RADOVAN

EASY INPUT V2.4
KEYBOARD EMULATOR IC

SPECIFICATION AND
APPLICATION NOTES

PRELIMINARY
(All parameters subject to modification)

Version 11/13/05

INTRODUCTION

The EASY INPUT chip is designed to allow the transfer of ASCII data into keyboard based PC programs such as word processors and spread sheets directly from external hardware devices. It does this by effectively emulating a USB keyboard and using generic default drivers. Therefore, end users can use the USB keyboard drivers, which come with their operating system making the development of custom drivers unnecessary. This chip is designed to be embedded into external hardware devices and it can accept data input through parallel or synchronous serial connections.

ELECTRICAL CHARACTERISTICS

Ambient temperature with power applied: 0 to 70 deg C
Supply voltage: 4.2 to 5.5vdc
Supply current: 20mA Max
Input low voltage: 25% of Vcc Max
Input high voltage: 65% of Vcc Min
Output data line (Busy): Low = 0.4vdc Max @ 2.0mA
Output data line (Busy): High = Vcc - 2vdc Min @ 2.0mA

DATA TRANSFER RATE

Since the EASY INPUT chip is effectively a USB keyboard emulator, it must abide by the USB keyboard device specification to achieve system compatibility. Each keyboard action must be sent on a 10ms interval. An action in this case is any keyboard action such as a key press code, control shift, alt, etc.

Key release codes may be sent, however, it is not necessary because each key code sent implies a release of the previous key. If a new key code is not pending after the last character sent, the EASY INPUT chip will send a release code at the next 10ms interval. For example: "<shift> EZ" will be sent as <shift>, E, <shift>, Z. The EASY INPUT chip will then send the release code for the "Z" for a total time of 50ms.

COMPATABILITY

PC: Win 98 or higher (It is not compatible with WIN 95).
Macintosh: OS 9, OS X

TYPICAL APPROACH (DEVELOPER)

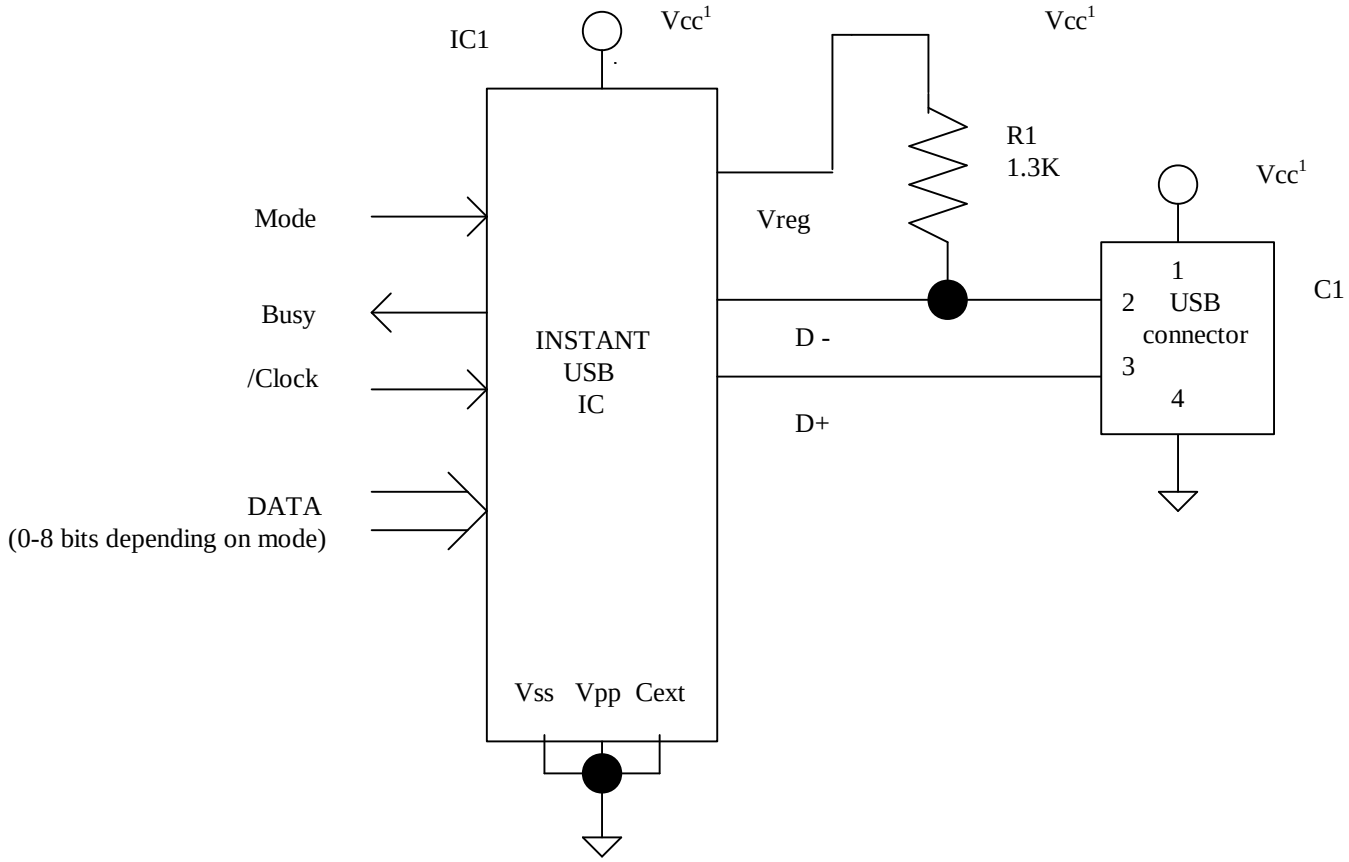
1) Determine which application software (spreadsheet, word processor, etc.) is best for use with your products output.
2) List other keystrokes, which are required to use within the application software. For example, moving between cells or entering cell formulas in spreadsheets or formatting and headers required for reports, etc.
3) Confirm that all required actions can be achieved with keystrokes supported by the EASY INPUT chip.
4) Build a prototype PCB, which will allow you to test the EASY INPUT, chip in your product.
5) Test and evaluate the prototype PCB to be sure the EASY INPUT chip will support you're application in every scenario as required by your business plan.
6) Embed the EASY INPUT chip into your products circuit board based on the required protocol

