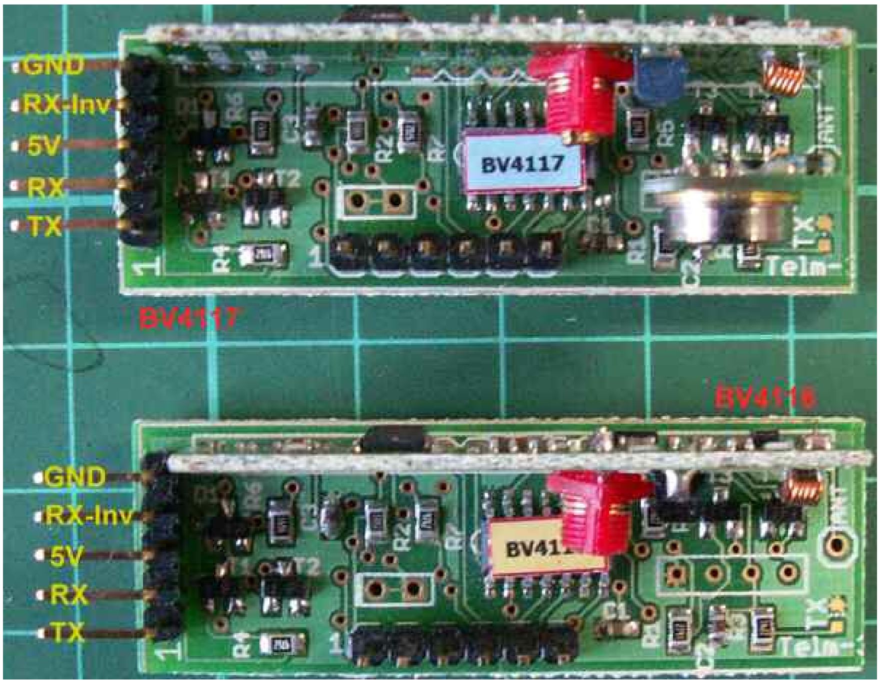


BV4116

RF Packet Receiver

Product specification.

February 2007

Contents

1. Document Versions

1.0 February 2007

2.0 October 2007

3.0 August 2009, added troubleshooting and some minor corrections.

2. Board variations

There are two boards available, the BV4116 which is a packet receiver. This can be used stand alone to switch outputs on having a correct packet received. It can also be used as part of a microcontroller system to receive packets form a BV4115 or BV4117.

The BV4117 is a transmitter and receiver, transceiver. This can also be used stand alone for recieveing but is primarily designed to transmit and receive information as part of a microcontroller system.

3. Introduction

The boards are short range (approximately 100m) low speed (2400 Baud) packet transceiver devices. The following description applies to the BV4117 transceiver part. The BV4116 is exactly the same but with the Transmitter and associated firmware removed.

The packet receiver is a dual purpose device, it can be used for receiving packets from the BV4115/7 and translating this into serial form for use with IASI (Intelligent Asynchronous Serial Interface) or It can be used a an independent device, receiving packets from the BV4117 and outputting the results to its digital outputs.

The device is a low data rate receiver and is intended for applications that do not require an analogue or fast response. The application areas for the independent mode are switching on lights, alarms etc.

NOTE the IASI factory default for this device is a fixed 9600 Baud rate rather then the normal Auto Baud Detect.

4. Features

- 255 packet addresses set by user
- Single channel receiver, low cost, TX/RX for the BV4117
- Dual purpose, can work independently
- Receives 433.92MHz. with a packet length of up to 32 bytes
- Approximately 100m range depending on conditions
- Four digital outputs
- Size 27mm x 50mm x 15mm high (BV4117 19mm high)

5. External Interface

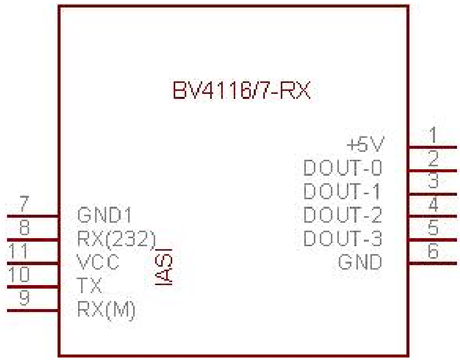
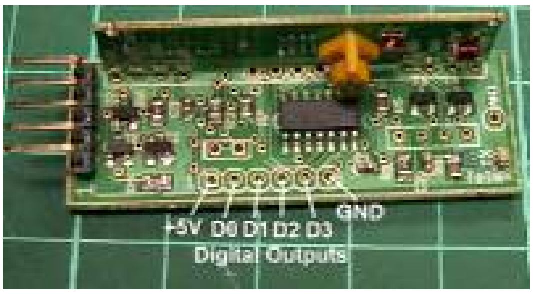


Figure 1 Circuit Symbol Bv4116/7

The BV4116/7 has an IASI (Intelligent Asynchronous Serial Interface) for stetting up the device and for use as a packet receiver.

