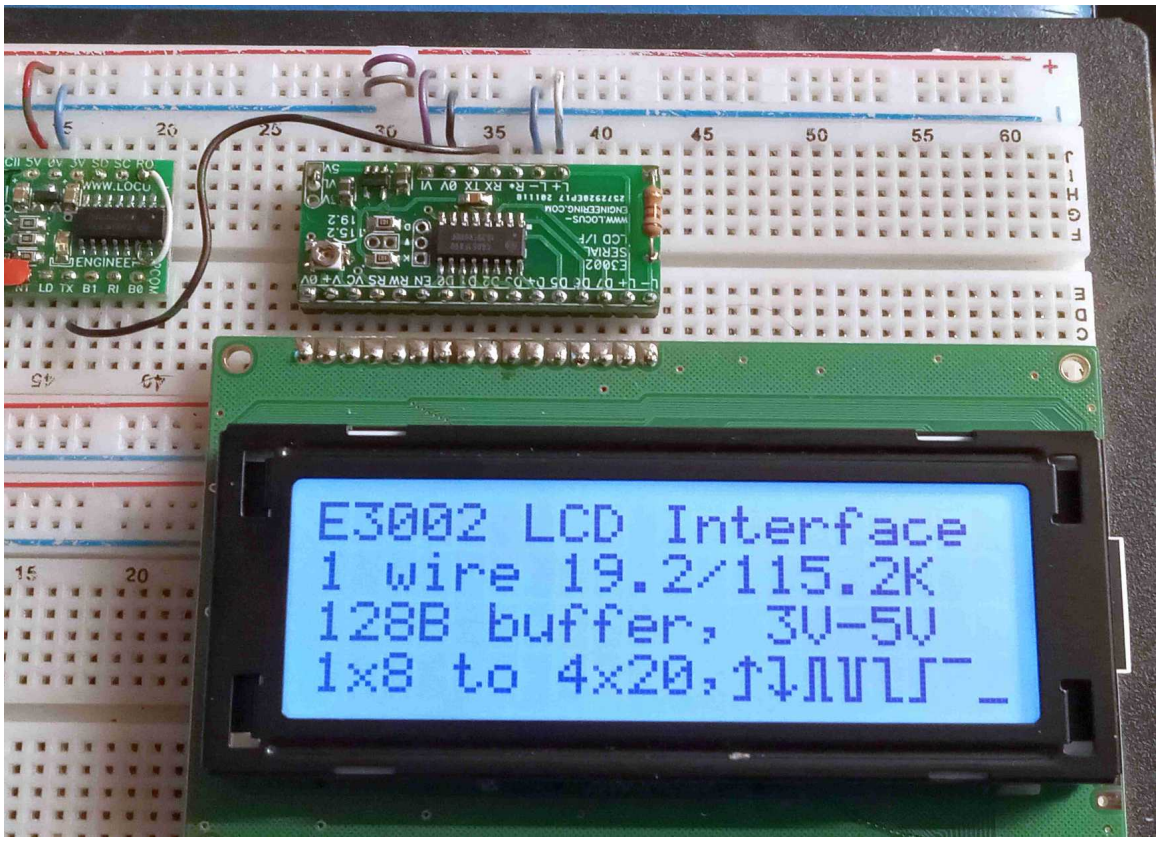
Locus Engineering Inc E3000/2

Single Wire Serial LCD Interface



FEATURES

- **Single wire serial LCD interface for 1x8 to 4x20 sizes**
- **single byte for data, two for instructions**
- **standard and special LCD instructions**
- **handles all LCD timing**
- **fast byte writes to LCD**
- **128 byte buffer**
- **19.2K/115.2Kbaud with echo**
- **contrast adjustment and user backlight R**
- **mates with standard 1x16 LCD pinout**
- **code examples**
- **3V-5.5V operation**
- **0.7”x1.6”, module size**
- **low cost and easy to use**
- **ideal for remote displays or small micros**

DESCRIPTION

The E3000/E3002 Single Wire Serial LCD Interface allows standard LCDs to be controlled with a single serial pin using only single bytes for data and two for instructions. All LCD timing is handled by the E3002. This is accomplished using a command set compatible with the HD44780 and variant LCD controllers. Non-standard commands and delays are easily handled with special instructions. The E3002 are compatible with both 3.3V and 5.0V LCDs using a jumper. A low dropout voltage regulator ensures proper logic levels to the LCD.

Table 1. Absolute Maximum Ratings

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{dd}	Supply voltage			6.0	V
I_{OUT}	Port pin output current			100	mA
V_{rst}, $V_{io,}$	Port pin or reset voltage	-0.3		5.8	V
T_{MAX}	Temperature	-55		150	°C

Table 2. Electrical Characteristics

Test Conditions: Supply Voltage V_{dd} = +3.3V, $T_{ambient}$ = 25° C, unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{DD}	Supply voltage	2.0	3.3-5.0	5.0	V
I_{DD}	Supply current		1.6		mA
V_{IH}	Input high voltage	2.5			V
V_{IL}	Input low voltage			0.6	V
V_{OH}	Output high voltage -10mA	3.2	2.5		V
V_{OL}	Output low voltage +8.5mA		1.0	0.6	V
T_{RESET}	Reset input pulse	15			µsec

General Precautions

Charged devices and circuit boards can discharge without warning. Proper ESD precautions should be followed to avoid failure.