TYPICAL APPROACH (END USER)

First usage:
1) On the PC, close any applications that are open.
2) Plug the USB connector into the PC's USB port
3) If prompted for a driver, place that system's WIN 98 (or greater) disk in the disk drive and indicate that the system should look on that disk for the driver. Install the driver normally.
4) Open a text, spreadsheet, or other keyboard based document as required.
5) Be sure the cursor is in the correct start position as required.
6) Start the data transfer on the external device (by pressing a button, etc.).
7) Allow the data transfer to continue until it is complete.
8) Use the data as required.

Subsequent usage:
1) Open a text, spreadsheet, or other keyboard based document as required.
2) Plug the USB connector into the PC's USB port.
3) Be sure the cursor is in the correct start position as required.
4) Start the data transfer on the external device (by pressing a button, etc.).
5) Allow the data transfer to continue until it is complete.

6) Use the data as required.



Parts list

Reference	Description	Manufacturer	MFG Part Number
IC1	USB Keyboard emulator IC	Radovan Robotics	EASY INPUT IC
C1	USB Series "A" connector with attached cable (as a permanent cable)	Molex, etc.	Describe to vendor
	Series "B" receptacle (for use with a detachable cable)	Molex, etc.	Describe to vendor
R1	Resistor, carbon, 1.3K, 1/4W, 5%	Generic	Generic

1) IMPORTANT: Please note that Vcc is provided by the host via connector C1 for the circuitry above. Any additional circuitry MUST have a separate power source, which is NOT CONNECTED to the Vcc used by the EASY INPUT chip or the circuitry shown above.

Mode selection

Mode 0 Input	Mode 1 Input	Mode
0	0	USB Test
1	0	Parallel
0	1	Serial
1	1	Reserved

FIRST TIME USAGE IN ANY MODE

Normally, first time usage will operate as listed here:

1) On the PC, close any applications that are open.
2) Open the application that will be used.
3) Place the cursor where the data entry is to start
4) Plug the USB connector into the PC's USB port
5) If prompted for a driver, tell the system to look for the best driver. Normally, it will find it without the need to load a disk. In some cases, it may be necessary to place that system's WIN 98 (or greater) disk in the disk drive and indicate that the system should look on that disk for the driver. Install the driver normally.
6) The product can now be used as described below.

