

NAMB-6200B LCMP BOARD User Note

1 - Introduction

NAMB-6200B is to use “Parallel port” as the interface for both LCD display and keypad. A simple protocol is further defined so that applications can directly communicate with this module no matter what the operating system is.

There are only two connectors in this module, as shown in Fig 1; power connector and Parallel port connector.

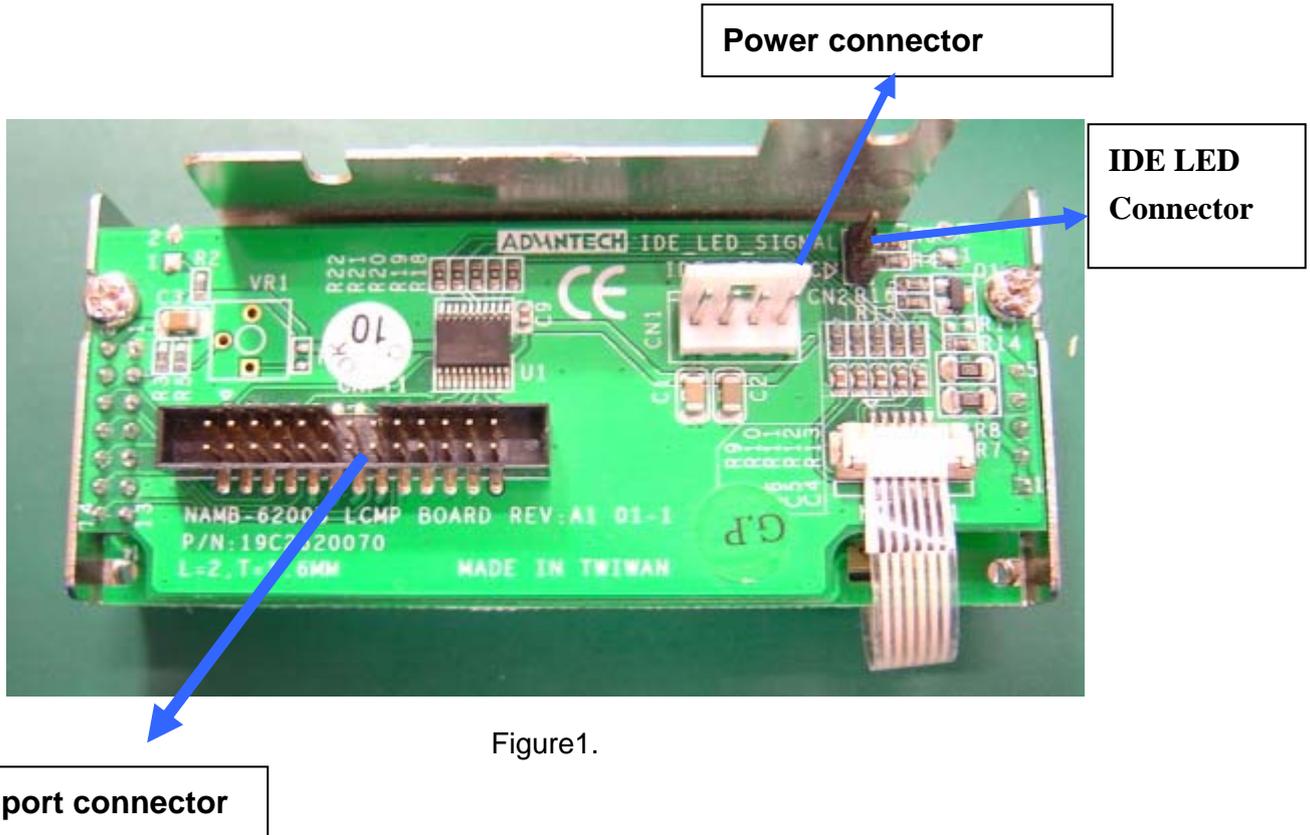
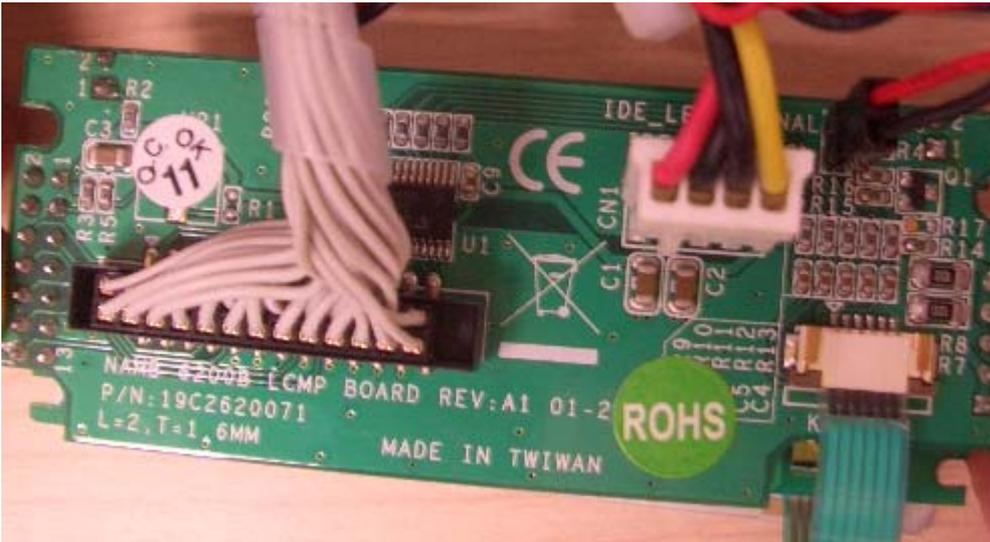