This device is not authorized for use in any product where the failure or malfunction of the product can reasonably be expected to cause failure in a life support system or to significantly affect its operation.

Locus Engineering Inc. reserves the right to make changes at any time without notice to improve product features or reliability.

Information is provided by Locus Engineering Inc. with the best of intentions without any warranty expressed or implied. As such Locus Engineering Inc. disclaims all liabilities or responsibilities for any use of the information, any inaccuracies or fitness for a particular purpose.

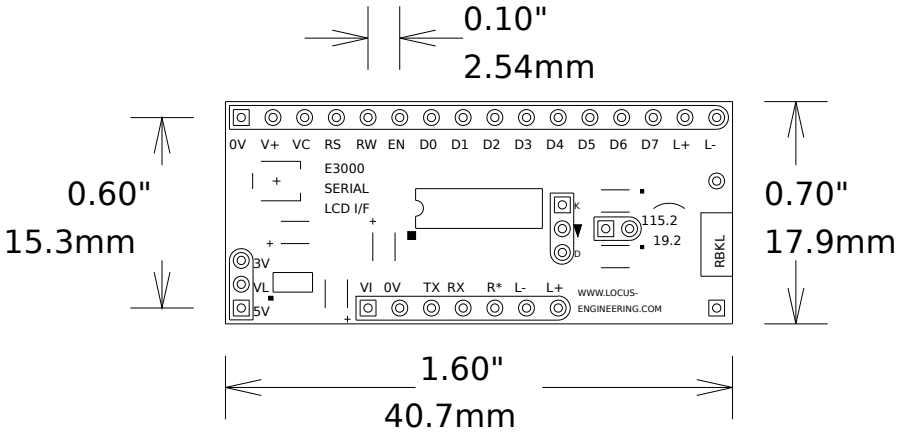


Figure 1. E3000 Module Dimensions

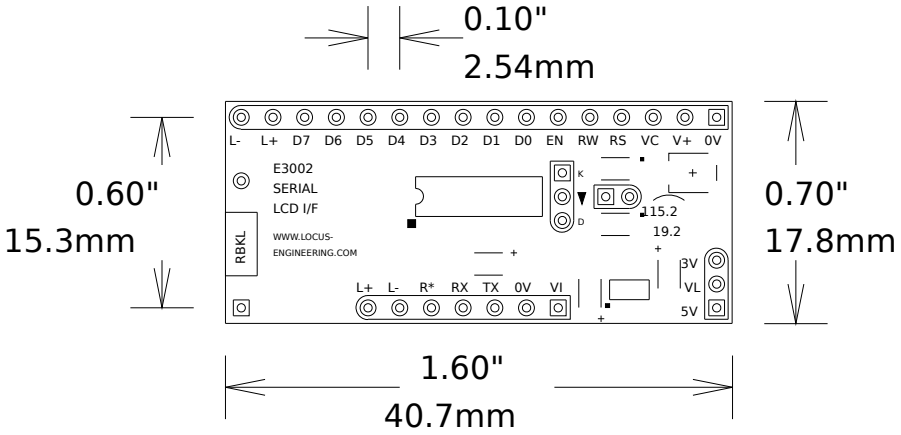


Figure 2. E3002 Module Dimensions

Module Installation

The small module size allows it to be easily piggybacked behind the LCD in either top (E3002) or bottom (E3000) placed 1x16 pinouts. For prototyping on a breadboard the E3002 lines up with the top placed 1x16 LCD connector while the E3000 lines up with the bottom placed 1x16 LCD connector. All pins are 5V tolerant even as the E3000/2 operates from 3.3V.

Only a single serial data line to the E3000/2 module is needed; the serial data echo from the E3000/2 to the host microcontroller is optional.

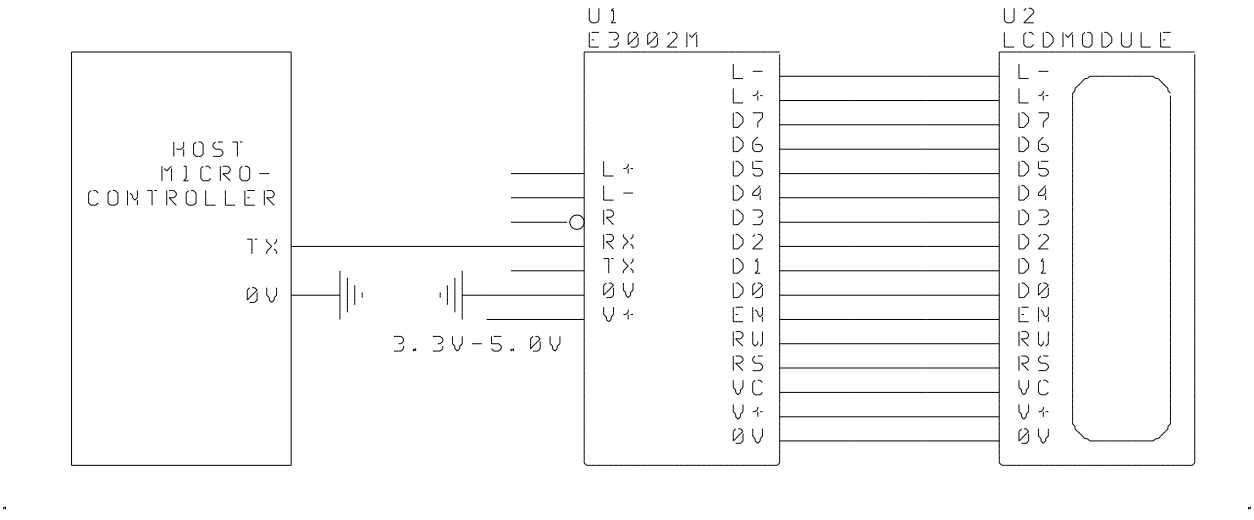


Figure 3. E3000/2 Module Connections

Figure 4. E3000 module on bottom mounted 1x16 LCD connector.