The four digital outputs can respond to packet information when the device is used for stand alone reception. The application areas for this are switching lights, alarms etc.



Shown here is a BV4116 with the connections for the digital outputs but the BV4117 also has the same digital output functionality.

The +5V and ground are provided for stand alone mode, they are directly connected to the +5V and ground of the IASI connector. If the IASI connector is used then there is no need to connect the power here.

6. IASI Mode

When the receiver is use in this way it enables packets to be read either from a BV4115 transmitter or a BV4117 transceiver. A transceiver could be used instead of this device but if only reception is required then this is a lower cost option as the transmitter module is not required.

It will act as a standard IASI device with the full command set, described later. When used in this mode the BV4116 can still be used as an output device and will still respond to the special packet.

7. Independent Mode

In this mode the device is a 4 channel remote digital switch, it is quite likely that a BV3117 transceiver will be used to control it.

When used in this mode a 'special' packet is sent to the receiver that will activate the outputs.

7.1. Special Packet

The packet required for switching the outputs is distinguished form other packets by two extra bytes that are sent after the address, when these are received the next byte determines the state of the outputs.

<length><address><payload><checksum>

L	A	XX	D	C

Length (8 bits)
The length will always be 4

Address (8 bits)
This is the device packet address and is set using the TA command, see the section on commands. By default it is set to 0x10

XX (16 bits)
This consists of two bytes 0xA3 and 0x55, it is these two bytes that distinguish this packet from any other packet that arrives at this address.

D (8 bits)
This byte which must follow the 'XX' above contains the information required to set or clear the digital outputs. Bit 0 is digital out 0, bit 1 is digital out 1 etc. The remaining bits 8:4 (high nibble) will be ignored.

Checksum (8 bits)
When using the BV4117 the checksum is automatically generated before the packet is sent.

The outputs remain in the current state until another packet is received which may change them.

It should be noted that for the device to receive and act upon a packet the following must be correct:

- Packet address must match the device packet address
- Two bytes must be A3,55 following the address
- The checksum must be correct.

When all of the following are correct the packet will be accepted and the outputs changed accordingly. This is not foolproof but does go a long way to preventing false signals being received.

As an example, assume the receiver address is 0x20 and we are sending a packet from a BV4117. To make pin 1 of the digital output of the BV4116 receiver go low the following command would be issued form the BV4117:

TS0420A35500

To make pin 1 go high the command would be:

TS0420A35501

7.2. Preparation for Independent Mode

There is no need to set the device for independent use as it will be set for this mode from the factory. The default packet address is set to 0x10 for the BV4116 and BV4117.

The following describes the considerations for using independent mode, these differ from the normal IASI devices.

This device is unusual in that it has a fixed Baud rate setting as a factory default. The reason for this is that the automatic Baud rate mechanism, on start up, holds up the processor until user input is generated. If this were the factory default the device would not work until some input had been received form the IASI interface and this is not what is wanted.

The default configuration is ZC0 (see the Z commands) which will fix the IASI interface at 9600 Baud. The checkpoints for using the device for independent mode are:

- Set the desired packet address
- Make sure Auto Baud Detect is disabled, the easiest way to do this is by using the config. command **ZC0 –r**

The packet address is down to user choice and the actual Baud rate is unimportant as this only affects the IASI interface.

8. Packet Description

NOTE: All packet information received and transmitted is in HEX 2 bytes. The case is unimportant. For example the decimal number 12 can be either 0C or 0c, 'c' on its own is not acceptable.

Information is transferred to and from devices BV4115, 6 and 7 in packet form and in short bursts. The packet data rate is 2400 Baud and consist of the following information:

<age><length><address><payload><checksu
m>

All packets transmitted have this format but some of the information is added by the transmitter or receiver. The maximum packet length is 32 bytes including the address and checksum.

The checksum is generated by the transmitter and decoded by the receiver automatically and so no user intervention is required.

The protocol is designed for low speed information transfer. The address, length and checksum are 8 bits but the payload can be any multiple of 8 bits up to 30 bytes. If more information than this is required then multiple packets can be used.

