

## SerialKey PC Serial Keyboard Interface for Embedded Systems

### General Description

**SerialKey** is an integrated circuit that translates the multi-byte keycodes of a standard PC keyboard into simple serial characters which even non-PC based systems can understand. It alleviates a host system from having to respond to the keyboard in real-time, and makes the keyboard look like a simple RS-232/422 serial terminal which any embedded system can communicate with through a serial port at most standard baud rates.

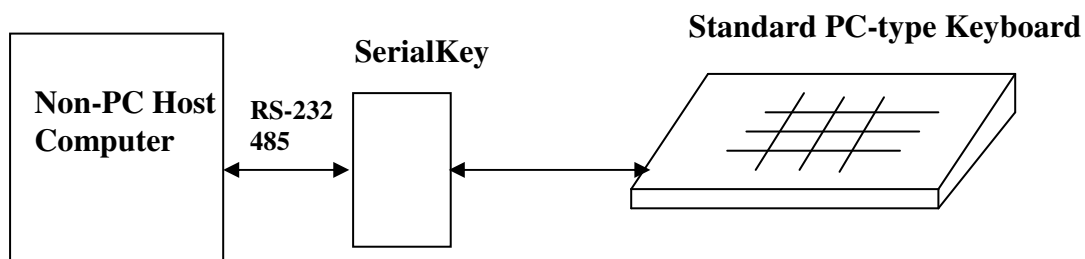
SerialKey also handles the three standard keyboard LED's transparently, freeing the host from this task. Keys which affect the state of the LED's are detected by SerialKey, and the LED's are turned on/off automatically. Upper/lower case translations as well as NUMLOCK features are also handled by SerialKey so your application does not have to manage the state of the keyboard. The ASCII characters generated are pre-conditioned by the state of the keyboard CTRL/ALT/SHIFT etc. keys.

By handling the complicated two-way communication protocol of the keyboard, managing the LED's, making the translation to simple ASCII characters, and allowing for 'hot-plugging' the keyboard, SerialKey provides the flexibility to allow PC-type keyboards to be used with any embedded computer system. This includes even those systems not based on the Intel architecture, and those 4/8-bit systems which do not have the resources to deal with a keyboard's real-time communications requirements.

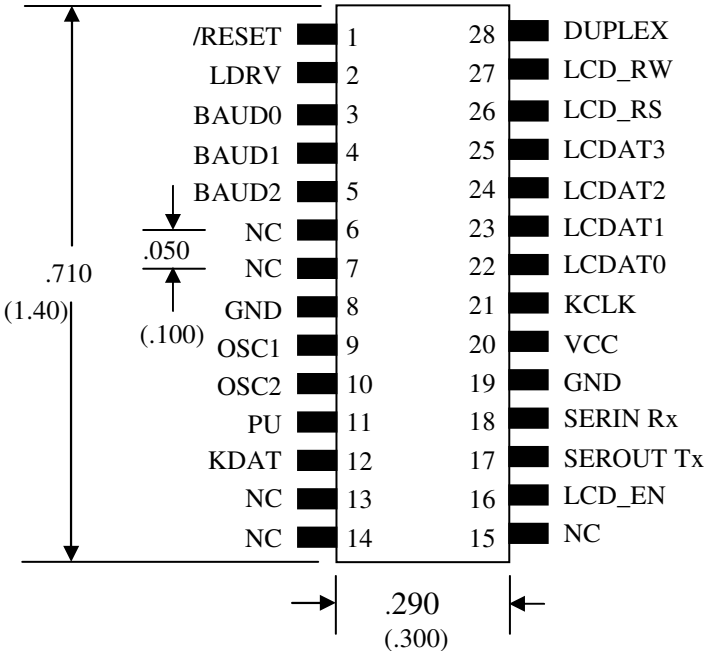
And SerialKey can drive a standard LCD 4-Line X 20 Character display so you can see characters being typed, as well as characters received from the host. So the addition of this display provides a full text-terminal capability.

