

ZG1B Manual

1. Document

Document information

Info	Content
Keywords	
Abstract	

Document Revisions

Rev	Date	Description
	14 January 2009	
	10 June 2009	
0.3	20 June 2009	

2. Contents

1. DOCUMENT	1
2. CONTENTS	2
3. INTRODUCTION	4
3.1.1 Product Preview	4
4. ZG1B PRODUCT	5
4.1 ZG1B BLUE Part: Summary of Features	5
4.2 Identification	6
4.3 BLUE Part ID and Ordering	6
4.4 ZG1B Layout	7
4.5 Pin definitions	8
4.6 Pin Types	9
4.7 Electrical Limits	10
4.8 Electrical Characteristics	10
4.8.1 Power supply	10
4.8.2 Voltage Monitor	10
4.8.3 Reset	11
4.8.4 UART	11
4.9 Table of Reset/Start Up Timing	12
4.10 Accessing the Bootloader	12
4.11 Viewing Bootloader ID and App ID	12
4.12 BLUE Part Sample Use Schematic	13
5. ZG1B LCD FUNCTIONALITY	14
5.1 2x16 Character LCD Control Characteristics	14
5.1.1 2x16 Character LCD Controller Interface	14

5.1.1.1 Range of access to LCD controller commands	14
5.1.2 Speed	14
5.1.3 Voltage	14
5.2 LCD Printable and Remapped Control characters	15
5.2.1 Table of added characters for remapped LCD control	16
5.2.2 The six remapped C language escape characters	17
5.2.3 The escape control character	17
5.2.4 The flush control character	18
5.2.5 Caution: interpreting low hex digits	18
5.2.6 Caution: embedded null in a string	18
5.2.7 Bit Commands	19
5.2.8 User RAM character commands	20
5.3 Escape command sequences and sending embedded nulls	20
5.3.1 Using C language to send embedded nulls	20
5.3.1.1 Example of sending a command to turn off then on the LCD	21
5.3.1.2 Example of setting a user RAM character	21
6. ZG1B RE/PB FUNCTIONALITY	21
6.1 RE/PB Interface Characteristics	21
6.2 Rotary Encoders Tested	21
6.2.1 Single Travel RE	22
6.2.2 Double Travel RE	22
6.3 Table of Signals passed from ZG1B	22
7. CONTACT INFORMATION	24

3. Introduction

ZGUS provides the ZG1B product

For general information on ZGUS and ZGUS products please visit web site zgus.com or view document <http://s.zgus.com/documents/ZGUS.pdf>

Updates of this manual can be obtained at <http://s.zgus.com/documents/ZG1B.pdf>

3.1.1 Product Preview

This information applies to products under development. Their characteristics and specifications are subject to change without notice.

There are two practical implications that buyers of preview products should to be aware of. One is that a higher than normal level of support than would normally be expected for a product that is not at a preview stage may become available and the second is that changes to documentation, characteristics and specifications will not necessarily be included in a list of changes made or will later be dropped from such lists.

PRODUCT PREVIEW

4. ZG1B Product

4.1 ZG1B BLUE Part: Summary of Features

The BLUE or 'Button, LCD, UART, Rotary Encoder' chip is an IC that

- Allows convenient interfacing between a UART on one device and momentary push buttons, a rotary encoder and a 2x16 character LCD (HD44780 controller compatible and CMOS compatible)
- Uses a single speed of 2400 BAUD which allows use of inexpensive lengthy unshielded cabling
- Uses few and cheap additional components. No external crystal is required.
- Has a convenient PDIP 28 pin footprint
- Can operate at very low power, turn of the LCD and turn the LCD on again with restoration of display
- Can update data for eventual display even when LCD is turned off
- Does not require pauses between sending data for display
- Has a bootloader for field upgrades
- Can operate between 3.0v and 5.0v
- Provides a means to provide a negative voltage for additional contrast voltage if not operating at 5.0v.
- Has a SMP to boost voltage provided a minimum current is being drawn (such with other parts)
- Will notify if a single button is being held down (as opposed to just momentarily pressed)
- Will notify when a single button that has been held (as opposed to just momentarily pressed) has being released
- Will interface with certain rotary encoders (defined as 'double travel'). Please see [Rotary Encoders Tested](#)
- Has a convenient interface for using with C language.