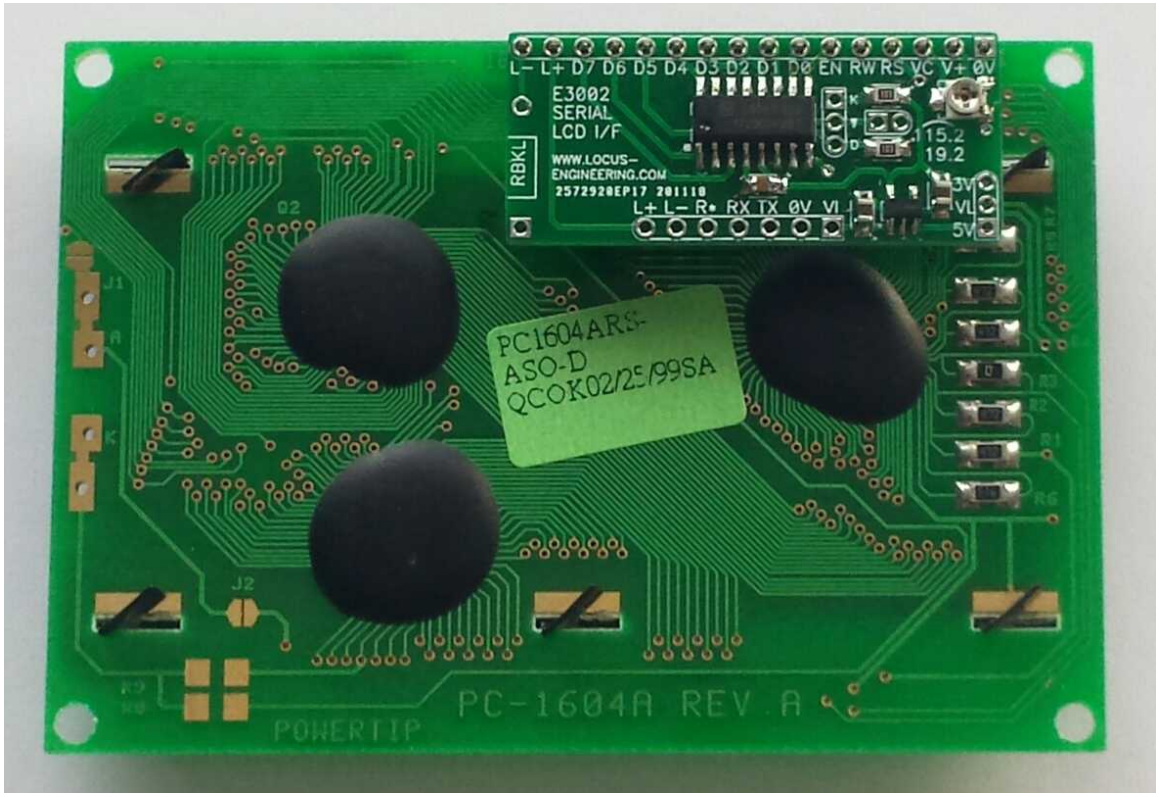


Figure 5. E3002 module on top mounted 1x16 LCD connector.

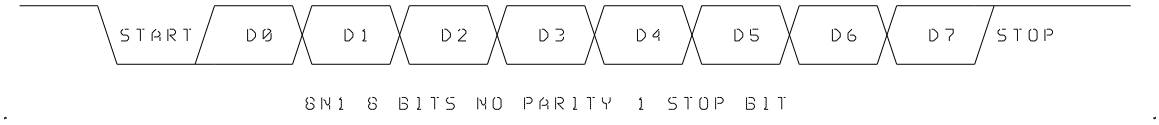
User Interface Pin Descriptions

These connections are controlled by the user.

VI Voltage Input can be between 3.3V to 5.0V typically to match the LCD display requirements. An on-board low dropout voltage regulator powers the micro and guarantees TTL level signals to the LCD. The dropout voltage is only 50mV at 15mA. The regulator powers the LCD but not the backlight which has its own separate connections. A jumper needs to be placed to select either Vin or the 3.3V regulator output for the LCD power.

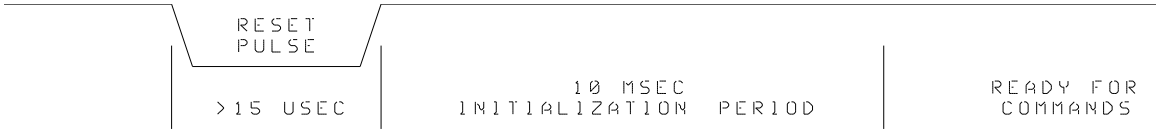
0V Ground is the common return for the module except for the backlight connections L+ and L-.