Figure1.

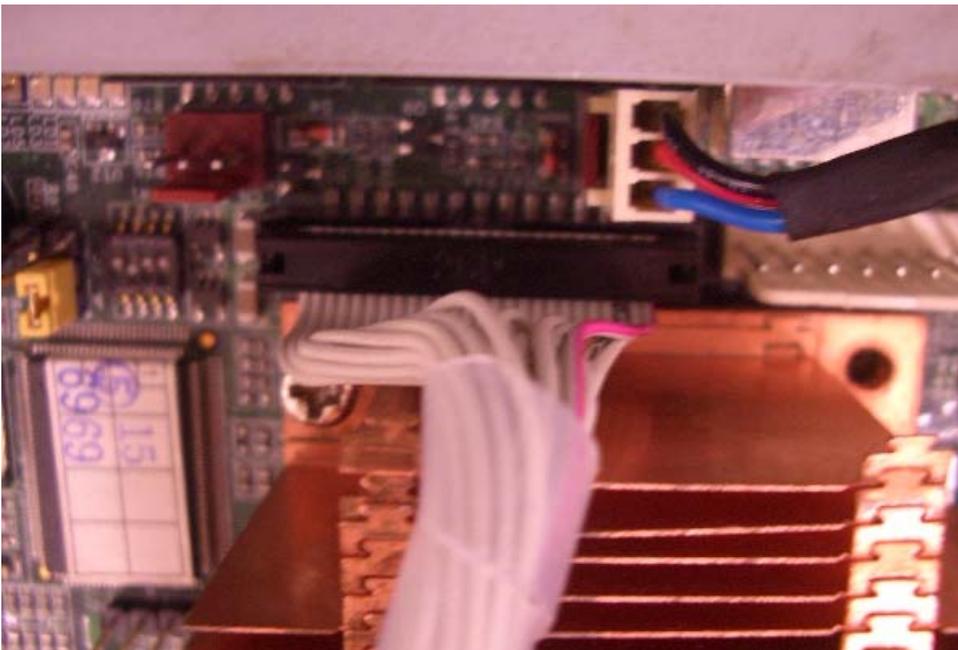
The power source into this module is 5volt only. There are only three pins being used in the Parallel port interface