4.2 Identification

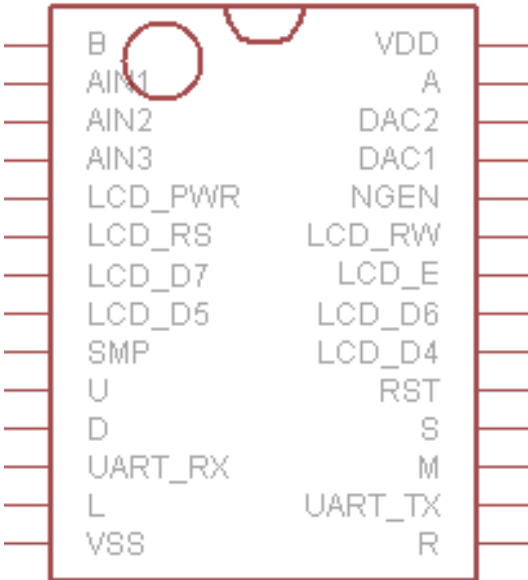
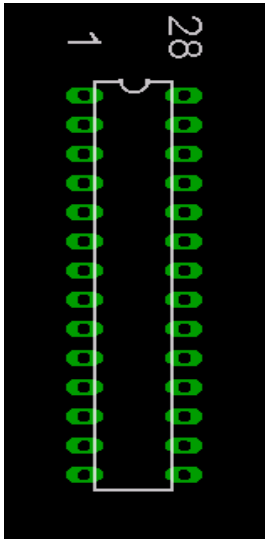
The application name is ZG1B.

Other than identifying marks and documentation that may accompany shipping, the chip, bootloader firmware and app firmware may be identified as indicated in the ZG1L manual <http://s.zgus.com/documentation/ZG1L.pdf>.

4.3 BLUE Part ID and Ordering

A part with BLUE firmware is identified by the prefix ZG1B. Orders can be placed through [ZGUS](http://www.zgus.com). For 'Off the shelf' orders a C9-D28 part is specified with BLUE or ZG1B firmware.

4.4 ZG1B Layout

Part	C9-D28 with ZG1B firmware
Pin/Pad Schematic	
Package	PDIP28
Sample Pad/PCB layout overlaid with Eagle tDocu/Place layers	
Target Socket Module	Breadboard

4.5 Pin definitions

Pin Ref	Pin Name	ZG1B on C9-D28	Pin Type	Description
1	B	1	DI	RE (Rotary Encoder) B Pin
2	AIN1	2	NC	Leave unconnected
3	AIN2	3	NC	Leave unconnected
4	AIN3	4	NC	Leave unconnected
5	LCD_PWR	5	DO	Connect to LCD VCC/Power pin
6	LCD_RS	6	DO	Connect to LCD RS pin
7	LCD_D7	7	DO	Connect to LCD D7 pin
8	LCD_D5	8	NC	Connect to LCD D5 pin
9	SMP	9	PWR	Leave unconnected. Currently undocumented.
10	U	10	DO	Connect to UP Button
11	D	11	DO	Connect to DOWN Button
12	UART_RX	12	DI	UART Receive pin
13	L	13	DI	Connect to LEFT Button
14	VSS	14	PWR	Ground
15	R	15	DI	Connect to RIGHT Button
16	UART_TX	16	DO	UART Transmit pin
17	M	17	DI	Connect to MIDDLE Button
18	S	18	DI	Connect to RE Button or a separate SELECT button
19	RST	19	DI	Pulled Down reset pin. A high signal resets device
20	LCD_D4	20	DO	Connect to LCD D4 pin