RX This pin receives the serial commands from the host microcontroller. The serial port is setup as 8 bits, No parity, 1 stop bit (8N1). The bit sequence is start bit, the byte sent least bits first, then followed by the stop bit. Either 19.2Kbaud or 115.2Kbaud are accepted depending on the jumper setting. The default no-jumper setting is 19.2Kbaud. Soldering a jumper sets the baud rate to 115.2K. The baud rate setting can be changed at any time. The host microcontroller Tx port pin needs to be set to push-pull and not open drain.



TX Serial Transmit sends an echo for each byte written to the buffer as it is read to be processed at the selected baud rate. It is delayed at least one baud rate period from the received byte.

R* Reset is an optional active low >15usec signal which resets the module. The buffer is cleared as well as various internal flags and registers. It does not reset the LCD. The module has a power on reset so an RC network is not needed. The reset pin has a 10K pullup to 3.3V. The module is ready for serial commands 10 milliseconds after power up or a hardware reset. The baud rate is reset to the jumpered setting. The LCD should always be initialized after a hard reset since a hard reset will interfere with any instructions in progress to the LCD.



L+ Backlight anode is the positive connection to the LCD backlight. It may or may not be internally connected on the LCD to the supply voltage so check first before connecting.

L- Backlight cathode is the negative connection to the LCD backlight. It may or may not be internally connected on the LCD to the supply voltage so check first before connecting. Note the backlight resistor is in the L- path to the LCD.

LCD Interface Pin Descriptions

Apart from the power connections, the E3000/2 module controls the LCD signal pins.

0V Ground
V+ LCD Supply voltage is determined by the 3V/5V jumper setting.
VC Contrast voltage is determined by the contrast trimmer on the module.
RS Register Select determines if the byte being written is an instruction (RS=0) or data (RS=1).
RW Read/Write is tied low for writes only to the LCD. It is not possible to read the LCD. The E3000/2 module provides the LCD timing plus a margin to guarantee proper operation.
EN Enable is a 3 µsec pulse applied to write data or commands to the LCD.
D0-D7 Data is written to the LCD as a byte for fast execution time. Most data and commands are executed within 80usec once the serial byte is received.
L+,L- LCD backlight anode and cathode are connected to the user side connector.

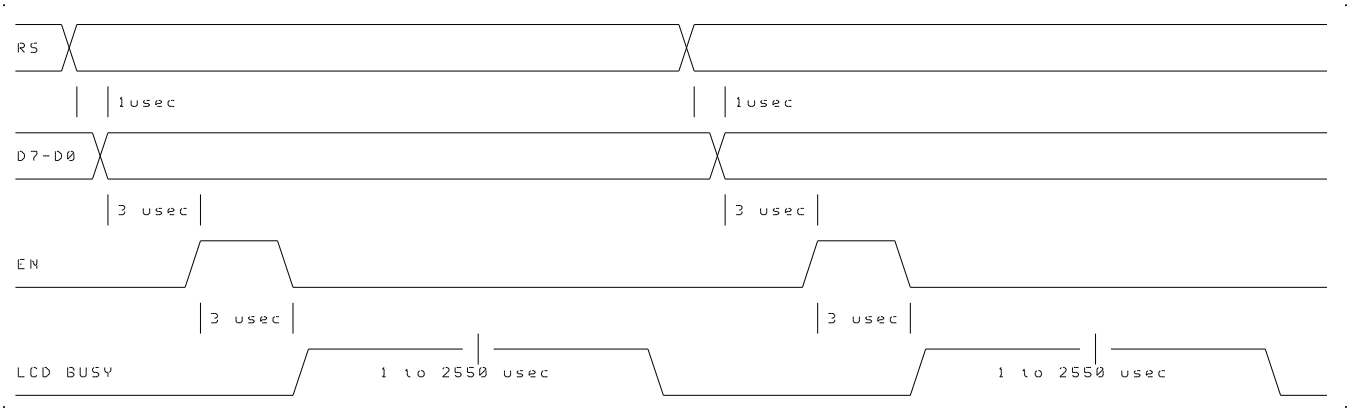


Figure 6. E3000/2 LCD Timing

All serial commands are written to a 128 byte circular buffer. After a command is processed, the next command is read from the buffer. In general the LCD display timing will settle to the incoming baud rate or the LCD timing, whichever is slower. At 19.2Kbaud the received byte rate is ~520usec, and at 115.2Kbaud the received byte rate is ~87usec. Unless delay instructions are used, the display rate is similar to the received baud rate and the 128 byte buffer is hardly used. Either a hardware or software reset will clear the buffer.

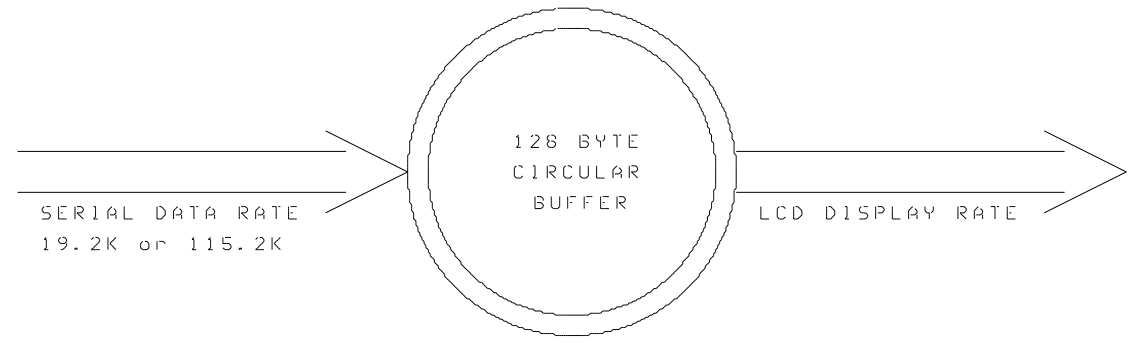


Figure 7. 128 Byte Circular Buffer

Module Settings

The E3000/2 module has a few user settings including LCD supply voltage, baud rate, contrast adjustment, and backlight resistor value.

