Connecting RS232 board (STB0205)

Regular Voltage

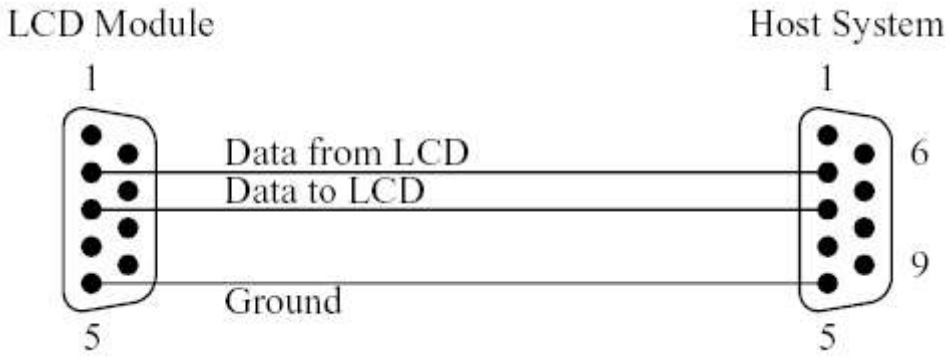
Power is applied to the white or brown four pin SIP connector (CN4) as follows:
Pin1: +5Vdc(4.75V ~ 5.25V)
Pin2: Unused
Pin3: Unused
Pin4: Gnd
If the sole data source is via the RS-232, the data input is via the DB9 connector. Pins2 and 3 are not used.

DB9 Connector Pin Out

RS-232 port: This connector is wired so that a standard "straight through" 9 pin D-sub cable may be used to connect the modules on a standard serial port such as comm ports on PCs. Note that this device complies to the EIA232 standard in that it uses signal levels from +/- 3V to +/- 12V. It will not operate correctly at TTL (0 to +5V) levels.

Pin Number	Description
2	Data Out
3	Data In
5	Ground

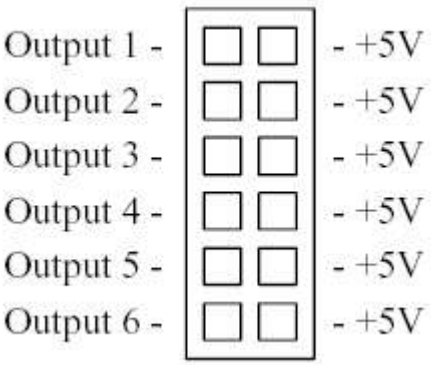
Note: Signals are shown as interpreted by the Liquid Crystal Display module.



General Purpose Output (GPO)

Each of the GPO's is meant to be used as a pair. The positive side of the GPO's is connected to a power source of +5Vdc supplied by the module at 8mA. The negative side of the GPO's is capable of finding a path to ground through a 600 ohm resistor. This resistor will limit the current flow through a GPO to approximately 8mA in the event of a short circuit. If the device which is being driven by a GPO requires a relatively high current (such as a relay) and has an internal resistance of it's own greater than 250 ohms, then the 600 ohm resistor may be shorted. You will find this resistor directly below the negative pin of the general purpose output.

Note: This operation requires soldering. The GPO's do not have any over current or over/under voltage protection so care must be taken if the user decides to connect the negative side differently. For instance, if the external device is a relay it must be fully clamped to absorb any generated back electromotive force (EMF).



Configuring STB0205

Default configuration is J1 and J2 installed. This configures the RS-232 to 19200 baud and depending on which connector is used. Please refer to the Appendix of this manual for programming notes.

RS-232 port: J1, J2, J3 – control baud rate

RS-232 format is 8N1 (8 characters, no parity, one stop bit)

Baud Rate	J3	J2	J1
1200	Off	Off	Off
2400	Off	Off	On
9600	Off	On	Off
19200	Off	On	On
	On	X	X

The Keypad Interface

Keypad Interface via RS-232

By default on any press of a key, the module will immediately send out the key code at the selected baud rate.

Auto Transmit:

If the auto transmit mode is on (default) then on any keypress, the module will immediately send out the key code at the selected baud rate. If auto transmit is off, when the poll command is sent, the module will immediately send the key code.