21	LCD_D6	21	DO	Connect to LCD D6 pin
22	LCD_E	22	DO	Connect to LCD E pin
23	LCD_RW	23	DO	Connect to LCD RW pin
24	NGEN	24	DO	Used with additional circuitry for LCD contrast voltage. Not required if operating at 5.0v
25	DAC1	25	NC	Leave unconnected
26	DAC2	26	NC	Leave unconnected
27	A	27	DI	RE (Rotary Encoder) A Pin
28	VDD	28	PWR	Power Pin

4.6 Pin Types

Pin Type	Description
PWR	Power
NC	No Connect
DO	Digital Out
DI	Digital In
DIO	Digital In or Out
AI	Analog In
AO	Analog Out

4.7 Electrical Limits

Parameter	Conditions	Min	Typ	Max	Units
VDD supply voltage, Absolute Limits		3.0		5.25	V
VDD supply voltage, Recommended Limits		3.0		5.25	V
UART TX and RX 2400bps 8N1 at 5.0V*		TBD	2400	TBD	bps
VPOR Power On Reset Voltage			TBD		
RST pin high time for reset			TBD		
			TBD		
			TBD		
			TBD		
			TBD		
			TBD		

* At voltages other than 5.0 volts there will be slight variations in BAUD that are expected to remain within acceptable limits.

TBD: To Be Documented

4.8 Electrical Characteristics

4.8.1 Power supply

The BLUE is powered through VDD and GND pins or pads. A valid supply voltage is 3.0 V to 5.25 V on VDD with respect to GND.

4.8.2 Voltage Monitor

The BLUE has an internal power on reset voltage monitor with monitoring voltage VPOR.

If VDD falls below VPOR then the device will be held in POR (power on reset)

4.8.3 Reset

The RST reset pin is a device reset input pin which is pulled down by an internal resistor. Holding this pin high will initiate a device reset.

4.8.4 UART

The UART pins are TX and RX. TX is a digital push-pull output for sending serial UART data from the BLUE. RX is a digital input pin for receiving serial UART data into the BLUE.

The baud rate is 2400 bits per second plus or minus TBD%. The data format is 8N1 (8 data bits, no parity bit, 1 stop bit). There is no handshaking. As per normal with microcontroller UARTs, the STOP state is high.

To communicate direct with PCs using USB there are numerous options with microcontroller development tools.

The purpose of UART is to send button data to an attached device and to receive data for display on the LCD as well as provide some control of the LCD, including turning on and off.

4.9 Table of Reset/Start Up Timing

Please see ZG1L documentation <http://tech.zgus.com/documents/ZG1L.pdf>. This document is distributed with Hidprobe <http://tech.zgus.com/hidprobe>.

If the ZG1M does not have a bootloader then a dummy bootloader ID will be issued

4.10 Accessing the Bootloader

This is covered by a separate document <http://s.zgus.com/documents/ZG1L.pdf>

Updating the App with the Bootloader will override stored values in memory. Hence before an upgrade or App change, a record should be kept of memory contents necessary.

