Product Change Notification

PCN No: 2006-07-17-01
Report Date: July 17, 2006
Subject: LCD0821 PCB 2.0 New Board Revision Release

List of New Products:

LCD0821 PCB 2.0 including all applicable extensions and combinations

-WB, -GW, -IY, -R (display colour)
-VPT, -V, -E (power and temperature options)

Reason for Change:

The LCD0821 Matrix Orbital module series have been upgraded to offer new features and benefits; the physical connections on the PCB boards have been standardized and the components are now RoHS compliant.

Product Change:

The following list of hardware and firmware changes are all effective on all the variants of the LCD0821 as stated under 'List of New Products'. The old PCB 1.4 is now superseded by PCB 2.0, and the old firmware version v1.6 is now superseded by v5.6. The changes are stated to compare the old and the new PCB revision.

Hardware Changes:

1) Both old and new PCBs have 1 general purpose output (GPO). The difference is the type of GPOs. The old PCB has an active low GPO; the new one has an active high GPO. Active low GPO means that turning the GPO on (command 87) will make the GPO pin go low (0V). Active high GPO means that turning the GPO on will make the GPO pin go high

(5V).

2) RS232 speeds and I2C slave addresses are now configurable via software. As such, RS232 speed and I2C setting jumpers are not present in PCB 2.0. In place of the baud selection / I2C address jumpers, we now have a jumper for manual override.

3) Changing serial communication protocol is now more intuitive in PCB 2.0 by the use of the new protocol select jumpers.

4) The 2 middles pins of the power / data connector which were only for I2C in PCB rev 1.4 is now for TTL, RS232 and I2C in PCB 2.0. The 2 middles pins are selected via the serial communication select jumps.

5) With PCB 1.5, the module can not hold bidirectional RS232 communication; it is only able to do so in TTL. The new PCB is able to carry out bidirectional communication in both TTL and RS232.

Please refer to http://www.matrixorbital.ca/manuals/LCDVFD_series/LCD0821/LCD0821_rev_20.pdf for the LCD0821 PCB 2.0 manual.

Firmware Changes:

Release v5.4 (released March 20, 2006):

1) We now offer 16 bytes of EEPROM for the customers

- Customer data upload: Command: 254/52/<16 bytes>
- Customer data download: Command: 254/53

The display now has 16 bytes to store whatever the customer would like. These 16 bytes can be used for anything including serial numbers, addresses, or whatever you can think of that will fit in 16 bytes.

2) Added a watch dog timer: The units now make full use of a watchdog timer to prevent the display from remaining locked up. The default timeout is approximately 2.1 seconds (depending on ambient

temperature). Upon a watchdog reset the display will send 254/212 to the host to indicate that watchdog has reset the module.

3) The old PCB revision used to have an 80-byte buffer. We have increased the buffer size to 128 bytes in the new PCB rev.

4) I2C now properly ACKs and NAKs the master when it is addressed

I2C now sends a NAK when the module is busy or when the communication buffer is full. Since the module uses the same In and Out buffer with the RS232, there are commands necessary to be able to configure the buffer accordingly. These commands are:

- Command 161: Clear Out buffer – since I2C transactions are master-initiated, the replies of the slave module are placed in an Out buffer (16 byte buffer) until the master decides to read. If for some reason the master is out of sync with the data in the Out buffer because it did not clock the data out when it was supposed to, the master has the option of clearing the Out buffer but then it would have to re-send query commands and clock the data out from the module immediately after (provided the end of the command, so all the parameters should be clocked to the module first).

- Command 160: Set RS232 Tx Data <on/off> - just like how Out buffer is shared between I2C and RS232, so is the In buffer. Because of this there is a need to tell the module if data is coming from RS232 or from I2C. When this command is sent with parameter 0 (off), this hints that the data is from I2C, and that data responses in the Out buffer are checked against a flag that holds the value of the parameter for this command. When the parameter is 0 (off) the data remains in the Out buffer and NOT transmitted out via RS232. When the parameter is 1 (on – default case) the data responses are automatically transmitted via RS232. This command MUST be issued with parameter 0 when in I2C and expecting to read the module. The default for this flag is on, which means the module will send its reply or report via RS232.

- Command 163: Dump Out buffer via RS232 – this command is used to dump the Out buffer via the RS232. In the case that command 160 is sent with parameter = 0, the data responses are placed in the Out buffer and can be accessed via RS232 using this command.

5) The Init Narrow Vertical Bar command (115) now needs extra initialization before use. The custom characters that make up the narrow vertical bar needs to be sent via command Store Custom Chars (193) and loaded by Load Custom Chars (192)

- Command 193: Store Custom Characters: 254 / 192 / <bank> / <Id> / <8 char definition>

This command stores the custom chars in the EEPROM in the <bank> specified. Eg if the user specifies bank 1, he may be making custom chars for horizontal bars, and he may utilize these custom chars by sending commands 104 and 124 (Init Horizontal Bar and Place Horizontal Bar), but he also may make a totally different set of custom chars that he may load whenever he wants by using the following command. The definition for the old horizontal, vertical, medium numbers are all default in the module.