3V/5V Selection
LCDs usually operate from either 3.3V or 5.0V. Solder a jumper to provide LCD power from either the VI pin or the onboard 3.3V regulator output. The regulator dropout voltage is 140mV at 300mA and only 50mV for standard LCDs operating at 15mA.

19.2K/115.2K Baud Rate Selection
Solder a jumper to select the 115.2K baud rate or leave it open for 19.2K baud. This pin has an internal weak pullup of ~200K to 3.3V. Ensure the module is reset after any baud rate changes.

Contrast Adjustment
A trimmer potentiometer used to adjust the LCD contrast usually just once. Send the LCD some data then adjust for best viewing.

Backlight Resistor
A user provided resistor bent to 0.4" lead spacing is needed to set the LCD backlight intensity. Consult the LCD datasheet to determine the backlight forward voltage Vf, the operating current If, and also whether it is already internally connected. Some LCDs have the backlight anode connected to the LCD power, others have the cathode connected to the LCD ground. The LCD backlight should not be run from the E3000/2 internal regulator.
The backlight resistor value is Rb=(VI-Vf)/If. For example for a 5V input, Vf=3.1V, If=16mA, Rb=(5.0-3.1)/0.016=118.8 ohms, use 120 ohms. Avoid connecting a backlight directly to any supply voltage unless there is a series resistance to limit the current. It is better to use a larger resistor to limit the current from 5V than to use a small resistor to limit the current from 3.3V as it allows better intensity control.

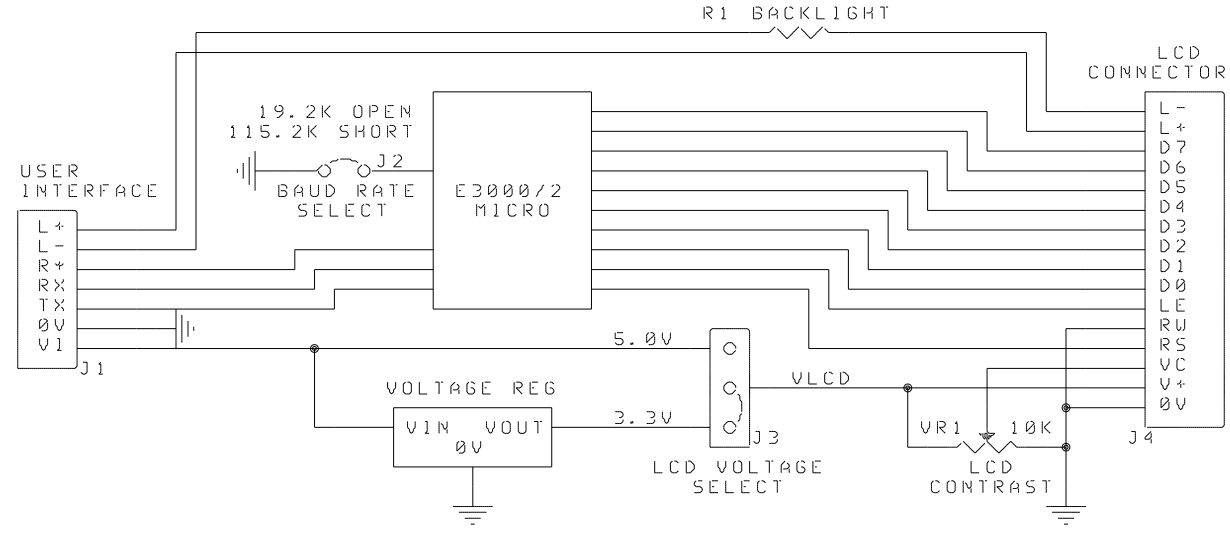


Figure 8. E3000 /2 Functional Diagram

E3000/2 Command Set

Most LCDs use an instruction set based on the HD44780 LCD controller or its many variants such as KS0066, NT7605, S6A0069, SPLC780, ST7066. To control an LCD, up to ten signals need to be controlled- eight Data, the Register Select, and the Enable for the fastest operation. To simplify operation, the LCD's Busy flag is not polled, instead a fixed delay is appended following each instruction.

The main problem in sending serial information is the eight bits per byte limitation vs the nine bits (eight data + register select) needed to control the display. There is thus the problem of differentiating between instructions and data. The E3000/2 solves this problem by taking advantage of the eight redundant custom character codes 0x08 to 0x0F which are used to identify and initiate a one or two byte instruction sequence. Using a one or two byte sequence for the instructions is a good compromise given that the majority of commands sent to the display are for data. When not in instruction mode, all bytes are interpreted as data by the E3000/2 except for bytes 0x08 to 0x0F.

Instructions can be either one or two bytes. Two byte instructions include a prefix instruction type followed by a value. Instruction values include those from the standard LCD set as well as some special ones. The only single byte instruction is 0x0F which is the module reset.