<b><u>FEATURES</u></b>	<b><u>APPLICATIONS</u></b>
<ul style="list-style-type: none"> <li>• Converts Keyboard Codes to Simple ASCII</li> <li>• Non-ASCII Keys Return 8-bit Characters</li> <li>• Keyboard LED's Fully Operational With No Host Intervention</li> <li>• Optional LCD display shows characters typed, as well as returned from host</li> <li>• ALT/CTRL Keys Easily Detected By Host</li> <li>• RS-232/485 To Host @Selectable Baud</li> <li>• Standard 28-pin DIP or SOIC Packages</li> <li>• 5 Volt Operation - Keyboard Can Be 'Hot-Pluggable'</li> </ul>	<ul style="list-style-type: none"> <li>• Embedded systems requiring full or part time keyboard attachment</li> <li>• Full size keyboard as override of miniature or limited-key keypads</li> </ul>

### TYPICAL APPLICATION



**PIN CONFIGURATION – TOP VIEW**



**28-PIN DIP or SOIC**  
**Package dimensions shown for SOIC, (DIP)**

**ABSOLUTE MAXIMUM RATINGS**

Ambient Operating Temperature	-55 to +125°C
Storage Temperature	-65 to +150°C
Voltage on any pin w/respect to GND	-0.3V to VCC + 0.3V
VCC Voltage	-0.3V to + 7.5V
Operating Current	35mA (excluding keyboard)

## PIN DESCRIPTION

PIN		TYPE	NAME	DESCRIPTION
1		In	/RESET	Reset, active low, 1 $\mu$ s minimum width
2		Out	LDRV	Flashes an LED @ 1/2 second when operating
3		In	BAUD0	Baud Rate Control bit
4		In	BAUD1	Baud Rate Control bit
5		In	BAUD2	Baud Rate Control bit
6		-	NC	No Connection
7		-	NC	No Connection
8		Pwr	GND	Ground
9		In	OSC1	XTAL or external 20.00 MHz CMOS OSC
10		In	OSC2	XTAL or do not connect if external OSC is used
11		In	PU	Test Input - connect to VCC w/pullup resistor
12		OD	KDAT	Connect to keyboard Data Signal
13		-	NC	No Connection
14		-	NC	No Connection
15		-	NC	No Connection
16		Out	LCD_EN	Control signal for optional LCD DISPLAY
17		Out	SEROUT	CMOS level serial Tx output
18		In	SERIN	CMOS level serial Rx input
19		Pwr	GND	Ground pin
20		Pwr	VCC	+5VDC power input
21		OD	KCLK	Connect to keyboard Clock Signal
22-25		Out	LCD0-3	LCD DISPLAY Data Bits 0-3
26		Out	LCD_RS	LCD Control Signal
27		Out	LCS_RW	LCD Control Signal
28		In	DUPLEX	0 = Full Duplex, 1 = Half Duplex

\*OD denotes open drain signals

## DC Characteristics

DC Characteristics		Standard Operating Conditions: Operating Temperature 0°C to + 70°C Operating Voltage 4.5V ≤ VCC ≤ 5.5V			
Parameter	Min	Typ	Max	Unit	Conditions
Input Low Voltage KDAT, KCLK, SERIN, PU, BAUD0, BAUD1, DUPLEX /RESET OSC1	Gnd Gnd Gnd	- - -	0.15VCC 0.2VCC 0.3VCC	V V V	
Input High Voltage KDAT, KCLK, SERIN, PU, BAUD0, BAUD1, DUPLEX /RESET OSC1	0.8VCC 0.8VCC 0.7VCC	- - -	VCC VCC VCC	V V V	Gnd ≤ Input ≤ VCC
Analog Signals AN0-AN3 VREF	0.0 0.0	- -	VCC VCC	V V	
Input Leakage Current	-	-	±5	µA	Gnd ≤ Input ≤ VCC
Output Low Voltage OSC2 OTHER OUTPUTS	- -	- -	0.6 0.6	V V	IOL = 1.8mA, VCC = 4.5V IOL = 8.8mA, VCC = 4.5V
Output High Voltage OSC2 OTHER OUTPUTS	VCC-0.7 VCC-0.7	- -	- -	V V	IOH = -1.3mA, VCC = 4.5V IOH = -2.5mA, VCC = 4.5V
Brown-out Reset Voltage	3.7	4.0	4.4	V	
VCC Rise to Ensure Reliable Power-On Reset	0.05	-	-	V/ms	

For more detailed DC and AC specifications refer to the Microchip PIC16F73 datasheet.