However, the developer must be aware of the "first use" time delay. The first use time delay is the time required for the user to go through about 5 quick windows to tell the system to find a driver within windows' pre-existing set of available drivers. In human terms, this is a very quick and easy procedure. However, during this time, the host system indicates that it is accepting characters. Since the windows which load the drivers do not generally use those characters, all or part of the data will often be lost. This problem occurs only on the first use of Easy Input on any host system.

Therefore, it is recommend that developers establish a procedure to make the data loss innocuous. For example, instruct the end user that data will be lost during this initial set up period on any "new" host system. That way, they can be sure to loose only test data and not something valuable to them. Also, it is possible to use a download button on your external equipment such that data is not sent to the P.C. until the download button is pressed. Then, the end user can be instructed to allow the host system to install its drivers before the download button is pressed.

It may be helpful to test the first use scenario on any given P.C. This can be done by removing the driver so that it can be reloaded to test the first use. To remove the Easy Input driver, follow this procedure:

1) Unplug all USB devices from the P.C.
2) Connect the Easy Input device to a USB connector on the P.C.
3) Allow enumeration to complete (wait for the hourglass to disappear).
4) Select Start, Settings, Control Panel, System, and Device Manager.
5) Expand "Human Input Devices"
6) Highlight "USB Human Interface Device"
7) Click on remove.
8) Unplug the Easy Input device from the P.C.

First use can now be tested by following the first use procedure at the beginning of this section.

UINTENTIONAL DATA OR CONTROL SEQUENCES

Be aware that the Easy Input device is capable of sending virtually any key sequence that can be sent by a conventional keyboard. Therefore, the developer must be cautious to avoid key sequences, which may interfere with the application software, windows or driver loading by sending other unintentional data.

Generally, this can be avoided by only sending keys or key sequences which represent data to the application it is to be used with. Be cautious of sequences such as ALT F3, etc. And, always test your usage thoroughly under both first time and normal usage scenarios.

USB TEST MODE

USB test mode pin definitions

Pin #, 24 pin device	Description	Signal
1	Parallel data bit 0 (lsb)	NC
2	Parallel data bit 1	NC
3	Parallel data bit 2	NC
4	Parallel data bit 3	NC
24	Parallel data bit 4	NC
23	Parallel data bit 5	NC
22	Parallel data bit 6	NC
21	Parallel data bit 7 (msb)	NC
5	/Clock	Tie low
20	Busy	NC
6	Mode 0	0
19	Mode 1	0
7	/keyboard shift	NC
18	/keyboard control	NC
8	/keyboard alt	NC
17	NC	NC
12		NC
13		NC
11	Vreg	1.3k to pin 15
16	USB D+ Connection	
15	USB D- Connection	
10	Vpp	Tie low
14	Vcc	+5v
9	Vss	Ground

USB Test Mode Usage:

1) Bias the chip as described by the schematic and USB test mode pin descriptions.
2) Follow the “First time usage in any mode” procedure above
3) After driver installation is complete, unplug the USB cable
4) Open a blank text or word processor document
5) Plug the USB cable back into the PC’s USB port
6) The test message will be entered into the document
7) If desired, repeat steps 4 and 5 using a blank spreadsheet.

PARALLEL DATA MODE

Parallel data mode pin definitions

Pin #, 24 pin device	Description	Signal
1	Parallel data bit 0 (lsb)	Data bit 0 (lsb)
2	Parallel data bit 1	Data bit 1
3	Parallel data bit 2	Data bit 2
4	Parallel data bit 3	Data bit 3
24	Parallel data bit 4	Data bit 4
23	Parallel data bit 5	Data bit 5
22	Parallel data bit 6	Data bit 6
21	Parallel data bit 7 (msb)	Data bit 7 (msb)
5	/Clock	/Clock
20	Busy	Hi indicates busy
6	Mode 0	1
19	Mode 1	0
7	/keyboard shift	Optional, leave open if not used
18	/keyboard control	Optional, leave open if not used
8	/keyboard alt	Optional, leave open if not used
17	NC	NC
12		NC
13		NC
11	Vreg	1.3K to pin 15
16	USB D+ Connection	
15	USB D- Connection	
10	Vpp	Tie low
14	Vcc	+5v
9	Vss	Ground

Parallel Mode Data Transfers:

Parallel mode data transfers can be achieved through one of three different methodologies: Open Loop, Busy Feedback or Fully Automatic. The following tables describe these methods in terms of correspondence between advantages, design algorithm and end user requirements. In all designs, it is important to keep the Vcc for the Easy Input circuitry in the schematic above separate from the Vcc of the interface circuitry.