8.1. Receiving

Internally the packets are stored in two buffers, an upper buffer an a lower buffer. The upper buffer will store any incoming information regardless of its destination. The processor will then match its own packet address (TA command) with the packet received, if there is a match and the check sum is valid then the packet is transferred to the lower buffer.

The lower buffer contains information for this device, whereas the upper buffer contains any information that is on the airwaves.

When a transfer occurs from upper buffer to lower buffer the age byte is set to 0xff, this age byte is then decremented at intervals (approximately every 9 seconds). The age byte then determines how old a packet is, the lower the number the older it is. A system may want to reject, or report errors if a packet reaches a certain age.

8.2. Transmitting

This does not apply to the BV4116 but is included here for completeness.

A packet is transmitted in a single burst, this has power saving advantages particularly for battery operated equipment. The transmitter will accept the packet and add the checksum at the end.

9. Commands

Command	Name
Receiver	
TA	Set, see packet address
TB	Print buffer contents
TC	Clear buffers
TD	Discover addresses
TP	Print Packet
TS	Send Packet[1]

[1] BV4117 only

9.1. TA

Name: **Set Packet Address**

Command Parameters: **n [-w]**

Typical Use **TA12**

This will set the packet address for the device, this is not the same as the device address that is set by the ZA command. The packet address will determine which packets are for this device. It is a temporary store so a controller can change the packet address to receive packets from different devices.

For use in independent mode the packet address needs to be permanent, to write the address to EEPROM use the –w switch. This is used on its own to write the current address to EEPROM, thus:

TA –w

Setting the packet address to a new one will clear the buffers (as TB command). The reason is to make sure that command TP only prints valid packets for this device, see the TP command.

9.2. TB

Name: **Print buffer contents**

Command Parameters: **None**

Typical Use **TB**

This is used exclusively for debugging although it could be used for other ‘unthought-of’ purposes.

It shows both the upper and lower buffers, the lower buffer is first, followed by the upper buffer. For more information about buffers see section 8.1.

9.3. TC

Name: **Clear buffers**

Command Parameters: **None**

Typical Use **TC**

This simply resets all of the buffer contents, including the address discovery buffer to 0. This is more useful for address discovery as the addresses do not age and will remain in the buffer until the device is switched off.

9.4. TD

Name: **Discover Addresses**

Command Parameters: **None**

Typical Use **TD**

This is a 10 byte buffer that is filled with all of the addresses that are within range. It is continuously updated and its main use is to let the processor know what addresses are available.

This could be used in a fully automated system where devices may be introduced after the system has been established, an additional temperature gauge for example.

It is also useful for debugging purposes as it will inform the user what the transmitter address is set to.

9.5. TP

Name: **Print packet**

Command Parameters: [*****]

Typical Use **TP**

This is the main packet output and will print a packet in the following format:

<age><packet> -

The age can be inspected to check for an active transmitter, the higher the number the younger the packet. The length can be used by an attached controller to set up a buffer.

This will only print out valid received packets, a packet is considered to be valid if it has the correct packet address and the checksum is valid.

The address is set by the TA command. Note that changing the address will clear the buffers and any valid packet that may be already in the buffer.

The TP command prints:

0000 –

If there is not a valid packet available. Note that the TP command print out is terminated by a hyphen.

9.6. TS

Name: **Send packet**

Command Parameters: [**see text]**

Typical Use **TS03200102**

The send packet command available on the BV4117 only is used to communicate with other BV4116/7 devices.

Using two BV4117 devices it is possible to create a low data rate wireless link. The TA address is not used in this communication but given as part of the TS command. The format is:

TS<Length><Address><Payload>

The length includes the payload and the address, all numbers are specified as hex bytes. The address is the address that is transmitted and so the receiver to pick up this information must have its packet address set to the same address.

Example:

We have a BV4116/7 that will act as the receiver and its packet address is 0x27 (TA27). To send 'Fred" (0x46,0x72,0x65,0x64) to this device the command would be:

TS052746726564

Provided the receiving BV4116/7 has the correct packet address set this data will be revealed on the receiving device using the TP command.

10. Error codes Specific to this interface

Code	Description
22	Packet error

11. Trouble Shooting

This site http://www.asi.byvac.com contains information about the IASI2 (ASI) protocol that does **NOT** apply to these devices. The information for this device is below.

If using on a microcontroller it is better to set the configuration for this first. The default configuration is for RS232 and so any information returned from the device will be inverted which will look odd. Information sent to the device will be perfectly okay provided the correct hardware pin is used.