## SERIAL COMMUNICATIONS RATE

The serial baud rate at which the SerialKey communicates with the host is programmable according to the following table. Note that the controlling bits BAUD0, BAUD1, BAUD2 are read only during power-up or reset. Therefore if the bits are changed during operation, the baud rate does not change until the next power-up/reset cycle. The communications protocol includes 8 data bits, 1 stop bit, and no parity.

**BAUD RATE SELECTION TABLE**

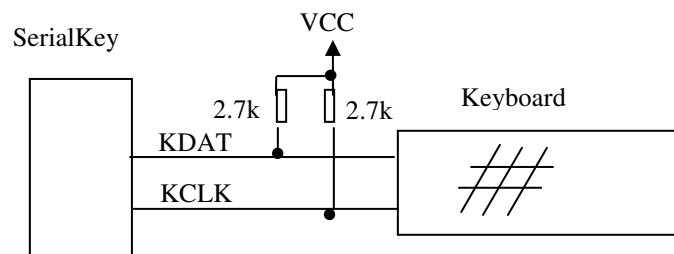
BAUD2	BAUD1	BAUD0	Serial Baud Rate
0	0	0	2400
0	0	1	2400
0	1	0	4800
0	1	1	9600
1	0	0	19.2k
1	0	1	38.4k
1	1	0	57.6k
1	1	1	115.2k

## LED DRIVER

SerialKey includes an LDRV pin which may be used to flash an LED as a visual indicator that it is powered-up and operating. If this function is not needed, leave LDRV unconnected. LDRV will flash an LED approximately twice per second.

## IBM PC KEYBOARD INTERCONNECT

The main function of SerialKey is to communicate with the PC-type keyboard. The two pins which accomplish this task are KDAT and KCLK which are open-drain outputs. KDAT must be connected to the keyboard's DATA signal, and KCLK must be connected to the keyboard's CLOCK signal. Pull-up resistors must also be added to these signal lines due to the open-drain pin configuration which allows for bi-directional communication. The recommended value for the pull-up resistors is 2.7k ohms.



## OSCILLATOR

SerialKey requires a 20.0 MHz ( $\pm 0.1\%$ ) clock source for operation. This may be generated by a dedicated crystal as shown in the typical application circuit below, or may be provided by an external oscillator. If an external oscillator is used, the 20MHz clock signal must be connected to OSC1 with OSC2 left unconnected. The signal applied to OSC1 must present CMOS compatible levels as indicated in the DC Characteristics Table above.

## OPTIONAL LCD DRIVER

SerialKey can also drive an optional 4-line by 20 character LCD display. With the display connected as shown below, the display operates in full or half-duplex mode depending on the state of the DUPLEX input pin. In full duplex, SerialKey displays only (printable) ASCII characters which are received from the host on all 4 LCD display lines. Nothing that is typed on the attached keyboard is displayed on the LCD. In half-duplex, the two lower lines of the display show the (printable) characters that are typed on the local PC keyboard and sent to the host, while the top two lines display the characters that are received from the host. In all cases, carriage returns (CR) are treated as CR/LF, so that a CR causes the cursor in the associated section of the display to move to the start of the next line. If the cursor was at the bottom of the display section, the display section scrolls up. SerialKey is compatible with the use of the Samsung UC-20401 four line by 20 character LCD display module, and is generally compatible with displays which use the Hitachi 44780, or Samsung KS0066U LCD drivers.

If the LCD display is not used, all associated signal pins (LCDAT0-3, LCD\_EN, LCD\_RS, and LCD\_RW) should be left unconnected, and DUPLEX should be grounded.

## OPERATING POWER

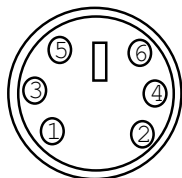
SerialKey detects the presence of the keyboard when it powers up. It waits up to 2.5 seconds for the keyboard to respond with its 'successful power-up code'. If it does not get this automatic keyboard power-up response it reports that the keyboard is not present. If it receives the response, it reports that the keyboard is connected. If the supply voltage to SerialKey drops to the Brown-out voltage level, SerialKey stops operating and enters a reset state. SerialKey will resume operation from a reset state when the power again reaches a valid operational level. The /RESET pin may be connected directly to VCC if an external reset function/button is not needed. SerialKey contains an internal power-up reset which will reliably reset the device on power-up as long as the power supply rises at a rate of 0.05V/ms or greater.