In summary, single bytes 0x00 to 0x07 display the custom character set, and bytes 0x10 to 0xFF display the regular character set. Bytes 0x08-0x0F are interpreted as prefix bytes for a one or two byte instruction sequence. This allows a simple interface with minimal changes to the user firmware.

The E3000/2 command set is as follows:

0x08 LCD Instruction Mode + Instruction (clear display, return home, entry mode, display on/off, cursor/display shift, function set, cursor position, + any custom instructions)

0x09 LCD Data Mode + Data 0x00 to 0xFF
0x0A Single Delay + Value 0x00 to 0xFF
0x0B Continuous Delay + Value 0x00 to 0xFF
0x0C Independent Delay + Value 0x00 to 0xFF
0x0D Reserved
0x0E Reserved
0x0F Module Reset

When a serial byte is received, if an instruction prefix byte 0x08 to 0x0F is detected, the E3000/2 processes it along with any following value bytes, sends it to the LCD, then exits the instruction process. This sequence is repeated for each instruction. Depending on the data or instruction, the Register Select line is set appropriately, and the Enable line is pulsed.

Standard LCD Instructions

The E3000/2 module emulates the standard LCD instructions as well as some special instructions. Commands are defined in this document as the bytes that are sent to the E3000/2 module while instructions and data are what are sent to the LCD. Of the commands, data is interpreted as 0x00-0x07 (custom characters) and 0x10-0xFF (standard characters), while instructions are triggered by the prefix bytes 0x08-0x0F followed by a single value byte.

The standard LCD instructions remain the same except for the 0x08 prefix command.

As each instruction is written to the LCD, there is an associated wait time allotted since the LCD Busy Flag is not checked. With the exception of the Clear Display and Return Home instructions which are allotted 2.7msec, all other instructions or data writes are allotted 80usec. These wait times can be changed with the Single Delay which runs only once for the following command, or continuously for the Continuous Delay. The E3000/2 module looks after the LCD execution timing, so the user need only send commands to the module.

On power up the E3000/2 module is reset and ready to accept commands. The LCD needs to be initialized however a word of caution- LCDs need their own reset period following power up during which commands are not accepted. This reset period can be up to 40msec, so waiting 50msec after power up is advisable before sending commands to the E3000/2. The LCD datasheet will indicate the recommended wait time.

Following the LCD power up wait period, the LCD needs to be initialized by instructions using the Instruction Mode 0x08 command. Consult the LCD datasheet to confirm or change the initialization values.

Following are the typical standard alphanumeric LCD instructions using the 0x08 prefix:

Function Set sets the data length 4/8 bits, number of display lines 1/2, and character font 5x8 or 5x10. For the E3000/2 module, the data length is 8 bits.
e.g. 0x08 + 0x38

Cursor/Display Shift sets the cursor shift and the display shift.
e.g. 0x08 + 0x10

Display On/Off sets the display on/off, the cursor on/off, and the cursor blinking on/off.
e.g. 0x08 + 0x0C

Entry Mode sets the cursor move direction left/right and the display shift on/off.
e.g. 0x08 + 0x06

Clear Display clears the display and returns the cursor to the home position.
e.g. 0x08 + 0x01

Return Home returns the cursor to the home position.
e.g. 0x08 + 0x02

Standard LCD Instructions (cont'd)

The Custom Character library can be programmed using the command 0x08 followed by the address 0x40 to 0x7F. Bits 0-2 of the address represent one of the eight rows while bits 3-5 represent one of the eight characters in the CGRAM. Following this the CGRAM data which is limited to 0x00-0x1F can be sent. Eight bytes are sent for each of the eight custom characters. Each byte represents one row in the 5x8 character bitmap. The CGRAM row address is incremented automatically with every CGRAM data write. Sending 0xF0 when the CGRAM Data is active returns the module to regular operation otherwise data will continue being written to the CGRAM instead of to the DDRAM. A typical sequence would be:

0x08 + CGRAM address 0x40-0x7F + (n) CGRAM data 0x00-0x1F + exit 0xF0

The cursor position can be placed using the command 0x08 followed by the address 0x80 to 0xFF. Following this, any data bytes will be written to the LCD starting at the cursor position. The LCD auto-increments the cursor so the user can send a character string once the initial start cursor position is sent.

0x08 + DDRAM address 0x80-0xFF + (n) DDRAM data 0x00-0x07 or 0x10-0xFF

Special Module Commands

The special commands include the Data Mode, the Delay functions, and a Module Reset.

Data Mode is a special command allowing the next byte 00-FF to be written as data to the LCD. This allows even 0x08 to 0x0F to be written as data. Following this byte write, the module returns to regular operation.

0x09 + Data 0x00 to 0xFF

The delay commands provide a set time period to be either included with the next instruction execution time or inserted prior to the next instruction. This is useful to accommodate any timing irregularities. There are two delay commands for instructions, one for a single event, the other for continuous use. There is also an independent delay command that delays the execution of the next data or instruction without affecting the LCD.

Since the module does not check the busy flag, it is prudent to use a 100% safety margin on the worst case LCD execution time, e.g. for a 37usec execution time use 80usec. The delay applies to the following instruction. The default execution time for all data or instructions is 80usec, however instructions such as Clear Display or Return Home which usually take ~1.5msec to complete have a default of 2.7msec and can only be changed by the Single Delay.

The Single Delay command 0x0A provides a one time delay in microseconds x10 for a range from 10usec to 2550usec. After the Single Delay command 0x0A is sent, the following byte specifies the delay value. For example if an additional delay of 2000 microseconds is needed for a special instruction, then 0x0A, 0xC8 (=200d) is sent prior to the command requiring the special delay.