Open Loop Data Transfers

Advantages / Disadvantages	Design Algorithm	End User Must Insure The Following Before Sending Data
1) Easiest to implement 2) Slow data transfer	1) Place data on data lines 2) Drop /Clock low. 3) Delay for 128uS 4) Bring /Clock high. 5) Delay 33ms for data transfer 6) Repeat for each data byte (Note: 33ms should work on almost any system. However, if data loss occurs, the problem can be corrected by increasing this value.)	1) Application is open 2) Cursor is located where data is to start sending. 3) USB connector is connected. 4) USB enumeration has completed (when enumeration is complete, the "hour glass" on the PC will disappear).

Busy Feedback Data Transfers

Advantages / Disadvantages	Design Algorithm	End User Must Insure The Following Before Sending Data
1) Faster data transfer	1) Place data on data lines 2) Wait for not Busy 3) Delay for 128uS 4) Drop /Clock low. 5) Delay for 128uS 6) Bring /Clock high. 7) Repeat for each data byte.	1) Application is open 2) Cursor is located where data is to start sending. 3) USB connector is connected. 4) USB enumeration has completed (when enumeration is complete, the "hour glass" on the PC will disappear).

Fully Automatic Data Transfers

Advantages / Disadvantages	Design Algorithm	End User Must Insure The Following Before Sending Data
1) Faster data transfer 2) Data transfers automatically when USB connector is plugged in.	1) Place first data on data lines 2) Wait for not Busy 3) Delay for 1.28 sec. 4) Wait for not Busy 5) Delay for 128uS 6) Drop /Clock low 7) Delay for 128uS. 8) Bring /Clock high. 9) Place next data on data lines 10) Loop to step 4 (See note below on circuit preparation for fully automatic mode)	1) Application is open 2) Cursor is located where data is to start sending.

Note: Initially, during Fully Automatic Operation, the Easy Input device and its immediate circuitry (see schematic) will not have power because the USB connector has not been plugged in yet. Therefore, the circuitry interfacing to the Easy Input device must pull-up the Busy line to its own separate 5v-power supply. A 10k pull-up resistor is recommended. The pulled up /Busy line will tell the interface circuitry to hold the data. Then, when the USB connection is made, the Easy Input device will pull the line low indicating that the connection has been made.

/Control, /Shift and /Alt input lines:

These lines modify data sent to the EASY INPUT chip in the same way as a Left Control, Left Shift or Left Alt key on a standard keyboard. These inputs are available on 24 lead chips only. They can be left open if not used. To assert any of these lines, pull it low and keep it low while the corresponding data byte is being sent to the EASY INPUT chip.

Note also that for many of the ASCII characters accepted by the EASY INPUT chip, the use of one of these modifier lines is implied. For example, capitol letters and many characters which are accessed by first holding the shift key can be sent without using the /shift input line. See the ASCII to USB conversion table for details.

Since the Easy Input device uses the /Shift key to create capitol letters, a /Shift will be generated whenever capitol letters are used. This is important to know because, in cases where it is not desirable to generate a /Shift, lower cases letters must be used.

For example, "<CTRL> V" is the "paste" command in many windows applications. But "<CTRL><SHIFT> V" will not perform a paste function. Since the Easy Input device always uses "<SHIFT> v" to generate "V" (capitol V), the "<CTRL> V" (control, capitol V) sequence will not perform a paste function. To correct this, use " <CTRL> v" (control, lower case v). The resulting string will be "<CTRL>v" and the paste function will be performed.

SERIAL DATA MODE

Serial data mode pin definitions

Pin #, 24 pin device	Description	Signal
1	Serial data bit	Data bit
2	Tristate with weak pullup	NC
3	Tristate with weak pullup	NC
4	Tristate with weak pullup	NC
24	Tristate with weak pullup	NC
23	Tristate with weak pullup	NC
22	Tristate with weak pullup	NC
21	Tristate with weak pullup	NC
5	/Clock	/Clock
20	Busy	Hi indicates busy
6	Mode 0	0
19	Mode 1	1
7	Tristate with weak pullup	NC
18	Tristate with weak pullup	NC
8	Tristate with weak pullup	NC
17	NC	NC
12		NC
13		NC
11	Vreg	1.3K to pin 15
16	USB D+ Connection	
15	USB D- Connection	
10	Vpp	Tie low
14	Vcc	+5v
9	Vss	Ground

Serial Mode Protocol:

Serial data is transferred to and from the Easy Input device using a synchronous serial interface. That interface consists of a clock bit, busy bit and a data bit. Data will be sent to/from the device in the form of a packet. Each packet consists of 19 bits of input data followed by 6 bits of output data. The entire packet must be clocked into / out of the device before the next packet can begin.