2. Hardware Installation

1. Connect the Power/IDE_LED cable to connector the power/ IDE_LED connector of this module.



2. Connect the straight-through cable between parallel port of this module and computer.

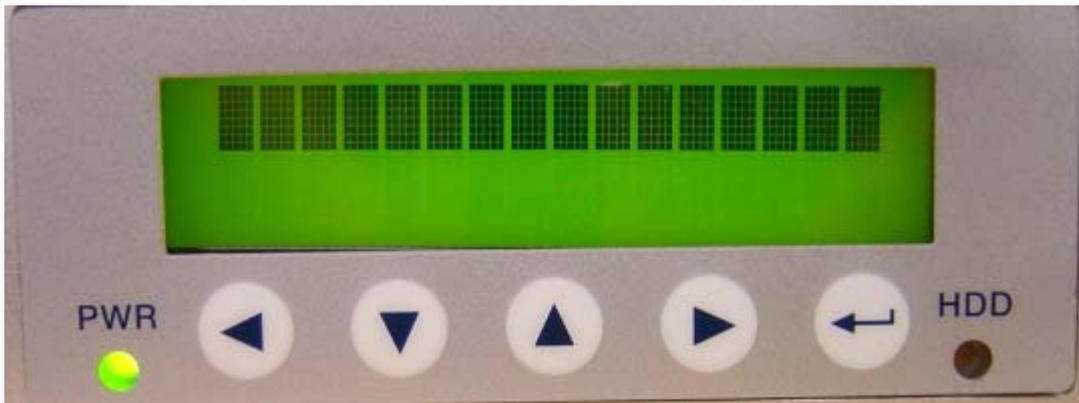


3. Demo Program

1. Kindly refer “**How to use LCM sample program .pdf**” to test NAMB-6200B LCM function.
2. Advantech LCM sample program only shows **ASCII 20H~ASCII 7FH** by NAMB-6200B LCM module.
3. If customer needs to show character of **ASCII 80H~ASCII FFH**, please refer LCM sample program to modified.

4. Boot-up message.

1. Kindly refer below picture; default LCM message has show squares.



2. If customer needs to show the peculiar message (ex: company name) during the system boot up, please contact Advantech sales to discuss message modification requirement.

5. NAMB-6200B LCMP Command

A. LCM Display message

NAMB-6200B is an intelligent device which will display those data received from Parallel port and reply key pressing status to polling command from Parallel port. There are command and data from Parallel port. To distinguish between data and commands, the LCD/key-pad Module recognizes a command prefix, 254 (Hex 0FE).

The byte following “254” will be processed as a command. For example, to clear the screen, send the command prefix (254) followed by the LCD clear-screen code (1). The valid data range is as following table shows.

Valid data range	Displayed characters
0 ~ 7	Customized icon 0~7
48 ~ 57 (30 ~ 39 Hex)	0 ~ 9
65 ~ 90 (41 ~ 5A Hex)	A ~ Z
97 ~ 122 (61 ~7A Hex)	A ~ z
Other ASCII characters	{ }, / + - []etc.

To get the key pressing status, a “read key” command can be issued to this module which will check the key-pressing status and reply accordingly. The following are the command and corresponding Decimal/Hex value:

Functions/commands	Decimal/Hex	Remark
Clear screen	1/01	
Home cursor	2/02	
Blank display (retaining data)	8/08	
Hide cursor & display blanked characters	12/0C	
Turn on (blinking block cursor)	13/0D	
Show underline cursor	14/0E	
Move cursor 1 character left	20/14	
Move cursor 1 character right	20/14	
Scroll 1 character left	24/18	
Scroll 1 character right	28/1C	
Set display address (position the cursor) location	128(Hex080) + Location	Note 1
Set character-generator address	64(Hex 040) + address	Note 2

B. LCM Key-pad

The LCD/key-pad module will check the status of every key and reply with status command accordingly. The replied message from LCD/key-pad module consists of a header and a status byte. The header byte is 253 (Hex0FD).