A recommended initial setting can be sent an written to the configuration register as follows:

zc7 –w nnnnyyn

zc7 –r

The above 2 commands will permanently (until a factory reset or changed deliberately be the user) set the device to be a fixed Baud rate of 9600 (assuming the zb command has not been used), ready to accept an respond to any command via a UART (non-inverted).

11.1. Sending

The most common mistake when sending a command is to not terminate it correctly. The IASI protocol buffers all characters it receives and does not do anything until it receives a CR (byte 13). At that point it will process the command. A common error is to not send any termination character or send CRLF or LFCR (13,10)(10,13). The LF may be interpreted as part of a command and cause errors.

Invert (RS232) and non-invert (UART) does not apply to sending information to the device as the RS232 is connected to a different pin from the UART connection and inversion is carried out by hardware.

11.2. Receiving

If only one device is in the system then it con be configured to listen to every command (as in the configuration recommendation above). On receiving a command or just a single CR, the device will prompt with:

L>

This can be useful because a microcontroller can receive information until '>' and then it will know that all the command has been received. Some packet information is terminated with '-' for the same purpose.

11.3. Antenna

The devices will work quite well without an antenna over a distance of a few metres, this is useful for experimentation and setting up a system. To improve the transceiver distance a short length of wire is recommended. This should be 170mm. This would be a ¼ wave antenna.

The device will only transmit when a packet is sent, and in the case of a transceiver, the receiver is switched off while transmitting. To keep within the law transmission should only take place 1% of the time. Assuming the maximum packet length is used this will be

32+2, 34 bytes (374 bits with 1 start and 2 stop bits). At 2400 baud this will take 156mS. It follows then that the maximum packet transmission rate should be about 1 every 15s. This is the word case, using the smallest packet this would be about 1.5 seconds.

11.4. Packets

A checksum is generated by the transmitter and is checked by the receiver. If it does not match the packet is rejected. There is a greater chance of this happening with larger packets so the smaller the packet the better. It may be more reliable to send 2 x 10 byte packets then 1 x 20 byte packets. This of course depends on the circumstances.

12. Revisions to the IASI Section

Rev	Change
May 2006	Additional information about connecting multiple devices and hardware factory reset.
June 2006 V1.01	Troubleshooting guide added

13. Introduction to IASI

The Intelligent Asynchronous Serial Interface (IASI) 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 interface. The 'Intelligent' aspect is derived from the fact that each particular IASI knows about the connected hardware so a simple command can make the hardware perform a reasonably complex function. Scroll text on an LCD display for example.

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.

14. IASI 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 or it can be configured in software to a fixed, default baud rate. The device detects the baud rate on receiving a CR character (0Dh). Other received characters will be ignored until the Baud rate has been established.

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.

15. 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.

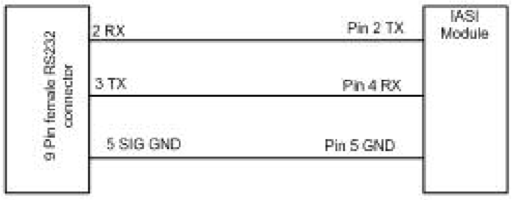


Figure 2 Connection to a PC

Figure E1 shows the connections to a 9 pin D type connector found on most PC's.

The Baud rate is automatically determined by detecting the first Carriage return it receives or it can be configured to a fixed baud rate in software. See the section on commands (**ZC**) for further details.

16. Factory Configuration

When an IASIM (Intelligent Asynchronous Serial Interface Module) leaves the factory it is configured to automatically detect the baud rate and interface with a standard PC com port. Regardless of the configuration the device will receive both forms of electrical interface (inverted and non-inverted) depending on which pin is used. So even if it may not be possible to see the output from the device, it will normally accept the input.

The transmit is configured by software and this is configured to invert. The interface does not need this pin to be connected to receive commands.

Factory settings can be restored both in software and hardware. The command ZF (see later) will restore the factory configuration. To restore factory defaults using hardware see Appendix A

17. Non-Inverted Mode

As previously mentioned the device is capable of operating with a standard RS232 communication port and this is the factory default and is useful for configuring and learning the device capabilities.

Most microcontrollers however operate on non-inverted logic and output 0V for zero and +5V for logic 1. To operate in this mode issue the command:

ZC3 –r

See Section 22 or the command set for more information.

18. 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, see Section 24.

A very simple command protocol is used that essentially operates in two modes. An interactive mode and an addressing mode. The general format of the command is:

Interactive

<command><command-parameters><CR or ,>

e.g. **ZA**

Addressing

:<address><command><command-parameters><CR or ,>

e.g. **:00ZA**

The purpose of the interactive mode is to enable the user to learn the device capabilities and also to be used in a system where only one device is connected to the bus.

