Hardware and Electronics Reference

for the

Palm® and Stowaway™ Portable Keyboards

Revision 1.1

www.thinkoutside.com

Table of Contents

I. Introduction

Think Outside is the inventor of the Stowaway Portable Keyboard, the world’s first 100% full-size keyboard that folds to handheld dimensions.   Versions of our keyboard that support Palm handhelds are distributed by Palm, Inc. under the name **Palm Portable Keyboard**. Keyboards that support Handspring Visor handhelds, the HP Jornada 540 series, and the Compaq iPaq PocketPC are distributed by Targus Group International under the name **Targus Stowaway Portable Keyboard**.

What does this document cover?

This document covers the **hardware and electronics specifications** for both the Targus Stowaway and Palm Portable Keyboards, supporting the following devices:

- **Palm Handhelds**
- **Handspring Visor Handhelds**
- **HP Jornada 540 Series**
- **Compaq iPaq PocketPC**

Who should reference this document?

This document is intended for mobile device developers who wish to support our keyboard as an input solution for their products. Typically, the developers who reference the information contained in this document are building hardware connectors, writing software drivers, and integrating enterprise solutions.

Where is the latest Developer Support information?

The latest developer information is available on our web site:

> ***http://www.thinkoutside.com/devsupport/***

When you sign up, you may download the latest developer materials. You may also specify that you wish to receive email notice when the materials are updated.

Questions and suggestions about developer support may be sent to:

> ***devsupport@thinkoutside.com***.

How can I contact Think Outside?

Think Outside is headquartered in Carlsbad, California, about 30 miles north of downtown San Diego.

> **Think Outside**
> **5790 Fleet Street Suite 130**
> **Carlsbad, California 92008**
> **(760) 431-9090**
> ***www.thinkoutside.com***

We welcome questions, comments, and other feedback regarding our developer support. Please send email to ***devsupport@thinkoutside.com***.

II. Document Overview

The technical information in this document has been organized into four areas:

- **Physical Connectors** - Pin assignments and electrical connections.
- **Keyboard Detection** - How a device may typically detect a keyboard and begin communications.
- **Interface Details** - Notes about communications and interface.
- **Key Code Matrix** - Keyboard signal -> key code translation.

III. Document Conventions

The keyboards supported in this document are sold under the names:

- *Palm Portable Keyboard*
- *Targus Stowaway Portable Keyboard for Handspring Visor*
- *Targus Stowaway Portable Keyboard for HP Jornada*
- *Targus Stowaway Portable Keyboard for Compaq iPaq PocketPC*

In this document, we may simply refer to a keyboard as the **Stowaway Portable Keyboard** or **Keyboard for [*a supported device*]**.

IV. Physical Connectors

Pin Assignments for Palm Handhelds

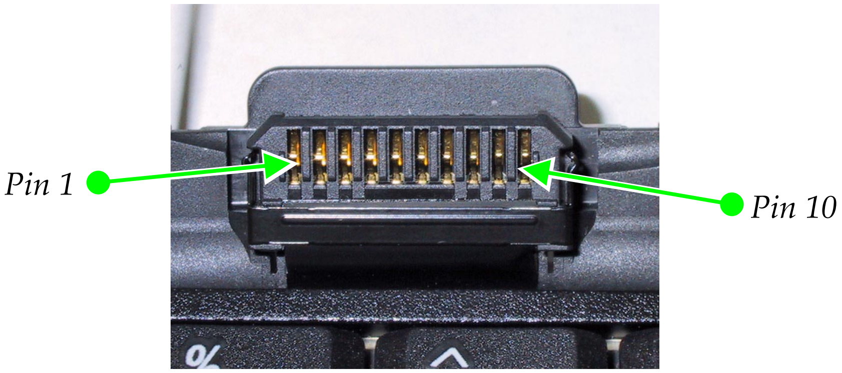

Figure 1: Connector Pins for Palm Keyboards

The pin assignments are:

Pin 1	N/C
Pin 2	**VCC** – Input to keyboard (3.0V DC to 5.5V DC).
Pin 3	**RXD** – Output from keyboard.
Pin 4	**RTS** – Input to the keyboard.
Pin 5	N/C
Pin 6	N/C
Pin 7	**HotSync** – Output from keyboard, used for handshaking.
Pin 8	N/C
Pin 9	N/C
Pin 10	**GROUND**

Pin Assignments for Handspring Visor Handhelds

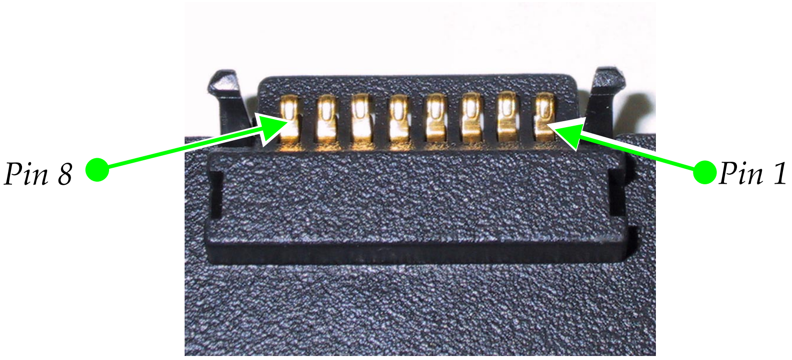

Figure 2: Connector for Handspring Visor Keyboards

The pin assignments are:

Pin 8	**VCC / TXD**
Pin 7	N/C
Pin 6	N/C
Pin 5	N/C
Pin 4	**GROUND**
Pin 3	N/C
Pin 2	N/C
Pin 1	**RXD** – Output from keyboard (receive serial line to device).

Pin Assignments for HP Jornada 540 Series

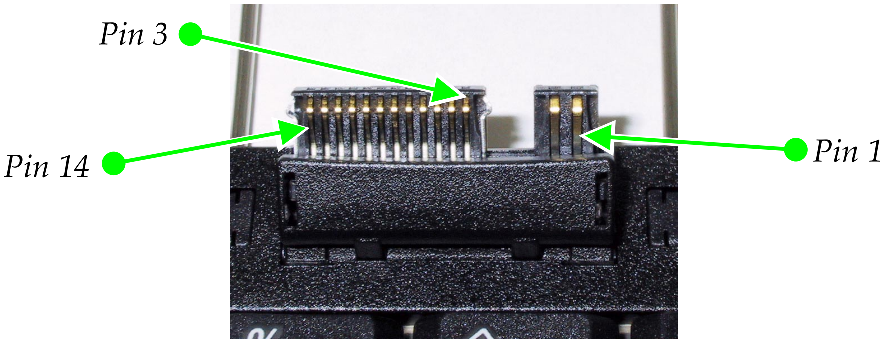

Figure 3: Connector for HP Jornada 540 Series Keyboard

The pin assignments are:

Pin 14	N/C
Pin 13	N/C
Pin 12	N/C
Pin 11	**GROUND**
Pin 10	N/C
Pin 9	**RTS** – Signal input to keyboard.
Pin 8	N/C
Pin 7	**DTR (VCC)** – Power input to keyboard (3.0V to 5.5V DC), NOT active when unit is powered off.
Pin 6	N/C
Pin 5	**RXD** – Output from keyboard (receive serial line to device).
Pin 4	**DCD (ActiveSync)** – Output from keyboard (used for handshaking).
Pin 3	N/C
Pin 2	**DC_ Adapter Ground** – (used as feed through to PDA).
Pin 1	**DC_Adapter_Voltage** – (used as feed through to PDA).