The high nibble (with the most significant bit) of the status byte is “4” or “5” and the low nibble (with the least significant bit) of the status byte is used to indicate key pressing status of the key-pad module. There are five keys in this module- upper arrow, down arrow, left arrow, right arrow, enter (ENT), The relationship between the function key, corresponding status bit, and status byte is as following table.

Function key	Corresponding status bit	Status byte
Up arrow	(0100 0001)	41 (H)
Down arrow	(0100 0010)	42 (H)
Left arrow	(0100 0100)	44 (H)
Right arrow	(0100 1000)	48 (H)
Enter	(0101 0000)	50 (H)

More than one key can be pressed at the same time so that there may be more than one “1”s in the low nibble of status byte.

For example, if Up and Down arrow keys are pressed at the same time while ” ready key” command being received, the replied status will be “Hex043”.

Note 1:

This command can be used to place the cursor at any location. The corresponding address for each character on the screen is as following:

For 16X2 Display Address

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Location	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

The address of characters at the same row are continuous, so moving cursor commands can be applied to shift the cursor position back and forth. However, the address of characters between upper and lower row are discontinuous. To change cursor position between upper row and lower row, this command will be applied.

Note 2:

This command can be used to create customized icon. The starting address is 64 and every character will take 8 bytes to create a 5 (width) x 7 (height) resolution picture, as shown in following:

CG RAM MAPPING

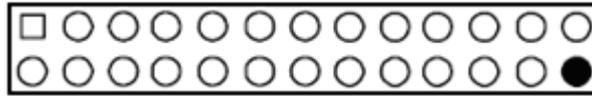
CG RAM Address					Character Pattern (CG RAM data)									
5	4	3	2	1	0	7	6	5	4	3	2	1	0	
High		Low				High		Low						
0	0	0	0	0	0	*	*	*	0	1	1	0	0	← Character Pattern
			0	0	1				1	0	0	1	0	
			0	1	0				0	0	1	0	0	
			0	1	1				0	1	0	0	0	
			1	0	0				1	1	1	1	0	
			1	0	1				0	0	0	0	0	
			1	1	0				0	0	0	0	0	
			1	1	1				0	0	0	0	0	
0	0	1	0	0	0	*	*	*	1	1	1	1	1	← Character Pattern
			0	0	1				1	0	0	0	1	
			0	1	0				1	0	1	0	1	
			0	1	1				1	0	1	1	1	
			1	0	0				1	0	1	0	1	
			1	0	1				1	0	0	0	1	
			1	1	0				1	1	1	1	1	
			1	1	1				0	0	0	0	0	
.		
.		
1	1	1	0	0	0	*	*	*	1	1	1	1	1	← Character Pattern
			0	0	1				1	0	0	0	1	
			0	1	0				1	1	1	0	1	
			0	1	1				1	0	0	0	1	
			1	0	0				1	0	1	1	1	
			1	0	1				1	0	0	0	1	
			1	1	0				1	1	1	1	1	
			1	1	1				0	0	0	0	0	

To show the customized icon, just send the data between "0" to "7" to this module. For example, this module will display the customized icon at location 64 to 71 upon receiving data "0"; it will display the customized icon at location 72 to 79 upon receiving data "1". There is a built-in watch dog timer in the module. This module will reset itself and send out "reset packet" (0FDH, 0EH) there after.

6 Parallel Port Connector

Parallel port connector

1 3



2

26

Pin	Signal	Pin	Signal
1	STB#	2	AFD#
3	PD0	4	Err#
5	PD1	6	PAR_INIT#
7	PD2	8	SLIN#
9	PD3	10	GND
11	PD4	12	GND
13	PD5	14	GND
15	PD6	16	GND
17	PD7	18	GND
19	ACK#	20	GND
21	BUSY	22	GND
23	PE	24	GND
25	SLCK	26	NC