Serial Mode Clocking Method

Starting condition: /Clock line is high, Data line is tri-stated (not asserting data into the Easy Input chip)
Reset condition: Delay longer than serial timeout.

1) Wait for the Busy line to be low
2) Assert the data bit on the data line
3) Delay at least 5us for data to settle
4) Drive the /Clock line low
5) Wait for Busy high
6) Drive the /Clock line high
7) Repeat steps 1-6 for all input bits
8) Wait for Busy low
9) Drive the /Clock line low
10) Wait for Busy high
11) Read an output bit
12) Drive the /Clock line high
13) Repeat steps 8-12 for all output bits

Serial Mode Packet Definition

Bit Sequence in Packet	Bit Definition	Data Direction.
1	Data Bit 0 (lsb)	Input
2	Data Bit 1	Input
3	Data Bit 2	Input
4	Data Bit 3	Input
5	Data Bit 4	Input
6	Data Bit 5	Input
7	Data Bit 6	Input
8	Data Bit 7	Input
9	Left CTRL	Input
10	Left SHIFT	Input
11	Left ALT	Input
12	Left GUI	Input
13	Right CTRL	Input
14	Right SHIFT	Input
15	Right ALT	Input
16	Right GUI	Input
17	0=Send character, 1=Read status only	Input
18	Must be 0	Input
19	Must be 0	Input
20	Num Lock Status	Output
21	Caps Lock Status	Output
22	Scroll Lock Status	Output
23	Reserved Output	Output
24	Reserved Output	Output
25	Reserved Output	Output

Serial Mode Data Transfers:

The following tables describe these methods in terms of correspondence between advantages, design algorithm and end user requirements. In all designs, it is important to keep the Vcc for the Easy Input circuitry in the schematic above separate from the Vcc of the interface circuitry.

Open Loop Data Transfers

Advantages / Disadvantages	Design Algorithm	End User Must Insure The Following Before Sending Data
1) Easiest to implement 2) Slow data transfer	7) Place data on data lines 8) Drop /Clock low. 9) Delay for 128uS 10) Bring /Clock high. 11) Delay 33ms for data transfer 12) Repeat for each data byte (Note: 33ms should work on almost any system. However, if data loss occurs, the problem can be corrected by increasing this value.)	5) Application is open 6) Cursor is located where data is to start sending. 7) USB connector is connected. 8) USB enumeration has completed (when enumeration is complete, the "hour glass" on the PC will disappear).

Busy Feedback Data Transfers

Advantages / Disadvantages	Design Algorithm	End User Must Insure The Following Before Sending Data
2) Faster data transfer	8) Place data on data lines 9) Wait for not Busy 10) Delay for 128uS 11) Drop /Clock low. 12) Delay for 128uS 13) Bring /Clock high. 14) Repeat for each data byte.	5) Application is open 6) Cursor is located where data is to start sending. 7) USB connector is connected. 8) USB enumeration has completed (when enumeration is complete, the "hour glass" on the PC will disappear).

Reading Lock Key Status

The status bits returned from the Easy Input device contain the status of the Lock Keys (Caps Lock, Scroll Lock and Num Lock). Since that status information must be routed through the host, a time lag will occur between power up and valid lock key data and a time lag will also occur between toggling the lock keys and valid lock key data. Therefore, it is not valid to change a lock key and read back its status in the same packet. To guarantee valid lock key status, it should be read back from the Easy Input chip at frequent intervals.

Note that bit 17 can be used to read back the lock key status without sending data. In this case, the full packet must be clocked, but the data and control information is ignored.

Serial Mode Timeout

Serial mode has a built in timeout. The timeout will reset the serial bit stream to its beginning. The reset will occur when time between individual bits, either sent or received, exceeds the timeout value.

Function	Time between serial bits
No reset	< 18mS
Reset will occur	> 20mS

Although it is not necessary, it is recommended that the timeout be exceeded on a periodic basis to assure synchronization of the serial bit stream to the Easy Input device.