Pin Assignments for Compaq iPaq PocketPC

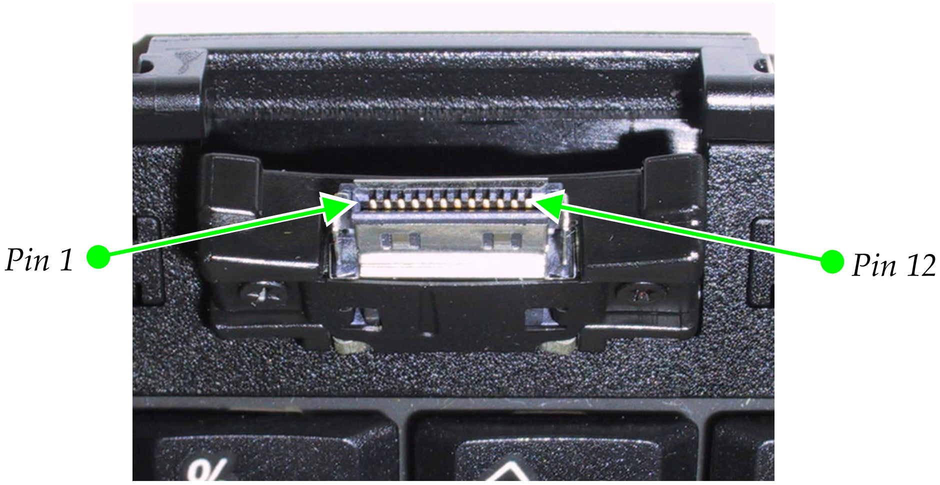

Figure 4: Connector for Compaq iPaq PocketPC Keyboard

The pin assignments are:

Pin 1	N/C
Pin 2	N/C
Pin 3	**DTR** – Power input to keyboard (3.0V to 5.5V DC).
Pin 4	N/C
Pin 5	N/C
Pin 6	**RTS** – Signal input to keyboard.
Pin 7	N/C
Pin 8	**RXD** – Output from keyboard (receive serial line to device).
Pin 9	**DCD** (**ActiveSync)** – Output from keyboard (used for handshaking).
Pin 10	**GROUND**
Pin 11	N/C
Pin 12	N/C

V. Keyboard Detection

This section covers keyboard detection sequences for:

1. Palm and PocketPC Devices
2. Handspring Devices

Detecting Keyboards for Palm and PocketPC Devices

If you are developing for a ***Handspring*** keyboard, please skip to the next section, ***Detecting Keyboards for Handspring Devices***.

POWERING UP THE KEYBOARD:

Palm Devices	PocketPC Devices
The keyboard electronics are powered up when the signal on the VCC line jumps from low to high, typically by attaching a powered Palm device.	On power cycling the PocketPC device attached to a keyboard: • Open serial port. • Make sure RTS is LOW. • Raise DTR (send power to keyboard).

After the keyboard electronics are powered up, the keyboard will send a pulse on the HotSync or DCD (ActiveSync) line.

ON DCD PULSE:

Upon detecting a low-to-high signal edge on the DCD line, your driver should:

- Open the serial port, if not already open.
- If the signal on the RTS line is LOW, RAISE it from low to HIGH.
- If the signal on the RTS line is HIGH, TOGGLE it from high to LOW back to HIGH.
- If the attached keyboard detects the low-to-high edge on the RTS line, then it sends an ID string of two bytes over the RXD line. The hex ID string is: **FA FD**
- If the driver receives this ID, the keyboard is attached. The driver is now ready to receive key codes over the serial line. A matrix of key codes is included towards the end of this document.
- ***Note: If keys are depressed when the RTS signal is raised, the keyboard will also send the key codes of all depressed keys addition to the ID string.***
- If the driver does NOT receive this ID, the keyboard is NOT attached. The driver should lower the RTS line, close the serial port, and pass the sync (HotSync or ActiveSync) notification on to the operating system.

LOW POWER MODE:

You may put the keyboard into low power mode by lowering the signal on the RTS line. For example, after a 5-second timeout (no key presses for 5 seconds), the Think Outside driver lowers the signal on the RTS to tell the keyboard to go into low power mode, and then closes the serial port. This conserves power usage for the keyboard.

The keyboard may automatically enter low power mode and NOT respond to key presses (see next section below, DETECTING KEY PRESSES, for details). To reset the keyboard from these states, lower or toggle the signal from ***high to low*** on the RTS line.

DETECTING KEY PRESSES:

If the keyboard is powered up and RTS is high, the key code is simply transmitted over the serial line.