4.11 Viewing Bootloader ID and App ID

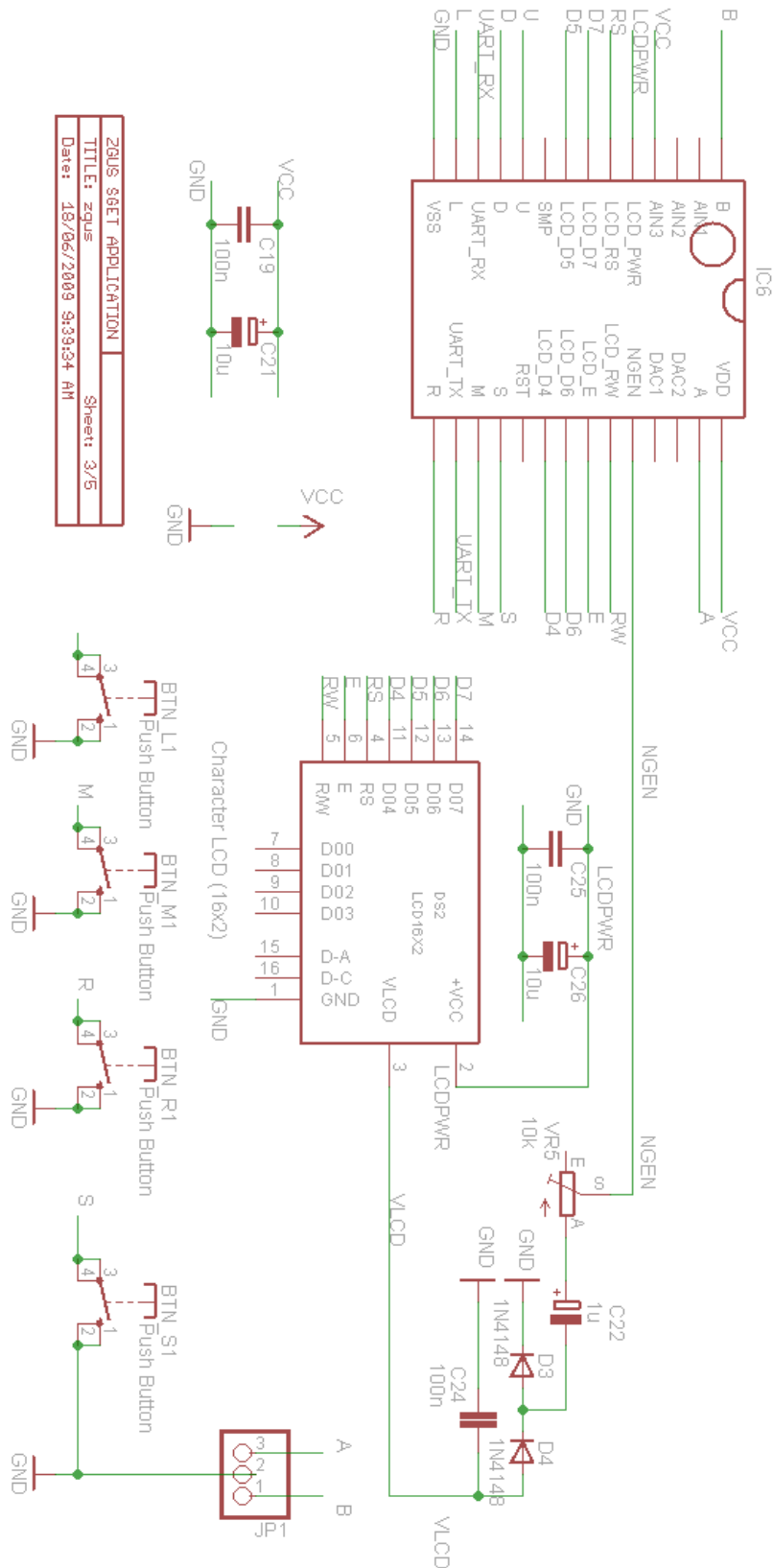
Please see ZG1L documentation <http://tech.zgus.com/documents/ZG1L.pdf>. This document is distributed with Hidprobe <http://tech.zgus.com/hidprobe>.

This also applies if there is a dummy bootloader ID.

4.12 BLUE Part Sample Use Schematic

VLCD is a negative voltage that. Suppose VCC is 3v and VLCD is -1.5v then the contrast voltage is $3 - (-1.5) = 4.5v$

With VCC at 5.0V C13, C20, D1 and D2 are not necessary.



ZGUS SA&ET APPLICATION	Sheet: 3/5
TITLE: zqus	Date: 18/06/2009 9:33:34 AM

5. ZG1B LCD Functionality

5.1 2x16 Character LCD Control Characteristics

5.1.1 2x16 Character LCD Controller Interface

Designed for LCDs driven by modern extended voltage range lower power CMOS controllers otherwise compatible with HD44780 2x16 character LCD controllers. Such compatible controllers are very common. To maintain strict compatibility, an external resistor for the HD47780 compatible controller RC oscillator should be for rated three volt operation.