0x0A + Single Delay Value 0x00 to 0xFF + command requiring single delay

Special Commands (con't)

The Continuous Delay command 0x0B allows a continuous delay in microseconds x10 to be applied to all future data or instructions. This allows the LCD execution time to be modified for a faster response or to accommodate slower displays. The continuous use delay is in multiples of 10 microseconds for a range from 10usec to 2550usec. A value of 5 would result in a delay of 50 microseconds being applied to all future command writes to the LCD. This allows other displays to be easily customized. A value of 0x00 reverts back to the default 80usec delay. The Clear Display 0x01, Return Home 0x02, and Function Set instructions are not affected by the continuous delay function.

0x0B + Continuous Delay Value 0x00 to 0xFF + commands requiring continuous delay

The Independent Delay command 0x0C allows a one time delay between 10usec to 2550usec to be inserted before the next data or instruction is processed. It is useful for increasing the execution time of the previous instruction or to handle any special timing. The LCD is not written to for this command.

0x0C + Independent Delay Value 0x00 to 0xFF

Commands 0x0D and 0X0E are reserved and will be ignored without any significant effect on the timing.

The Module Reset command 0x0F causes the module to be soft reset by clearing the buffer and resetting the delays to the default values. To ensure the CGRAM mode is completed, 0xF0 should be sent prior to the Reset 0x0F, otherwise the 0x0F will be written as CGRAM data if in that mode. The soft reset sequence for the module is thus:

0xF0, 0x0F

While the soft reset does not interfere with any LCD instruction timing as in a hard reset, it may interrupt any updates to the custom character library, so initializing the LCD afterwards always ensures operation from a known state.

Summary of "Standard" LCD Instructions vs E3000/2 Instructions

The E3000/2 commands allow the standard LCD instructions with only a 0x08 prefix byte while allowing a few special instructions to be added.

Table 3 Standard LCD Instructions vs E3000/2 Commands

Instruction	Standard (Binary)	Standard (Hex)	E3000/2 (Hex)
DDRAM Data	dddd dddd	0x00 to 0xFF	0x00 to 0x07 and 0x10 to 0xFF
DDRAM Address	1aaa aaaa	0x80 to 0xFF	0x08 + address 0x80 to 0xFF
CGRAM Data	000d dddd	0x00 to 0x1F	0x00 to 0x1F
CGRAM Address	01aa aaaa	0x40 to 0x7F	0x08 + address 0x40 to 0x7F
Clear Display	0000 0001	0x01	0x08 + clear display 0x01
Return Home	0000 0010	0x02	0x08 + return home 0x02
Entry Mode	0000 01 I/D S	0x04 to 0x07	0x08 + entry mode 0x04 to 0x07
Display On/Off	0000 1 D C B	0x08 to 0x0F	0x08 + display on/off 0x08 to 0x0F
Cursor/Disp. Shift	0001 S/C R/L 00	0x10 to 0x1C	0x08 + cursor/display shift 0x10 to 0x1C
Function Set	001 DL N F 00	0x20 to 0x3C	0x08 + function set 0x20 to 0x3C
Instr.Mode	N/A	Enter N/A	0x08 + instruction 0x00 to 0xFF
Enter Data Mode	N/A	N/A	0x09 + data 0x00 to 0xFF
Single Delay	N/A	N/A	0x0A + single delay 0x00 to 0xFF
Continuous Delay	N/A	N/A	0x0B + continuous delay 0x00 to 0xFF
Independent Delay	N/A	N/A	0x0C + independent delay 0x00 to 0xFF
Reserved	N/A	N/A	0x0D and 0x0E
Reset Module	N/A	N/A	0x0F
Exit CGRAM	N/A	N/A	0xF0 (in CGRAM mode only)

where I/D 1=increment, 0=decrement
S 1=display shift
D 1=display on, 0=display off
C 1=cursor on, 0=cursor off
B 1=cursor blink on, 0=cursor blink off
S/C 1=display move, 0=cursor move
R/L 1=shift right, 0=shift left
D/L 1=8 bits, 0=4 bits
N 1=2 lines, 0=1 line
F 1=5x10 dots, 0=5x8 dots

Examples

The following examples are typical for initializing, programming the custom character set, placing the cursor, and writing a message. The actual hex values are shown for completeness however your compiler will convert any text message into these bytes. The only caution is to be aware of any special instructions and make the necessary adjustments.

Initializing E3000/2 Module

The E3000/2 module may also be reset to the same state as a power-on or hardware reset with a reset sequence at any time. Using command 0xF0 exits from any active CGRAM mode or active two byte instruction, and the following reset command 0x0F resets the module. Due to the inherent delays of commands already being executed, by the time the reset command is run, any bytes sent immediately following the reset command will have been written into the buffer. When the reset command is run, the entire buffer will be cleared including any bytes sent immediately following the reset command. Therefore following a reset sequence, there should be a >10msec delay prior to any new commands being sent.

e.g. 0xF0, 0x0F + >10msec prior to next commmands

Initializing LCD

On power up the LCD needs to be initialized to clear the display memory, home the cursor, and set the operating mode. Allow at least 50msec after power up prior to sending commands to the E3000/2 module as the LCD itself may not yet be ready.

