **aN**

Pin 2 on the electrical interface that supplies the output information (Tx line), can be supplied inverted (at reset, start up) or non-inverted. Inverted is used if the device is connected directly to an RS232 PC Com. 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 inverted mode. To set the device to non-inverted use aN, reset is required to set the device back to inverted mode again.

This command is similar to Command 4 except this acts on a device individually whereas command 4 will set all of the devices on the same bus to non-inverted.

18.12. P

Name: **Print contents of EEPROM**

Command Parameters: **<start address>**

Typical use **aP10**

This will take the contents of the EEPROM at the given starting address and output the data to Tx (pin 2) as raw data (bytes) unlike the G command that will output the data as text numbers.

The command will stop outputting either when it reaches a value of 0 in the EEPROM or when the end of the EEPROM (255) is reached.

This command is the opposite of the B command, the B command will write text to the EEPROM and this command will read and output it.

The start address of the message location within the EEPROM needs to be specified, e.g. **aPA0**.

18.13. R

Name: **Reset**

Command Parameters: **none**

Typical Use **aR**

Resets and 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.

18.14. V

Name: **Version information**

Command Parameters: **none**

Typical Use **aV**

This simply returns a sting that contains the firmware and device version information.

18.15. T

Name: **Test macro**

Command Parameters: **none**

Typical Use **aT**

Runs the macro. This is created by the Z command. It is wise to test macros with this command before using the M command.

18.16. Z

Name: **Create macro**

Command Parameters: **see text**

Typical Use **aZac1;at'Fred'**

A macro is created at 0xB0 in the EEPROM space and so if it is used, it is up to the user not to write over it. The whole macro must be specified on one line (maximum number of bytes 63) and ';' (semicolon) are used to separate commands, they are interpreted as EOL when the macro is running. Example

aZaN;aV;aP10;

The above example will change the mode to inverted, print out the version number and print a message stored in the EEPROM at address 0x10. Note that the macro also finishes with a ';' .

19. Error Codes

Error codes will be displayed if the debug level (ZD) is set to greater than 0.

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 usually caused by specifying a hex number (say D0) when a decimal number is required.
5	No terminating quote, for example: **aB10 'Hello** would give this error.
6	Command locked, the command used should be unlocked with the **U** command before using.

20. Connecting and Configuration

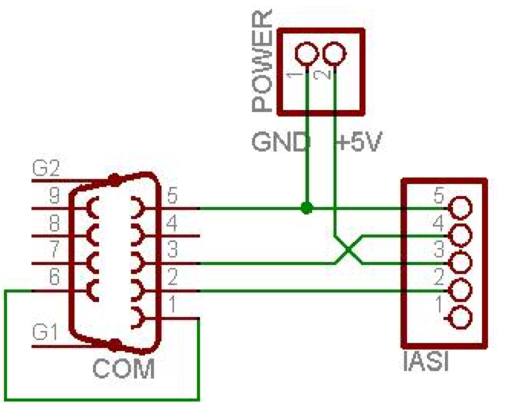


Figure 2 Connection wiring

The above wiring diagram shows the connections to a standard PC 9 way com port (RS232 connector). Pin 1 of the IASI-2 has no connection as this is used to connect to a microcontroller UART.

The factory defaults will work with the above configuration.

Start HyperTerminal or some other terminal software, BV Terminal is ideal and can be obtained from www.byvac.com The following settings should be used:

Baud rate 9600
Start bits 1
Stop bits 2
Handshake none
Local echo on

(The Baud rate can be one of the selected rates, see earlier)

Power up the device and press 'return' a few times. The device should now be listening. Press **CTRL-A**, this will send command 1 to the device, the device should respond with **a>**.

At this point if you are going to use multiple devices than this is where you would set the address. To set the address to 'b' for example the following is required:

aU

aAb

The first command unlocks and the second command sets the device address to 'b'. This can be verified by issuing **bV**, the firmware version should be returned.

20.1. Start Up

It may be that you want to use the device through a line driver device (MAX232) or microprocessor UART without bothering with the PC com port cable. This is also possible.