5.1.1.1 Range of access to LCD controller commands

The ZG1B does not permit full access to all the possible features of HD47780 controller. The more useful LCD controller commands are emulated to allow writing without pausing. These commands are conventionally accessed using C language alphabetic escaped control characters. Please see [Escaped command sequences](#)

5.1.2 Speed

The ZG1B can be written to at 2400 bps without pausing between characters as long as the escape command code is not used to issue some special commands, such as issuing clear and home through the escape command. The ZG1L provides a very convenient alternative way to issue clear and home character LCD commands.

5.1.3 Voltage

Character LCD are normally rated from manufacturers as five volt parts despite the processing circuitry consisting of CMOS circuitry. A detailed examination of datasheets for the processing parts used by manufacturers indicates that these parts can be used at three volts and remain within specification if a single external resistor that is responsible for frequency of oscillation within the HD47780 compatible controller is changed.

We have in fact not changed this resistor in any tests. Not changing the resistor in fact allows power to be reduced. To compensate for specifications not being reached by not altering the resistor we slowed down drive interfacing. Using this method we have not detected any problems between 2.2 V and 3.6 V. However if you must remain within specification then the resistor must be changed.

Normally the contrast voltage is about 4.5 V. Rather than the ground being common between the processing circuitry and the LCD glass circuitry, the positive supply is common. This means the ground of LCD glass circuitry need to be lower than the ground of the processing circuitry. This is achieved by using a square wave from the ZG1B with an external negative voltage multiplier.

5.2 LCD Printable and Remapped Control characters

The terms ASCII below refers to the seven-bit US ASCII character set consisting of 33 control characters from 0x00 to 0x1F with character 0x7E and of 95 printable characters from 0x20 to 0x7E. C language recognises seven control characters with the same control name as ASCII, recognises the same 95 ASCII printable characters (some are escaped) and has a way of directly representing byte values. The same ASCII and C recognised seven control characters have alphabetic escape sequence representations for them in C. C language implementations for embedded systems use these seven control character names to map the C language alphabetic escape sequences to ASCII control characters.

Standard character LCDs do not recognise ASCII control characters. They use a custom control method. The first 16 ASCII characters are mapped to a duplicated set of eight user definable characters. The next 16 have no defined use, character 0x7E has a printable representation and there are some changes in printable representation. There is also printable representation of most bytes values from 0xA0 and above. Standard character LCD control requires hardware control.

The ZG1L makes use of seven of the eight C escaped control characters for control purposes independent of normal LCD control commands.

The ZG1L ordinarily interprets the first eight ASCII control characters from 0x00 to 0x07 as editable but printable LCD RAM characters. The next eight from 0x08 to 0x0F are ordinarily interpreted by the ZG1L as control characters. A prior command sequence provides exceptions.

Currently byte values from 0x10 to 0x1F are reserved for future use but are currently passed as is to the LCD. Byte values 0x20 to 0x7F are passed as is for printable representation. Byte values from 0x80 to 0x9F are used or reserved for control purposes (currently undocumented). Byte values 0xA0 to 0xFF are passed as is to represent extended printable characters.

5.2.1 Table of added characters for remapped LCD control

ZG1C control character for LCD	Common C implementation	Regular Control	ZG1L control mapping
0x08	\b	BS (backspace)	Back one space
0x09	\t	TAB (horizontal tab)	Forward one space
0x0A	\n	LF (new line)	Go to start of next line
0x0B	\v	VT (vertical tab)	Go to start of first line
0x0C	\f	FF (form feed)	Delete current line and remain in position
0x0D	\r	CR (carriage return)	Go to start of current line
0x0E	\x0e		Escape command followed by additional characters
0x0F	\x0f		Flush command

Table of eight control characters

5.2.2 The six remapped C language escape characters

Six of the seven available C language escape characters are used for specific control purposes.

Three of the six escaped alphabetic control characters are to navigate to the start of particular lines (first, next and current)

Two of the six escaped alphabetic control characters are to move the cursor forward one space or back one space.