Keypad Interface (RS-232)

Auto Repeat:

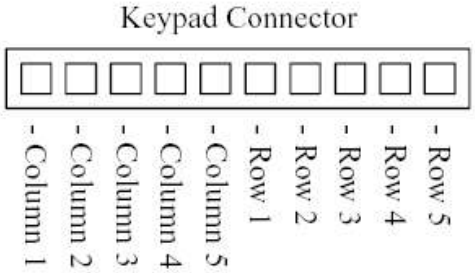
Two Modes of auto repeat are available and are set via the same command.
(1) Resend Key Code: This mode is similar to the action of a keyboard on a PC and is set via the "~" command with a value of0 as outlined in the Command Set section of this manual. In this mode, when a key is held down, the key code is trans-mitted immediately followed by a 1/2 second delay. After this delay, key codes will be sent via the RS-232 interface at a rate of about 5 codes per second.
(2) Key down / Key up codes: This mode may be used when the type matic parameters of the Resend Key code mode are unacceptable or if the unit being operated in polled mode. The host system detects the press of a key and simulates an auto repeat inside the host system until the key release is detected. This mode is entered via the "~" command with a data value of (0x01). In this mode, when a key is held down, the key code is transmitted immediately and no other codes will be sent until the key is released. On the release of the key, the key release code transmitted will be a value equal to the key down code plus 40 hex. The key code associated with key code "0" (0x30) is pressed, the release code is "p" (0x70). In RS-232 polled mode, theses codes are available in the same manner as keypresses without auto repeat ; however, the user should be careful to ensure that the poll rate is high enough so that simulated key repeats are avoided after the release of the key but before the next poll.

Physical Layout:

The returned key codes are as follows, but note that your keypad may be laid out in a different pattern. If this is the case, the user will need to interpret the key codes differently. The diagram 1 shows the logical layout (row 1, column 1 in upper left). The connector for the keypad is a 10 pin. 1" spacing male header. Pin 1 is indicated on the connector layout diagram. Pins 1 through 5 are columns and pins 6 through 10 are row. The keypad is scanned whenever a key is pressed. There is no continuous key scan. This means that key presses are dealt with immediately without any appreciable latency. This also prevents electrical noise which is often caused by continuous key scans.

Diagram A

		COLUMNS				
		1	2	3	4	5
ROWS	1	A	B	C	D	E
	2	F	G	H	I	J
	3	K	L	M	N	O
	4	P	Q	R	S	T
	5	U	V	W	X	Y



RS232 board Command Set (STB0205)

Write Text

This section of the manual allows the user to send commands to the STB0205 to alter the appearance or behavior of the display. To utilize a command, the user must send a command prefix followed by the command in the format described in this portion of the manual. Any characters sent without a command prefix will be interpreted as text and will be displayed on the module at the current cursor position. The current cursor position will be incremented for each character received.
Please note that unless line wrap is turned on, the text will follow the memory map of the module.
The command set discusses in detail what the STB0205's capabilities of and how to execute each command. The basic format of all the commands listed in this segment that do not require parameters is:
<command prefix> <command>
For commands that do require parameters however the format is as follows:
<command prefix> <command> [<parameter> <parameter> ……]

Basic Commands

All commands are prefixed by the command 0xFE, Hex FE, Decimal 254

Auto Repeat Mode On

ASCII "~", Hex 7E, Decimal 126
Syntax 0xFE 0x7E
To turn auto repeat on and select mode, send a command prefix followed by the character "~", followed by either 0x00 or 0x01:
= 0: 200ms typematic rate
= 1: Key down / Key up codes are sent
In either case, auto repeat is turned on.
Please see details in Keypad Interface section.

Auto Repeat Mode Off

ASCII "`", Hex 60 Decimal 96
Syntax 0xFE 0x60
To turn auto repeat off, send a command prefix followed by the character "`".

Auto Transmit Keypresses On

ASCII "A", Hex 41, Decimal 65
Syntax 0xFE 0x41
To activate the automatic transmission of keypresses, send a command prefix followed by the character "A". In this mode, all keypresses are sent immediately to the host system without the use of poll keypad command. This is the default mode on power up.

Auto Transmit Keypresses Off

ASCII "O", Hex 4F, Decimal 79
Syntax 0xFE 0x4F
To disable the automatic transmission of keypresses, send a command prefix followed by the character "O". In this mode, up to 10 keypresses are buffered until the unit is polled by the host system via the poll keypad command.

Poll Keypad

ASCII “&”, Hex 26, Decimal 38
Syntax 0xFE 0x26
To return any unbuffered keypresses via the RS-232 interface, send a command prefix followed by the character “&” and then set up the host system to receive the key codes. When a keypad module receives this command it will immediately return any unbuffered keypresses which may have not been read already. If there is more than one keypress buffered, then the high order bit of this returned keycode will be set. If this is the only buffered keypress, then the high order bit will be cleared. If there are no buffered keypresses, then the returned code will be 0x00.