This is the default protocol the prompt for this is:

L>

The 'L' indicates that the device is in 'Listen' mode and will react to any command given and also produce error messages if non-existent commands are given.

An alternative protocol is available that allows multiple devices to be connected to the same serial bus. This is configured using the **ZC** command (see later).

In non-listen mode the prompt is:

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 23.4)
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 E1 Serial Connection Details

:>

This indicates that a colon and address is required. In this mode the device will only accept commands with its own address and ignore everything else. In this mode the previous **ZA** command would be:

:00ZA

Assuming that the device address is 00.

Note that all addresses must be preceded by a colon. The prompt can be turned off by the **ZC** command.

18.1. Addressing

All devices have a two **character** address, this enables several devices to be connected to the same serial lines.

The address can be any two ASCII characters and it is case sensitive. Typical examples are **L1 AA 25**, etc. note that **aa** will be a different address to **AA**.

There is a special address **00** (zero, zero) that all devices will listen to regardless of their own address. There is nothing to stop more than one device having the same address, this can be useful if both devices always require the same commands.

As mentioned earlier if the device has been set to interactive mode and address is not required.

18.2. Command Format

The command is a two character string **ZA** for example and is **not** case sensitive, so **zc** would be the same as **ZC**. All commands have two characters and in general the first character is a class of command and the following letter is a subclass of that command. All commands beginning with **Z** are common to all devices and usually deal with configuration and communication.

Other two letter commands deal with the device direct and the same commands may mean different things for different devices.

18.3. Command Parameters

The command parameters can be anything and are specific to that command. In most circumstances spaces are ignored and can be used freely. Spaces do however serve to separate parameters (space is the delimiter) where more than one is required. For example given the command below that is used on the LCD IASM

L>WT3 "Hello"

The '3' requires a space after it so that the number can be distinguished from the text. In general though spaces are ignored and can be used freely.

18.4. Command Line (Buffer)

All commands strings are buffered in a 64 byte buffer and devices will store any text received on the bus. Any lines greater than 64 characters will be ignored and generate an error so it is important not to exceed this figure on an automatic system.

Providing the debug level is set to greater than zero (see **ZD** command) then this will be reported as an error.

When a Carriage Return (ASCII 0dh) is received (configurable) the text string is passed to the command processor. In non-listen mode only devices with a matching address will respond, two or more devices can have the same address if required.

The line end character is set at the factory to be 13 (0dh) and this can be changed using the **ZL** command. Some systems send only this others send a Line Feed (10, 0ah) and others send both, either way round. The IASI will only accept the configured value and ignore or skip over anything else.

A comma (,) can be used to allow more than one command in a single line.

This is especially useful in non-listen (address) mode as the device only needs addressing once:

:>:00ZAL2

:>:L2ZD1* or *:>:00ZD1

can be sent as:

:>:00ZAL2,ZD1

(prompt)

(command)

The above will set the device address to L2 and the debug level to 1.

19. Command Set

Command sets consist of a two character command. The first character usually refers to the general type and the second later is specific to that command. The Z command set for example is the common command set for all devices. The Z commands configure the device interface and communications.

19.1. The Star *

As a general rule most commands issued on their own will return their value in printable format for example **ZD** will return the current debug level 0,1 or 2. This will be 'printed' on the terminal screen.

In absolute terms the values returned are in fact the values 30h, 31h or 32h which correspond to the printable values 1,2 or 3 (the ASCII codes for these characters).

When used with a microcontroller for example it may be more convenient to know the actual

BINARY value. By using a * this will return the actual value thus.

ZD*

The above will return the actual binary value 1,2 or 3 which will not be printable on the terminal screen. This may be particularly important for numbers greater than 9 as the printed version of 10 is in fact 1 (31h) and 0 (30h), two bytes. Whereas the actual value 10 is only one byte and much easier to handle.

20. Common Command Set (Z)

A summary of the complete common command set is given in the table. All common commands begin with Z, as previously mentioned commands are grouped by the first letter so any command beginning with a Z is a common command.

A full description of each command is given after the summary.

20.1. Summary

Command	Description
ZA	Address
ZB	Baud Rate
ZC	Device configuration
ZD	Debug level
ZF	Factory reset
ZK	Ack character
ZL	Line end character
ZM	Record Macro
ZR	Reset
ZT	Test macro
ZV	Version
ZW	Wait

Note that any example given will assume that the device is in interactive mode (listen flag on) so no address is required to precede it.