The last escaped alphabetic control character is the only one that will perform a deletion without sending escape special command sequences

All of the six control characters can be easily combined to perform convenient control on a 2x16 character LCD without invoking internal LCD control codes and without pausing.

C Language allows strings to be concatenated. Using C preprocessor defines it is very convenient to concatenate these sequences with strings. For example the following is valid in C

```
#define CLEARANDHOME "\f\n\f\v"
printf(CLEARANDHOME "welcome to\nMY APP");
```

Control Task	C Control Sequence
Clear and Home	"\f\n\f\v"
Go to first line	'\v'
Go to second line	"\v\n"
Go to and clear line 1	"\v\f"
Go to and clear line 2	"\v\n\f"
Back Space	'\b'
Forward Space	'\t'

Table of common control tasks using control characters

5.2.3 The escape control character

The C language hex character sequence '\x0E' is defined as an escape control character.

This command is followed by a device address, the number of bytes following and then the bytes. For the ZG1L the device address is 0. Straight away this means puts(str) cannot be used as a null ('\x00') will be embedded. This choice of address for the ZG1L is deliberate, as it will prevent a potential hard to pin point bug failures due to arbitrary embedded nulls that are part of a command sequence. If a driver command is used that depends on detecting an embedded

null then the command will fail, due to address 0 which inserts a null, whether or not there is an embedded null in a remaining arbitrary part of the command sequence.

The free sample source code has a macro to allow strings with embedded nulls to be coded as easily as null terminated strings with C.

Please see section on [Escape command sequences](#)

5.2.4 The flush control character

The C language hex character sequence '\x0F' is defined as a flush control character.

The flush command ('\f') can be used to force devices into a known state provided the number of flush characters sent is greater than the maximum number of binary character that might be expected (updating a user RAM character is an example)

5.2.5 Caution: interpreting low hex digits

Escaped C hexadecimal representations (\x hex digits) that are followed by digits 0-9 or A-F or a to f can be interpreted by C implementations in unexpected manners. For example "\xffog" is interpreted as "\xff" followed by "og" whereas "\xf fog" is interpreted as "\x0f" followed by " fog". Hence it is safer to always write a hexadecimal as two hex digits even if it can be written with one hex digit.

5.2.6 Caution: embedded null in a string

If a C string contains an embedded null such as str= "string1\x00string2" then this string cannot be sent with puts(str) or printf(str). There is an example of a method to send a string with an embedded null in the sample source code conveniently with a C preprocessor macro that adds a small additional amount of code. This might be required to set up user defined RAM characters. Please see section [Using C language to send embedded nulls](#)

5.2.7 Bit Commands

There are seven bit commands corresponding to bits 0 to 5 and bit 7.

Bits 3, 4 and 5 only have an effect if these bits are on.

If Bit 7 is on is on then the ZG1B resets.

Bits 0, 1 and 2 map to bits B, C and NOT D bits respectively of HD4780 LCD controllers. Bits 0, 1 and 2 are updated each time the bit commands are set. They will have no practical effect if the LCD is turned off but must be properly set and repeated if the LCD is turned on. After turning the LCD back on (as opposed to the LCD display), a character needs to be sent. To display the contents unaltered after turning the LCD back on send "\b\f" also.

Bit	Bit Name	Default	Action
0	LCDCMD_BITBLINK	0	Only applies if LCD is on. If 1 blink on at current position, if 0 blink off
1	LCDCMD_BITCURSOR	0	Only applies if LCD is on. If 1 cursor on at current position, if 0 cursor off
2	LCDCMD_BITDISPLAYOFF (1)	0	Only applies if LCD is on. If 1 then LCD display is off. If 0 LCD display is on
3	LCDCMD_BITCLEAR		If 1 clear now, return to home position, if 0 do nothing. This command applies even if LCD is off
4	LCDCMD_BITON (2)		If 1 turn on LCD if LCD is off when a printable character is received and show edited buffer even if characters received when LCD was off. If 0 do nothing.
5	LCDCMD_BITOFF (2)		If 1 turn off LCD if LCD is on now but retain record of current buffer and edit buffer if new printable characters are received. If 0 do nothing
6	ALTERNATIVE MODE	0	If 1 enter alternative mode now, unless RESET ZG1B is also set
7	RESET ZG1B	0	If 1 reset the ZG1B now. If 0 do nothing

- (1) If the LCD is turned on then the display can be turned on or off. The best way to save power is to turn the LCD off, rather than just the LCD display. The contents will still be remembered. A character also needs to be sent after turning back on. "\b\f" will leave contents unaltered (sends a back space then a forward space).