Backlight On

ASCII “B”, Hex 42, Decimal 66
Syntax 0xFE 0x42 <number of minutes>
To turn the backlight on, send a command prefix followed by the character “B” and the number of minutes for the backlight to remain on. If <minutes> is sent as zero then the backlight will remain on indefinitely. The maximum value for <minutes> is 100.
Example:
<command prefix> 0xFE
<command> 0x42
<minutes> 0 to 100

Backlight Off

ASCII “F”, Hex 46, Decimal 70
Syntax 0xFE 0x46
To turn the backlight off, send a command prefix followed by the character “F”.

Clear Display

ASCII “X”, Hex 58, Decimal 88
Syntax 0xFE 0x58
This command clears any text and graphics off the display. To clear the display, send a command prefix followed the character “X”.

Clear Key Buffer

ASCII “E”, Hex 45, Decimal 69
Syntax 0xFE 0x45
This command clears any unread keypresses. In a menuing application, if the user presses a key which changes the menu context, any following key presses may be inaccurate and can be cleared out of the buffer between menu changes to prevent jumping around the menu tree. It may also be used to, in effect, reset the keypad in case the application resets for whatever reason. To execute this command, send a command prefix followed by the character “E”.

Cursor On

ASCII “J”, Hex 4A, Decimal 74
Syntax 0xFE 0x4A
To turn the cursor on at the current position, send a command prefix followed by the character “J”. Note cursor is on by default at power up.

Cursor Off
ASCII “K”, Hex 4B, Decimal 75
Syntax 0xFE 0x4B
To turn the cursor off at the current position, send a command prefix followed by the character “K”.

Cursor Left
ASCII “L”, Hex 4C, Decimal 76
Syntax 0xFE 0x4C
To move the cursor one space to the left of current position, send a command prefix followed by the character “L”.

Cursor Right
ASCII “M”, Hex 4D, Decimal 77
Syntax 0xFE x04D
To move the cursor one space to the right of the current position, send a command prefix followed by the character “M”.

Cursor Blink On
ASCII “S”, Hex 53, Decimal 83
Syntax 0xFE 0x53
To turn on the blinking cursor at the current position, send a command prefix followed by the character “S”. Please note that the blinking cursor is on by default at power up.

Cursor Blink Off
ASCII “T”, Hex 54, Decimal 84
Syntax 0xFE 0x54
To turn off the blinking cursor at the current position send a command prefix followed by the character “T”.

Set Debounce Time
ASCII “U”, Hex 55, Decimal 85
Syntax 0xFE 0x55 <number to define debounce time>
To set the time between key press and key read, send a command prefix followed by the character “U” and a number to define the debounce time. All key types with the exception of latched piezo switches will “bounce” for a varying time, depending on their physical characteristics. The default debounce time for the module is about 65mS, which is adequate for most membrane keypads. This time equates to a setting of 8 using this command as there is a debounce time resolution of 8.192 microseconds.

Create Custom Character
ASCII “N”, Hex 4E, Decimal 78
Syntax 0xFE 0x4E <character between 0x00 and 0x07> <8 bytes>
This command creates a custom character. For the STB0205 the user can have up to eight custom characters. To execute this command, send a command prefix followed by the character “N”. The display will now await the number which identifies the custom character. This number must be between 0x00 and 0x07. When the module determines what character it’s working on, then the user must send 8 bytes which define the display character. See the diagram below for and explanation of the display character structure.

Diagram A

MSB							LSB	
*	*	*	1	2	3	4	5	Data Byte1
*	*	*	6	7	8	9	10	Data Byte2
*	*	*	11	12	13	14	15	Data Byte3
*	*	*	16	17	18	19	20	Data Byte4
*	*	*	21	22	23	24	25	Data Byte5
*	*	*	26	27	28	29	30	Data Byte6
*	*	*	31	32	33	34	35	Data Byte7
*	*	*	36	37	38	39	40	Data Byte8

Pixel Layout of Display Characters

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
Cursor Line				

General Purpose Output On

ASCII “V”, Hex 56, Decimal 86
Syntax 0xFE 0x56 <hex number corresponding to output>
To turn a general output purpose on, send a command prefix followed by the character “V” and hex number which corresponds to the output which will be turned on. To turn on output five, the sequence would be:
<command prefix> 0xFE
<command> 0x56
<hex number corresponding to output> 05H

General Purpose Output Off

ASCII “W”, Hex 57, Decimal 87
Syntax 0xFE 0x57 <hex number corresponding to output>
To turn a general output purpose off, send a command prefix followed by the character “W” and a hex number which corresponds to the output which will be turned off.

Go To Position

ASCII “G”, Hex 47, Decimal 71
Syntax 0xFE 0x47 <column number> <row number>
To set the current cursor position, send a command prefix followed by the character “G” and two values defining the column and row of the desired cursor position.