If the keyboard is powered up and RTS is low, pressing a key will send a pulse on the HotSync / DCD line (in addition to the key code on the serial line). On the pulse, the driver should check for the keyboard as detailed above.

WARNING: The keyboard will NOT send HotSync / DCD pulses under these circumstances:

- **AUTOMATIC LOW POWER:** If the keyboard is powered up, and the RTS line remains HIGH for more than 10 minutes, the keyboard will automatically drop into low power mode. ***Note that the signal on RTS may remain HIGH***. To reset the keyboard from this state, toggle the signal from ***high to low back to high*** on the RTS line. The keyboard will send its ID in response to the final raise of the RTS line. No HotSync/DCD pulses will be generated.
- **DRIVER IGNORES 3 SUCCESSIVE PULSES**: If the driver does NOT raise RTS after a HotSync / DCD pulse, the keyboard will send another pulse. If the driver ignores the second pulse, the keyboard will send a third pulse. ***If the driver ignores the third pulse, the keyboard will go to sleep and NOT RESPOND TO KEY PRESSES.*** To reset the keyboard from this state, raise the signal from ***low to high*** on the RTS line. If the RTS is already high, toggle the line from ***high to low back to high***.

CHECKING FOR DETACHED KEYBOARD:

There is no dedicated line to indicate a detached keyboard. A detached keyboard is indicated if raising the RTS line from high to low does NOT return a keyboard ID on the RXD line.

Check for a detached keyboard on at least 2 occasions:

- **On HotSync / DCD Pulse** - detailed sequence above.
- **During Key Repeat** - Your driver should also check for a detached keyboard during key repeat (for example, after every 10 keys).

Checking for a detached keyboard during key repeat precludes this type of scenario: The user depresses the Delete key and holds it to repeat. The keyboard is jostled and becomes detached. Since the key up code for Delete key was never detected, the delete is free to continue to repeat until the entire text field is deleted. However, if the driver checks for the detached keyboard after every 10 keys, the damage is limited to 10 deletes.

Detecting Keyboards for Handspring Devices

If you are developing for a ***Palm or PocketPC*** keyboard, please skip this section and refer to the previous section, ***Detecting Keyboards for Palm and PocketPC Devices***.

CHECK FOR KEYBOARD ON POWER UP:

The keyboard electronics are powered up when the signal on the VCC line jumps from low to high, typically by power cycling a device attached to the keyboard.

Approximately 12-15 milliseconds after the keyboard electronics are powered up, the keyboard will send an ID string of two bytes over the RXD line. The hex ID string is:

F9 FB

Your driver must be ready to receive this ID to detect the keyboard. Typically, you would open the serial port during the power up sequence of your device to prepare your driver for receiving this ID.

The serial port should stay open until the power is turned off, or until the keyboard is detached. ***Note that there is no handshaking involved.***

CHECKING FOR DETACHED KEYBOARD:

There is no dedicated line to indicate a detached keyboard. To check for a detached keyboard, toggle the VCC / TXD line (lower then raise the signal). An attached keyboard will return the ID string (**F9 FB**) on the RXD line. If no ID string is received, your keyboard has been detached, and you should close the serial port.

Your driver should check for a detached keyboard:

- **At Regular Intervals** -   For example, the Think Outside driver for the Handspring Keyboard checks for a detached keyboard every 30 seconds.
- **During Key Repeat** - Your driver should also check for a detached keyboard during key repeat (for example, after every 10 keys).

Checking for a detached keyboard during key repeat precludes this type of scenario: The user depresses the Delete key and holds it to repeat. The keyboard is jostled and becomes detached. Since the key up code for Delete key was never detected, the delete is free to continue to repeat until the entire text field is deleted. However, if the driver checks for the detached keyboard after every 10 keys, the damage is limited to 10 deletes.

Note: If keys are depressed when the when power is applied, the keyboard will also send the key codes of all depressed keys addition to the ID string.

VI. Interface Details

Here are the communications and interface details for the keyboards. Note the differences between the Palm / PocketPC keyboards and the Handspring keyboards.