/Control, /Shift and /Alt input lines:

These lines modify data sent to the EASY INPUT chip in the same way as a Left Control, Left Shift or Left Alt key on a standard keyboard. These inputs are available on 24 lead chips only. They can be left open if not used. To assert any of these lines, pull it low and keep it low while the corresponding data byte is being sent to the EASY INPUT chip. The /Control, /Shift and /Alt input lines are used with parallel mode only. They are ignored in serial mode. However, serial mode can send these control lines in a packet format (see the serial mode section).

Note also that for many of the ASCII characters accepted by the EASY INPUT chip, the use of one of these modifier lines is implied. For example, capitol letters and many characters which are accessed by first holding the shift key can be sent without using the /shift input line. See the ASCII to USB conversion table for details.

Since the Easy Input device uses the /Shift key to create capitol letters, a /Shift will be generated whenever capitol letters are used. This is important to know because, in cases where it is not desirable to generate a /Shift, lower cases letters must be used.

For example, "<CTRL> V" is the "paste" command in many windows applications. But "<CTRL><SHIFT> V" will not perform a paste function. Since the Easy Input device always uses "<SHIFT> v" to generate "V" (capitol V), the "<CTRL> V" (control, capitol V) sequence will not perform a paste function. To correct this, use " <CTRL> v" (control, lower case v). The resulting string will be "<CTRL>v" and the paste function will be performed.

ASCII to USB conversion table

ASCII character sent to the EASY INPUT chip	Character sent to the USB system
00h	<escape>
01h	F1
02h	F2
03h	F3
04h	F4
05h	F5
06h	F6
07h	<insert>
08h	<backspace>
09h	<tab>
0Ah	<windows>
0Bh	<menu>
0Ch	<print screen>
0Dh	<CR>
0Eh	<Pause / Break>
0Fh	<Num Lock>
10h	<Caps Lock>
11h	<Scroll Lock>
12h	F7
13h	F8
14h	F9
15h	F10
16h	F11
17h	F12
18h	<up arrow>
19h	<down arrow>
1Ah	<right arrow>
1Bh	<left arrow>
1Ch	<home>
1Dh	<end>
1Eh	<page up>
1Fh	<page down>
20h	<space>
21h	!
22h	“
23h	#
24h	$
25h	%
26h	&
27h	‘
28h	(
29h	)
2Ah	*
2Bh	+
2Ch	,
2Dh	-
2Eh	.
2Fh	/
30h	0
31h	1
32h	2
33h	3
34h	4
35h	5
36h	6
37h	7
38h	8
39h	9
3Ah	:
3Bh	;

3Ch	<
3Dh	=
3Eh	>
3Fh	?
40h	@
41h	A
42h	B
43h	C
44h	D
45h	E
46h	F
47h	G
48h	H
49h	I
4Ah	J
4Bh	K
4Ch	L
4Dh	M
4Eh	N
4Fh	O
50h	P
51h	Q
52h	R
53h	S
54h	T
55h	U
56h	V
57h	W
58h	X
59h	Y
5Ah	Z
5Bh	[
5Ch	\
5Dh	]
5Eh	^
5Fh	_
60h	`
61h	a
62h	b
63h	c
64h	d
65h	e
66h	f
67h	g
68h	h
69h	i
6Ah	j
6Bh	k
6Ch	l
6Dh	m
6Eh	n
6Fh	o
70h	p
71h	q
72h	r
73h	s
74h	t
75h	u
76h	v
77h	w
78h	x
79h	y
7Ah	z
7Bh	{

7Ch	|
7Dh	}
7Eh	~
7Fh	<delete>
80h	HID "Non-English" Key

Packaging

Please specify package configuration:

24 pin SOIC
24 pin DIP

Disclaimer

To the best of our knowledge, Radovan Robotics has built the Easy Input device to meet the USB (Universal Serial Bus) standard. By designing with or using this product, you are agreeing not to hold liable Radovan Robotics for any damages or damage to electronic hardware or software associated with the use of this product.

Compatibility of the EASY INPUT chip is based solely on Radovan Robotics best estimate of a "standard" USB keyboard. While it is in the best interest of Radovan Robotics to attempt to achieve maximum compatibility with all host systems, due to individual design by other manufacturers, Radovan Robotics does not claim or imply to claim compatibility with any given system.

RADOVAN ROBOTICS
PO Box 261,
Newbury Park, CA
91319-0261

PHONE: 805-375-7059
FAX: 805-375-6965

–