This example shows a typical initialization for left to right text entry. As the E3000/2 module writes to the LCD in 8 bit bytes, the D/L bit in the Function Set instruction needs to be set to "1" for 8 bit operation. The Function Set instruction 0x38 is repeated 3x during initialization. Note that the instruction mode prefix 0x08 is used for each LCD instruction.

;function set (0x08, 0x38), independent 2.55msec delay (0x0C, 0xFF), function set (0x08, 0x38), function set (0x08, 0x38), cursor/display shift (0x08, 0x10), display on (0x08, 0x0C), entry mode (0x08, 0x06), clear display and home cursor (0x08, 0x01)

e.g.0x08, 0x38, 0x0C, 0xFF, 0x08, 0x38, 0x08, 0x38, 0x08, 0x10, 0x08, 0x0C, 0x08, 0x06, 0x08, 0x01

On some very early HD44780A versions, it has been noted that an additional 1msec delay is needed following the Clear Display or Return Home instructions, otherwise the next few instructions will not execute properly. The later HD44780A and current HD44780U controllers work with no issues using the default 2.7msec execution times for these two instructions.

Programming a Custom Character

Custom characters are easily programmed by sending the Setup CGRAM Address command, then the CGRAM Address, then a byte for each of the eight rows for each custom character starting at the top. Only the least 5 bits of each of the eight bytes are used to create each row of a custom character. These eight bytes of five bits create the 5x8 custom character. Continue sending the next eight bytes for the 2nd custom character and so on as the CGRAM address pointer auto-increments. Send the 0xF0 character to exit the CGRAM mode.

Note- it's not possible to have more than 8 custom characters on the same screen

Setup CGRAM Address, CGRAM Address, +1 byte for each of 8 rows per character

The following example indicates the bytes sent to program eight bargraph characters:

```
;send CGRAM, CGRAM address
0x08, 0x40

;send custom character bit patterns in byte format
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F ;character 0, 1 bar
0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x1F ;character 1, 2 bars
0x00,0x00,0x00,0x00,0x00,0x1F,0x1F,0x1F ;character 2, 3 bars
0x00,0x00,0x00,0x00,0x1F,0x1F,0x1F,0x1F ;character 3, 4 bars
0x00,0x00,0x00,0x1F,0x1F,0x1F,0x1F,0x1F ;character 4, 5 bars
0x00,0x00,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F ;character 5, 6 bars
0x00,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F ;character 6, 7 bars
0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F ;character 7, 8 bars
0xF0 to exit CGRAM mode
```

Note that writing to the CGRAM loses the cursor position so the cursor needs to be placed before any writes to the DDRAM.

Placing the Cursor and Writing a Message on One Line

DDRAM addresses for each line vary according to the display format and the manufacturer. Consult the LCD datasheet for the correct values. Typical start addresses for a 4 line by 16 character alphanumeric display are

0x80 line 1, 0xC0 line 2, 0x90 line 3, 0xD0 line 4

The following example displays the classic "Hello world" at the beginning of line 1:

```
;place cursor for 1st position of line 1
0x08,0x80

;send "Hello world" as a text string or hex values representing the
ASCII characters

0x48,0x65,0x6C,0x6C,0x6F,0x20,0x77,0x6F,0x72,0x6C,0x64
```

Note that the LCD does not automatically continue the text onto the following line; the user needs to keep track of the end of line address and reposition the cursor accordingly.

Placing the Cursor and Writing a Message on Four Lines

In this example, the display will have a message indicated on four lines. While the LCD auto-increments the cursor position during each line, the user still needs to keep count of the cursor position as the LCD does not automatically move the cursor to the next line after completing the previous one. In fact, for the 4x16 LCD, the cursor will move to the beginning of the 3rd line after completing the 1st line. Thus the cursor needs to be repositioned after writing the 16th character on the 1st line to the beginning of the 2nd line which is at 0xC0. At the end of the 2nd line the cursor will be at 0xCF, and so needs to be repositioned to the start of line 3 which is 0x90. At the end of the 3rd line, the cursor will be at 0x9F, so again the cursor needs to be repositioned to the start of line 4 which is 0xD0. These hex values are obtained from the ASCII table for the HD44780 LCD controller datasheet. The lower character set is generally the same as that of other LCD controllers but the upper character set may change.

```
;place the cursor for 1st position of line 1
0x08,0x80

;send "Room Temperature" as a text string or hex values
0x52,0x6F,0x6F,0x6D,0x20,0x54,0x65,0x6D,0x70,0x65,0x72,0x61,0x74,0x75
,0x72,0x65

;place the cursor for 1st position of line 2
0x08,0xC0

;send "+24.3'C  +75.7'F" as a text string or hex values
0x2B,0x32,0x34,0x2E,0x33,0xDF,0x43,0x20,0x20,0x2B,0x37,0x35,0x2E,0x37
,0xDF,0x43

;place the cursor for 1st position of line 3
0x08,0x90

;send "Humidity     37%" as a text string or hex values
0x48,0x75,0x6D,0x69,0x64,0x69,0x74,0x79,0x20,0x20,0x20,0x20,0x20,0x33
,0x37,0x25

;place the cursor for 1st position of line 4
0x08,0xD0

;send "Time    15:37:02" as a text string or hex values
0x54,0x69,0x6D,0x65,0x20,0x20,0x20,0x20,0x31,0x35,0x3A,0x33,0x37,0x3A
,0x30,0x32

Notes- it's good practice to write blank spaces (0x20) in all unused
character positions to erase any remnants of previous messages.
```