- (2) If both LCDCMD_BITON and LCDCMD_BITOFF are set to 1 then the LCD will be turned off but will be tuned back on again when a printable character is received.

5.2.8 User RAM character commands

There are eight customisable characters. These characters have addresses 0 to 7 and can easily be coded in C language as '\x00' to '\c07'. Please note the warning about embedded nulls applies for character '\x00'.

The ZG1L makes it very easy to encode the customisable characters with C language, including when embedded nulls are sent. Please see example above.

5.3 Escape command sequences and sending embedded nulls

There are two escaped command sequences for the ZG1L. One is to send six bit commands through a single byte and the other is to send a nine byte command to update one of eight user RAM characters. For the ZG1L the type of command is specified by the command length specifier.

Format of escaped command sequence

Escape Character	ZG1B address	Bytes Following (1)	Command content
\x0E	\x00	\x01	Bit commands in one byte
\x0E	\x00	\x09	One byte (\x00 - \x07) specifying ram character then eight bytes of line pixels for the character

(1) For the ZG1L the number of bytes following also defines the type of command

5.3.1 Using C language to send embedded nulls

Embedded nulls (0 or '\x00') cannot be sent with C functions such as puts. The sample free source code includes a workaround which adds a small additional amount of code for sending strings with embedded nulls.

Below are two prototypes and a macro that can be used for sending null terminated strings and for sending strings with embedded nulls. There are definitions of the prototype declarations in the free sample source code.

The putslcd function CANNOT be used if there are embedded nulls in a string. The PUTSLCD macro can be used instead.

```
void putslcd(const char *str);
void putslcdlen(const char *str, unsigned int len);
#define PUTSLCD(A) {const char str[]=A;putslcdlen(str,sizeof(str)-1);}
```

5.3.1.1 Example of sending a command to turn off then on the LCD

```
#define GOTOANDCLEARLINE1 "\v\f"
#define LCDBITSCMD "\x0E\x00\x01" //string contains an embedded null
#define LCDCMDOFF "\x20" //set bit 5 on
#define LCDCMDON "\x10" //set bit 4 on and also set D on, C off, B off
PUTSLCD(LCDBITSCMD LCDCMDOFF);//turn LCD off
putslcd(GOTOANDCLEARLINE1 "LCD off");//edit buffer while LCD is off
PUTSLCD(LCDBITSCMD LCDCMDON ",now on");//turn LCD on again and add
```

5.3.1.2 Example of setting a user RAM character

```
#define LCDRAMCHARCMD "\x0e\x00\x09"
#define BARCHARSTR "\x00"
#define BARCHARDEFN "\xff\xff\xff\xff\xff\xff\xff"
```

```
PUTSLCD (LCDRAMCHARCMD BARCHARSTR BARCHARDEFN);
```

The entire user RAM character pixel display is set with the BARCHARDEFN which contains eight bytes to send. PUTSLCD safely sends the concatenation of three strings, despite the concatenated string containing two embedded nulls.

Please see the free sample source code for a complete example.

6. ZG1B RE/PB Functionality

RE stands for Rotary Encoder. PB Stands for Push Button.

6.1 RE/PB Interface Characteristics

The ZG1B notifies events such as push button pressed notification and rotary encoder rotation notification (clockwise and anti clockwise).

If a push button is held down continually then the ZG1B sends two notifications instead of one. These first is that a push button is being held down. The second event notification is when the held push button is released.