20.2. ZA

Name: **Address**

Command Parameters: **[*address]**

Typical use **:00ZA or ZA (listen on)**

This command will show, confirm or set the address. To see the address of the current device simply use the command without any parameters:

ZA

This will return the actual address of the device, this will be 00 after factory reset.

ZAK1

This will set a new address for the device, K1. Addresses consist of two characters (including numbers) and are case sensitive so setting the address to AA will be different from aa.

:K1ZA*K1

This command is used for confirming that a device is on the bus and working okay. In the above if device K1 is on the bus it will respond with ACK (see the ZK command for ACK). The single ACK character is usually easier to detect by a microcontroller than the two byte address returned without the * parameter.

20.3. ZB

Name: **Baud rate**

Command Parameters: **[*][rate]**

Typical use **:00ZB or ZB (listen on)**

At factory default the Baud rate is detected by receiving a carriage return character (CR value 13, 0dh) from the sending device. A minimum of two are required, one for the detection and the other for confirmation. The quality and length of the serial bus may also effect the detection process.

The detection will chooses from a fixed set of baud rates:

0. 9600
1. 14400
2. 19200
3. 38400
4. 57600
5. 115200

This has been found to be very reliable depending on the quality of the electrical interface.

The command enables viewing and setting of any of the above values 0-5 that correspond to the Baud rate. If the Baud Detect flag (see the ZC command) is set to off (N) then the Baud rate will be determined by whatever this command is set to.

As an example if ZB is set to 3 and Baud detect is off, the device will communicate at 38400 Baud when switched on. If ZB is set to 3 and Baud detect is off then the Baud rate will be determined by the first acceptable Carriage Return character received.

ZB	Baud Detect	Baud Rate
3	Off	38400
3	On	Set by receiving CR character from host device.

Config	Byte	Ack	Macro	Baud-d	Invert	Listen	Prompt	Echo	
0	1E	n	n	y	y	y	y	n	Inv-interact
1	1A	n	n	y	y	n	y	n	Inv-colon
2	58	y	n	y	y	n	n	n	Inv-auto
3	16	n	n	y	n	y	y	n	Interact
4	12	n	n	y	n	n	y	n	colon
5	50	y	n	y	n	n	n	n	Auto

The Baud rate can also be reported in binary, in the above example:

BR

will return 3 (ASCII value 33h)

BR*

will return the value 3 which will not be a printable character, this is useful for microcontroller input, see the section on the star command extension.

20.4. ZC

Name: **Configure**

Command Parameters: **n [*][-w aaaaaaa][-r]**

Typical Use **:00ZC3 -r**

This is probably the most complex command but will alter the way the device behaves and is only likely to be seldom used. There are 7 'flags' that will effect the device and these are:

1. ACK
2. Macro
3. Baud detect
4. Invert
5. Listen
6. Prompt
7. Echo

The above flags can either be on or off (Y or N) and they have the following meaning:

ACK

When the line end character (ZL command) is received by the device there is an option to send the ACK character (ZK command). If this is set to on (Y) then the ACK character will be sent. See the ZK command for details on why you would want to do this.

Macro

When the device is first powered on, optionally a set of commands can be run, see the ZM, ZT commands. If this flag is set to Y then the macro commands will be run at start up.

Baud detect

The device can operate at a fixed Baud rate or detect the host Baud rate at start up. When this is set to Y the device will wait for input from the controller Terminal to determine the Baud rate, see the ZB command)

Invert

This only applies to the transmit line of the device. When set to Y all transmitted data will be inverted, this is suitable for connection to normal RS232 type interfaces, a PC com port for example.

Listen

If this is set to Y a command will be accepted without addressing, this is useful for single devices connected to the bus and saves having to prefix all commands with colon, address.

Prompt

When set to Y will output on the transmit line a prompt. This is useful for experimenting with the device when using a standard terminal. There are two types of prompt depending on how the listen flag is set, see the section on addressing.

Echo

When this is set to Y all characters received will be transmitted to the transmit line. It is not advisable to have this on when used with automated systems. It may also slow down the command reception process.

The technique for setting these consists of storing a configuration in non-volatile memory and then restoring this configuration later.

There are eight configuration locations that can all be used. When restoring factory defaults the first 6 locations are overwritten with the values shown in the table.

To observe a configuration use:

ZCn or **ZCn***

where n is a number between 0 and 8 using ZCn without a * will produce a sting of Y and N. As an example:

ZC0 will produce **NNYYYYN** given the factory defaults.

ZC0* will return 1Eh which will not be printable using a Terminal but will be more understandable to a microcontroller.