For added reliability SerialKey implements a watchdog timer which times out if abnormal program activity persists for longer than 100ms. The timeout event causes a SerialKey reset which will re-start operation as though from a power-on reset. Operation continues with no host intervention after a short initialization period.

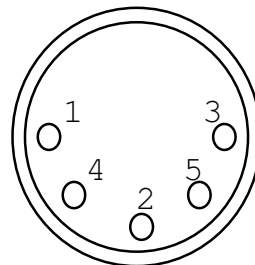
Go here to see a typical application schematic using SerialKey

<http://www.versalent.biz/manuals/webschematic1.pdf>

### **TYPICAL APPLICATION**



PS2 KEYBOARD  
CONNECTOR



5-PIN DIN KEYBOARD  
CONNECTOR

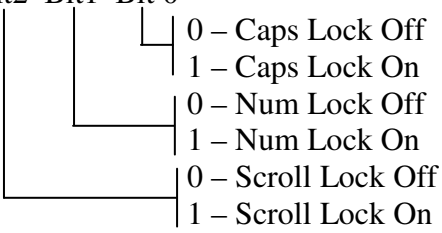
### **COMMON PC KEYBOARD CONNECTORS**

## SerialKey Serial Command Set

The serial command set is a subset of the ASCII control-characters which a host can send SerialKey to elicit a response. Typically these commands are used to determine the state of one of the keyboard LEDs, or to force them to particular states. The main function of SerialKey is to communicate with the keyboard and simplify the keyboard's keycodes for the host, so it responds to very few commands. The host must send these special commands as specified, or SerialKey will ignore the command. SerialKey always returns an 8-bit command response character (E4H) prior to any command response so that the host can isolate a command response from a keypress character. The character immediately following this isolation character is the command response if one is required. Commands consist of single ASCII characters as shown in the table below.

The host must wait for the response from each command before sending the next command. The response to a command may be the very next character(s) that SerialKey returns, or the response may be embedded within a series of keypress characters if the keyboard is being used. The host must identify the command response character within any series of characters, and extract the next character (if required) as the response to its command. Commands are executed within 2-4ms, which is at most a few serial-character times, so the response can be expected to be within the next 1-8 characters. If SerialKey receives anything other than the command characters listed below, it returns the single character E9H instead of an E4H to indicate that a command error has occurred.

### Serial Command Table

ASCII Command	HEX Value	Command Description	Command Response
DC4	14	Report Keyboard Status	Two 8-bit characters. E4H followed by a character representing the keyboard status: (Bits 7-3 undefined) Bit2 Bit1 Bit 0  <ul style="list-style-type: none"> <li>0 – Caps Lock Off</li> <li>1 – Caps Lock On</li> <li>0 – Num Lock Off</li> <li>1 – Num Lock On</li> <li>0 – Scroll Lock Off</li> <li>1 – Scroll Lock On</li> </ul>
NAK	15	Force Caps Lock ON	E4H
SYN	16	Force Caps Lock	E4H
ETB	17	Force Num Lock ON	E4H
CAN	18	Force Num Lock OFF	E4H
EM	19	Force Scroll Lock ON	E4H
SUB	1A	Force Scroll Lock OFF	E4H

The following two pages present a diagram of a 102-key PC keyboard with the keys arranged in their relative positions. Above each key is the key legend which appears on the actual key, and within the key outline are the hexadecimal (8-bit) character values which SerialKey transmits when these keys are pressed. Note that the topmost value is transmitted when no other special keys are in effect. The second code is the value sent when a SHIFT key, or the CAPS LOCK state is active, the third value is sent when the ALT key is active, the 4<sup>th</sup> value is sent when a CTRL key is active, and the last code is the value sent when the NUM LOCK state is active.

Note that the ALT key modifies most values sent by preceding them with the 8-bit character EEH. Refer to the following pages which present the PC keyboard pictorial with the associated serial codes that are generated for each key,

Click here to see a diagram of a 102-key PC keyboard with the associated characters and keycodes which SerialKey generates for each.

<http://www.versalent.biz/manuals/STAmankeymap.pdf>

## Revision History

Version	Released	Description
V1.01	8/16/09	Correct PS2 Male keyboard connector pin out
V1.02	10/29/09	Correct error in pin-description table, make PIN 13 No Connect and add reference to Hitachi and Samsung LCD driver types.