Go To Position

ASCII “G”, Hex 47, Decimal 71
Syntax 0xFE 0x47 <column number> <row number>
To set the current cursor position, send a command prefix followed by the character “G” and two values defining the column and row of the desired cursor position.

Go To Top Left

ASCII “H”, Hex 48, Decimal 72
Syntax 0xFE 0x48
This command resets the current cursor position to the top left of the STB0205 screen. To execute this command send a command prefix followed by character “H”.

Initialize Large Digits

ASCII “n”, Hex 6E, Decimal 110
Syntax 0xFE 0x6E
Before any large digits may be created, this command must be executed. It only needs to be sent once to initialize the custom characters for large digits. Due to the fact large digits use pre-determined custom characters, no user custom characters may be displayed or created while digits are in use.

Place Large Digit

ASCII “#”, Hex 23, Decimal 35
Syntax 0xFE 0x35 <column number> <digit number>
This command allows the creation of large digits on the STB0205 screen. To execute this command the user must send a command prefix followed by the character “#” and the column number where the digit is to be placed. Then the user must enter the number of the large digit which is required in hexadecimal. Numbers of almost full display height may be placed along side regular text on four row displays. The column number has a maximum value which is less than the display width because the digits are all three columns wide. The module must be initialized for large digit creation before large digits may be placed. If regular text and large digits are mixed on one screen, the user should always set the display cursor position before placing regular text because the creation of a large digit will leave the cursor position to the bottom right of the large digit and not at the last regular text write position. The format for this command is as follows:
<command prefix> 0xFE
<command> 0x23
<column number> 0x01 to 0x12 (1 to 18 for a 20 column display)
<digit number> 0x00 to 0x09 (numbers 0 to 9)

Initialize Horizontal Bar Graph

ASCII “h”, Hex 68, Decimal 104
Syntax 0xFE 0x68
Before any horizontal bar graphs may be created, this command must be executed. It only needs to be sent once to initialize the custom characters for bar graph creation. Due to the fact bar graphs use custom characters, no user custom characters may be displayed or created while bar graphs are in use.

Make Horizontal Bar Graph

ASCII " ", Hex 7C, Decimal 124
Syntax 0xFE 0x7C <column number> <row number> <direction> <bar length>
This command places a horizontal bar graph at the specified column and row with the specified width. The format of the command is as follows:
<command prefix> 0xFE
<command> 0x7C
<column number> 0x01 to 0x14 for a twenty column display
<row number> 0x01 to 0x04 for a four line display
<direction> 0 for left to right, 1 for right to left
<bar length> 0x00 to 0x64 (0 to 100) for a twenty column display
Bar length is in pixel widths. On a twenty column display the maximum bar graph width is one hundred (if the bar graph starts at the edge). This is due to the fact that each of the twenty columns on the display are five pixels wide. The width of the space between columns is not taken into account.

Initialize Thick Vertical Bar Graph

ASCII "v", Hex 76, Decimal 118
Syntax 0xFE 0x76
Before any thick vertical bar graphs may be created, this command must be executed. It only needs to be sent once to initialize the custom characters required for bar graph creation. Bar graphs use custom characters, therefore no user custom characters may be displayed or created while bar graphs are in use.

Initialize Thin Bar Vertical Graph

ASCII "s", Hex 73, Decimal 115
Syntax 0xFE 0x73
This command is excuted in exactly the same manner as the "Initialize Thick Bar Graph" command.

Make Vertical Bar Graph

ASCII "=", Hex 3D, Decimal 61
Syntax 0xFE 0x3D <column number> <bar length>
This command places a bar graph at the specified column with the specified height. The style of the vertical bar graph whether it be thick or thin is selected by the initialize vertical bar graph command. If thick bar graphs are preferred, the "v" command should be send to initialize the bar graph. If a thin vertical bar graph is desired, the "s" command should be sent to initialize the bar graph. The format for the command is as follows:
<command prefix> 0xFE
<command> 0x3D
<column number> 0x01 to 0x14 for an twenty column display
<bar length> 0x00 to 0x20 (1 to 32) for a four line display.
Bar height is in pixel widths. On a four line display the maximum bar graph height is thirty-two, this is because each of the lines in the display are eight pixels high. The width of the space between the lines is not taken into account.
Note: Large Digits may not be used with bar graphs. Vertical bar graphs may not be used with horizontal bar graphs, and thick bar graphs may not be used with thin bar graphs. This is because all these functions make use of the same "custom character" spaces. As a result, custom characters are also unavailable while any of these functions are in use.