To change a configuration **–w** is used to indicate 'write'. As an example to set a new configuration in slot 6 the following command can be issued:

ZC6 –w NNYYYYN

This will write a new configuration to slot 6 having the same values as the factory default, configuration 0. Any slot (configuration) can be overwritten at any time but using the **ZF** command, factory defaults, will reset the first 6 configurations to that in the above table.

To set the device to a configuration **–r** is used (read), so:

ZC0 –r

will set the device to configuration 0.

NOTES

The purpose of the 8 configuration slots are to enable quick changing of device configuration. In most cases one of the default configurations will suffice and so the user may not be concerned with the individual aspects of each flag.

The first 3 (0-2) configurations are intended for using on PC terminals that have an inverted logic. The next three are intended for use with a microcontroller output that does not have an inverted logic.

There is nothing to stop the user from specifying illogical configurations or even using a slot that has not been previously configured. This can make the device very difficult to communicate with. If this should happen than a hard reset will probably be required, this is described in Section 24.

NOTE The use of this command with –r or -w will cause the device to reset, similar to using the ZR command.

20.5. ZD

Name: **Debug**

Command Parameters: **[*][n]**

Typical Use **:00ZD0**

There are three debug levels available. level 0 will not report any errors at all. This is used where multiple devices are on the same bus and the error reporting may interrupt normal command flow.

Level 1 will report to the normal transmit line and level 2 will report to the device if this is possible, LCD displays for example. Note if the device is not capable of displaying the error then level 2 will be equivalent to level 0.

As in the previous commands ZD on its own will report to the terminal in ASCII form and ZD* will report in binary form.

A new level can be set by following the command with the new level, e.g.

ZD1 set debug to level 1.

20.6. ZF

Name: **Reset to Factory Defaults**

Command Parameters: **none**

Typical Use **ZF**

This will set the device to the factory default configuration, see section 24 for what this is.

Providing there is communication with the device this command can be useful for establishing a known configuration automatically. For example:

ZF
CR (carriage return)
CR
:00ZF
CR
CR

Should set the factory defaults even if the device is in interactive mode or not. This can then be followed by a desired configuration, **ZC3 –r** for example.

20.7. ZK

Name: **ACK (acknowledge)**

Command Parameters: **[*][n]**

Typical Use **ZK6**

When a line end (CR) is sent to the device the contents of the buffer are interpreted for a command and then acted upon. This takes a finite time to complete. In an automated system a delay can be used to allow for this. However the delay must be set for the longest time to ensure that all commands can be dealt with.

Optionally (see the ZC command) an acknowledge (ACK) mechanism can be switched on. If this mechanism is switched on, when the command has completed an ACK will be sent to indicate to the controller that another line is ready to be accepted.

This forms a simple but reliable handshake method that can be used for sending text files to a device or for using automated devices.

The ACK character can be set or displayed using this command.

ZK displays the character represented as a decimal printable number, **ZK*** will return the ACK character as a binary value and **ZK6** will set the ACK character to the factory default value of 6

20.8. ZL

Name: **Line end character**

Command Parameters: **[*][n]**

Typical Use **ZL13**

The line end character (EOL) is important as it informs the IASI to interpret the contents of the buffer. Unfortunately this character is not in universal use, some systems send 13, other send 10 and others send a combination of both which can be either way round.

The ZL command allows the specification of the character that will mark the end of line. This also gives the opportunity for any special requirements, e.g. 0 can be specified.

Examples:

ZL returns the value as a decimal number in text from.

ZL* returns the value in binary form (1 byte)

ZL2 sets the end of line character to 2. **WARNING** if you do this you will not be able to communicate with the device unless you end the command line with <CTRL>B (02)

NOTE:

The system will ignore or skip over any received values that are below a value of 32 (20h) except the following characters:

Escape	17, 1bh (interpreted as EOL)
Back space	8, 8h
ZL character	13, 0dh (by default)

20.9. ZM

Name: **Macro**

Command Parameters: **none**

Typical Use **ZM <see below>**

A macro is a common computer term to allow a sequence of commands to be recorded and played back at a later time. This command allows just that. The playback is normally at start up and allows programming of logos etc. on appropriate devices.

The command is used on its own; enter the command **ZM** and the following prompt appears:

175>

The number 175 (differing devices may have a different number) in front of the prompt shows the space available in characters. This will reduce as commands are entered.

Any valid commands can be used but the syntax is not checked until run time, the macro sequence can be checked with ZT. To exit this command use **Escape**.

Macros cannot be edited, the only way is to enter them from the beginning again. They can of course be sent as a text file from a terminal.