The signals are from 0x80 to 0x93 sent through a UART with two signals unused in this range. This range is outside the normal ASCII range. The UART configuration is 8N1 2400bps (+/- 2%).

Up to six push buttons can be connected. Five have prefix labels BN and the sixth has label RE_SEL as this pin that is intended for use with rotary encoders that include a select button.

6.2 Rotary Encoders Tested

Only double travel Rotary Encoders can be connected!

Two rotary encoders were tested. One is stocked by Altronics Australia (Part Number S3350 from www.altronics.com.au). The other is stocked by Mouser USA (Part Number 318-ENC111F-20P from www.mouser.com).

While we attempted to test both single and double travel rotary encoders (as defined below) we were only able to test double travel REs and so are only offering the ZG1B as developed for and capable of interfacing with double travel REs.

6.2.1 Single Travel RE

The Altronics part datasheet indicates a resting indent only occurs when both RE_A and RE_B are off (or open circuit with the common pin). For the sake of convenience we are defining this as a 'double travel' indent rotary encoder.

6.2.2 Double Travel RE

The Mouser stocked part was purchased as according to the data sheet resting indents occur when both RE_A and RE_B are both on (circuit closed) or are both off (circuit open). This is the normal expected behavior from rotary encoders. For the sake of convenience we are defining this as 'single travel' indent rotary encoder.

However tests revealed the Mouser stocked part resting indents only occur when both RE_A and RE_B are off (circuit closed)! Hence while the datasheet says the part is a single travel RE the part is really a double travel RE!

6.3 Table of Signals passed from ZG1B

All the six held key signals are the same non-held key signals with value nine added

Signal (Hex / Decimal / Decimal – 0x80)	Signal	Short Name	Pin Names
0x80 / 128 / 0	No Key Determined	N	
0x81 / 129 / 1	Middle Key	M	BN_M
0x82 / 130 / 2	Left Key	L	BN_L
0x83 / 131 / 3	Right Key	R	BN_R
0x84 / 132 / 4	Up Key	U	BN_U
0x85 / 133 / 5	Down Key	D	BN_D
0x86 / 134 / 6	RE Select Key	S	RE_SEL
0x87 / 135 / 7	RE A Key	A	RE_A
0x88 / 136 / 8	RE B key	B	RE_B
0x89 / 137 / 9	Indeterminate Key(s)	I	
0x8A / 138 / 10	Middle Key Held	MH	
0x8B / 139 / 11	Left Key Held	LH	

0x8C / 140 / 12	Right Key Held	RH	
0x8D / 141 / 13	Up Key Held	UH	
0x8E / 142 / 14	Down Key Held	DH	
0x8F / 143 / 15	RE Select Key Held	SH	
0x90 / 144 / 16	Unused		
0x91 / 145 / 17	Unused		
0x92 / 146 / 18	Indeterminate Key(s) Held	IH	
0x93 / 147 / 19	Key Released (or Over Key)	O	

7. Contact Information

Business Name and Address:

ZGUS
22 Ragnar St
Edmonton QLD 4869
Australia

Postal Address:

ZGUS
PO Box 793
Edmonton QLD 4869
Australia

Email support@zgus.com
Internet www.zgus.com

Telephone +61 7 4045 3118
Facsimile +61 7 4045 2408

Company Name Auscyber Pty Ltd
ABN 65 089 449 632
Business/Trading Name ZGUS

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Auscyber Pty Ltd assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Auscyber Pty Ltd assumes no responsibility for the functioning of unsubscribed features or parameters. Auscyber Pty Ltd reserves the right to make changes without further notice. Auscyber Pty Ltd makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Auscyber Pty Ltd assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Auscyber Pty Ltd products are not designed, intended, or authorised for use in applications intended to support or sustain life, or for any other application in which the failure of the Auscyber Pty Ltd product could create a situation where personal injury or death may occur. Should Buyer purchase or use Auscyber Pty Ltd products for any such unintended or unauthorised application, Buyer shall indemnify and hold Auscyber Pty Ltd harmless against all claims and damages.