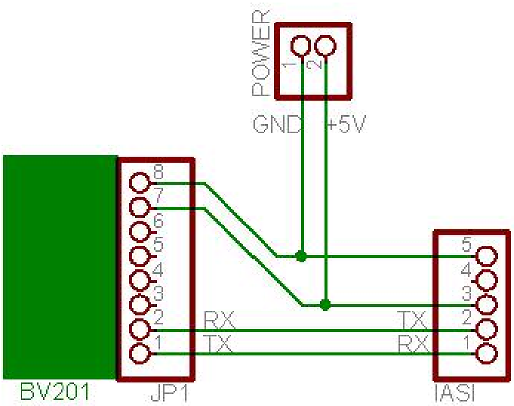


Figure 3 Using non-inverted

The above illustrates the connections used for, in this example a BV201 board that simply translates the PC com port to non-inverted 5V logic levels.

Using a BV101 USB solution could be provided and there would be no need for a separate power supply.

See www.byvac.co.uk for these products.

IASI-2 ALWAYS starts with the output (Tx pin 2) set to inverted mode. If the above connections are used then the device or devices need to be changed so that the output is non-inverted. This is easily achieved by issuing command 4. Once this command is issued all out put is then non-inverted.

Note that the device will revert back to non-inverted if reset or powered of and on again. This provides a consistent and easy to use interface without the additional complication of configuring the device.

21. Microcontroller Use

The output from a microcontroller UART is non-inverted, the Tx pin of the microcontroller will go to the Rx pin 1 of the IASI-2 device.

The start up code could consist of the following:

1) Set the Baud rate of the UART to match one of the rated for IASI-2.

2) Send CR (#13) 3 times: this will establish the Baud rate for any connected device.

3) Issue command 4: this will make all the devices use non-inverted output from now on.

4) Issue command 1: the devices on the bus will respond and reply in non-inverted mode through pin 2.

All of the devices are now ready to be used in the non-inverted mode. Each time the device is reset, either command N or 4 should be used to set the output to non-inverted.

21.1. Multiple Devices

In both modes, inverted and non-inverted many devices can be connected together and all will receive the correct input.

The output however on pin 2 is connected to an open collector and this must have ONE resistor to complete the circuit. This resistor is on each device by default and is permanently connected via a PCB shorting track.

To connect more than one device ideally only one resistor (one track shorted) should be used, the other tracks cut to accommodate this. In practice however several devices can be connected without any ill effects.

IASI – 1 Jumper x			
3	5	1	2

IASI -2 Jumper			
3	5	1	2

3 – Connect +5V together
5 – Connect Ground together
1 – Connect RX together
2 – Connect TX together

The above illustrates this principle where only one jumper is connected.

A side effect of this is that signalling can only be obtained by pulling the output low and so feedback can only be obtained on **multiple devices using the non-inverted mode**.

22. Restoring Factory Defaults

Factory defaults can be restored either by software or hardware. The factory default condition is:

Address = 'a' (#97)

CR value = #13

22.1. Software

Issue the command **aZYeS**.

22.2. Hardware

This is likely to be needed if you have accidentally changed the contents of the first 16 bytes of EEPROM.

1. Power down the device.
2. Temporarily connect the two holes on the device together as shown. If the picture does not match exactly, then look for 5 holes in a row, at one end there will be a square hole, this is hole 1. Connect together holes 1 and 5.
3. Power up the device, this will restore the factory settings.
4. Power down the device.
5. Remove the shorting link.

The device is now restored to the factory settings. NOTE that if a macro was programmed at the factory this will no longer show. On an LCD device for example it will not show the ByVac screen as it did when it left the factory, just the cursor will show.

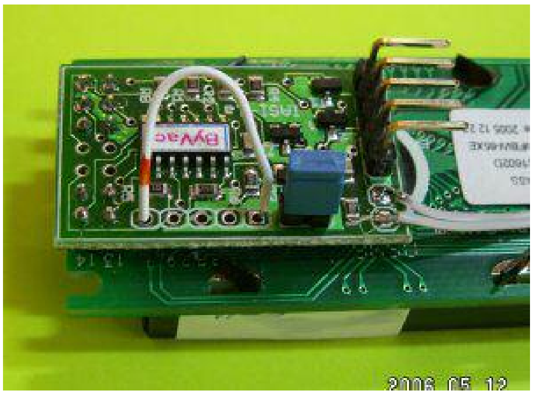


Figure 4 Example Shorting link