Note that if the macro flag is set, see the ZC command, the commands will be run at start up so care should be taken what commands are entered.

The factory defaults will not change the commands you have entered but it will set the macro flag to 0 so the macro will not run at start up. A brand new device has the macro flag set to 1 so it may be worth while setting this to 0 before experimenting. See the ZC command on how to do this.

The macro can be tested suing the ZT command.

20.10. ZR

Name: **Reset**

Command Parameters: **none**.

Typical use **ZR**

This is the reset command and will reset the hardware and put the communication mode as if it had just been switched on.

20.11. ZT

Name: **Test Macro**

Command Parameters: **none**

Typical Use **:00ZT**

This will test any stored macro and should be used before setting the macro flag.

20.12. ZV

Name: **Version information**

Command Parameters: **none**

Typical Use **ZV**

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

Additional information is also displayed but this does not apply to all devices:

The ZV command is also used for detecting errors and indicating if the factory defaults have been set. The format is:

[WnFn] <version information>

Where n is either 0 or 1. Under normal circumstances this would read [W0F0]. For information about this see the separate headings under watchdog and factory reset.

NOTE that reading the data clears it so if the first read was:

[W1F0] <version information>

The second call of this command would read:

[W0F0] <version information>

20.13. ZW

Name: **Wait**

Command Parameters: **n (1-65534)**

Typical Use **ZW110**

This command will simply suspend the device for a length of time. The time is entered in approximately 10ms intervals, so 100 is 1000 ms which is 1 second.

A typical use is to 'hold back' a text file when downloading or in a macro so that animated displays or actions can be implemented. At any time during the delay if a character is received the delay will abort.

This command is not suitable for highly accurate delays as the 10ms interval may vary, particularly from device to device.

Example:
ZW 200

The above example will delay approximately 2 seconds.

21.Z Command Error Codes

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

Code	Description
1	End of input buffer reached, buffer full. If this happens the whole of the input line will be ignored.
2	Unknown command, the command issued is not in the command table for this device.
4	Non-valid hex, where a command is expecting a hex value and something is entered outside 0-9, a-f, A-F. **NOTE** This is rarely used as most numbers values are in decimal.
5	Incorrect parameter following command. This refers to commands that expect a particular value to follow. This error will indicate that a command has been entered that is not in the correct / expected format.
6	Bad decimal number. This is because the command is expecting a number and something else has been entered. Note that a space should always follow a number.

22. Connecting and Configuration

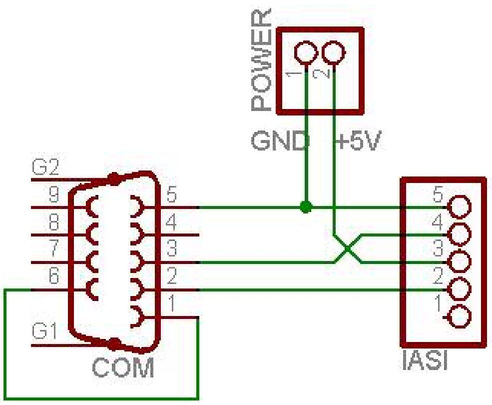


Figure 3 Connection wiring

The above wiring diagram shows the connections to a standard PC 9 way com port (RS232 connector). Pin 1 of the IASI 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 **Error! Reference source not found.**The following settings should be used:

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

(The Baud rate is not that important as the IASI will adjust to the terminals Baud rate)

Power up the device and press 'return' a few times making sure that the terminal is active (mouse cursor in the terminal window). The device should respond with:

L>

when it is ready to accept commands. If this is not the case check the power supply and terminal settings.

The device is now ready to accept commands, try **za**. This will return 00 which is the default device address.

After this prompt the device is now ready to be configured but it may be worthwhile spending some time looking at the command options available.

22.1. Reconfiguring

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.

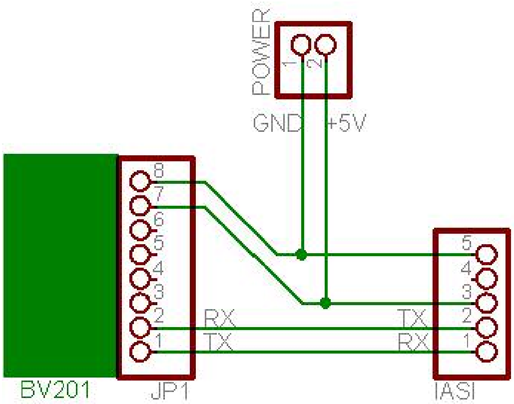


Figure 4 Using non