- Command 192: Load Custom Characters: 254 / 192 / <bank>

This command loads the custom chars defined in the <bank> specified. Again, for the example above, the user may send this command with bank 1 as parameter, and he may now use the characters just by issuing the Id (which is 0x00 to 0x07).

6) The display now has an enhanced data locking capability. The users can now lock the module in different levels.

- Command 202: Set Data Lock: 254 / 202 / 245 / 160 / <level>

- Command 203: Set and Save Data Lock: 254 / 203 / 245 / 160 / <level>

Level is an 8-bit value with the following bits defined:

bit 7 – when this bit is 1, users can not display anything on the module
bit 6 – when this bit is 1, users can not send any commands to the module but commands 202 / 203
bit 5 – reserved and should be left 0
bit 4 – when this bit is 1, users can not alter the module settings (internal EEPROM)
bit 3 – reserved and should be left 0
bit 2 – reserved and should be left 0

bit 1 – reserved and should be left 0
bit 0 – reserved and should be left 0
Eg: to lock the display (no text will be displayed) send command: 254 / <202 or 203> / 245 / 160 / 128
Eg: to lock the display propertied (no commands will be followed, with the exception of the Data lock command) send command: 254 / <202 or 203> / 245 / 160 / 192

7) List of New or Changed Commands:

Keypad Related Commands:

213: Assign Keypad Key ID

- new command, as described in item #1 under Firmware Changes

RS232 / I2C Related Commands:

57: Set Standard Baud - new command, as described in item # 4

164: Set Non Standard Baud - new command, as described in item # 5

160: Set RS232 Tx Data - new command, as described in item # 6

161: Clear Out buffer - new command, as described in item # 6

163: Dump Out buffer via RS232 - new command, described in item # 6

Special Characters Related Commands:

115: Init Narrow Vertical Bar - changed, as described in item # 7

124: Place Horizontal Bar - changed, as described in item # 7

192: Load Custom Characters - new command, as described in item # 7

193: Store Custom Characters - new command, as described in item # 7

Module Property Query Commands:

52: Write Customer Data - changed, as described in item # 2

53: Read Customer Data - changed, as described in item # 2

Display Screen Control Commands:

202: Set Data Lock - new command, as described in item # 8

203: Set and Save Data Lock - new command, as described in item # 8

Please see the manual for in depth details.

Release v5.5 (released May18, 2006):

1) RS232 speed (baud rate) is now software configurable.

- Command 254 / 57 / <parameter>

The following are the parameter values according to baud rate:

Parameter	***Baud Rate***
83	1200
41	2400
207	4800
103	9600
51	19200
34	28800
25	38400
16	57600

2) Baud rate can now be fine-tuned

- Command 254 / 160 / <speed low byte><speed high byte>

We now offer access to the baud rate generator registers in the uC so that users may be able to fine tune baud rates as necessary.

This command accepts 2 byte parameter which is derived from the equation:

speed = (8,000,000 / 8 x desired baud rate) – 1

Please see the manual for details on the use of this command. There are caveats as to how to use this command in the manual.

3) I2C address is now software configurable.

- Command 254 / 51 / <address>

<address> has to be an even number which is the same as the write address for the module. Read address is 1 up from this number. Eg if address is 0x52, read address is 0x53.

Release v5.6 (released July 11, 2006):

1) Under certain circumstances reading back of the I2C could be problematic. The old PCB 1.4 revision did not implement I2C data read before; it only had I2C keypad key reads. Reading back I2C data was implemented in firmware v5.4 but under circumstances, there was a flaw with this implementation. With firmware v 5.6, users are now able to read data other than keypad key presses via I2C.
2) Horizontal bar graph change. Only one directional horizontal bar graph can appear on the screen at a time. Either left facing or right facing horizontal graph can be used at a time. Trying out both directions within the same screen will garble the bars. If this is an issue with your design, please contact a Sales Rep for your options.
3) Firmware version 5.5 has the following control characters implemented:

   0x08 – Back Space

   0x0A – Line Feed

   0x0D – Carriage Return (not implemented in firmware version 1.6)

   0x0C – Clear Screen

   Firmware version 5.6 has the same control characters implemented with a difference in implementation of 0x0A. Firmware version 5.6 scrolls the display up when control character 0x0A is sent when the display is at the last row (row 2), and inserts a new line and puts the next insertion point on the start of the new line. With firmware version 5.5, when 0x0A is sent when the display is at the last row, the next insertion point goes back to home position; there is no scrolling of the text.

Contact Information:

Adam Lintott (Sales Manager): alintott@matrixorbital.ca

Reference Documents/Attachments:

Location: http://www.matrixorbital.ca/manuals/

Approvals:

Engineering: **James McTavish (Senior Engineer)**

Sales: **Adam Lintott (Sales Manager)**

Production: **Mike McTavish (Production Manager)**