Communications Interface

Communications interface details for all keyboards:

- 9600 bps
- 8 Data Bits
- No Parity Bit
- 1 Stop Bit
- Includes a start bit which is always in communication.

Palm and PocketPC Keyboards	Handspring Keyboards
Data is sent at **RS-232** level.	Data is sent at **TTL** level.

Basic Signal Interface

Minimum set of signals required for keyboard functionality.

- **GROUND** – the connection to ground from the PDA or controlling device.
- **VCC or DTR** – Power connection from the PDA or controlling device. The voltage level can be 3.0V DC to 5.5V DC. Power consumption is no more than2.5 mA when a key is depressed and less than 5 µA when the keyboard is sitting in power save mode.
- **RXD** – Receive data signal from the keyboard to the PDA or the controlling device. The signal is generated from the drain of a MOSFET transistor which inverts the signal from the keyboard microcontroller. This line is reliant on a pull-up resistor (100k) internal to the PDA or controlling device.

Handshake Signal Interface

Palm and PocketPC Keyboards	Handspring Keyboards
DCD (PocketPC ActiveSync or Palm HotSync): • Available for handshake keyboard detection. • When the keyboard first sees power, or a key is depressed while the attached keyboard is in low power mode, the keyboard pulses the HotSync / DCD signal. • See section on ***Keyboard Detection*** for details. • SIGNAL LEVELS – Data may be sent at RS-232 levels. The signal level depends on the input voltage supplied by the device.	Handshake signal is **NOT** available.

Signal Transmission

Data which is sent across the RXD signal is sent least significant bit (LSB) first and most significant bit last (MSB). This is standard with RS-232 serial transmission.

Palm and PocketPC Keyboards	Handspring Keyboards
No special notes.	Data is sent in TTL format, not RS-232 format. For example, logic level 1 is HIGH for this transmission, whereas in a full fledge RS-232 serial transmission the data is sent inverted and the RS-232 transceiver at in the device handles it.

Keystrokes

Keyboard correctly transmits keystrokes at a rate up to 10 keys per second, and buffers up to 6 keystrokes:

- **Rate**: Up to 10 keys / second.
- **Buffer**: Up to 6 keys.

Key Down Code – When a key is depressed, one 8-bit byte is sent:

- Most significant bit: 0 for Key Down.
- Next four bits: Y Address for Key (Y0 through Y11)
- Least Significant 3 Bits: X Address (X0 through X7)
- Y & X addresses are specified in the ***Key Code Matrix***.

Key Up Code – When a key is released, one 8-bit byte is sent:

- Most significant bit: 1 for Key Up.
- Least Significant 7 Bits: Y & X Addresses, same as Key Down.

Last Key Up - When no other keys are depressed at the time a key goes up, the 8-bit byte for that key up is sent TWICE.

Example 1: Type the letter "a". The keyboard will send one key down code for the "a" key, followed by two key up codes.

Example 2: Type a Shift-A. The byte sequence may look something like this:

- One key down code for the left Shift key.
- One key down code for the "a" key.
- One key up code for the "a" key (left Shift key is still depressed).
- Two key up codes for the Left Shift key (last key up).

IMPORTANT - When your driver receives the second key up code for a Last Key Up, CLEAR ALL MODIFIER FLAGS.

Phantom Keys

A phantom key situation occurs by simultaneously pressing three keys that form three corners of any square or rectangle in the key matrix.

The phantom key algorithm is based on a 3-key rollover algorithm. If three keys are depressed and held down simultaneously that form a phantom key situation, key down codes will be sent for the first two keys to be depressed.

The key down code for the third key that was depressed will NOT be sent until either one of the first two keys is released.

2 and 3 Key Rollover

The keyboard can detect any two or three keys that are depressed simultaneously, provided that the three keys pressed simultaneously do not comprise three corners of a rectangle in the key matrix. This is covered by the phantom key algorithm.

Modifier Key Combinations

The keyboard firmware only transmits key coordinates, key down, and key up information. Modifier key combinations are treated the same as any other 2-key or 3-key rollover combinations.

Power Consumption

- Less than 5 μA in sleep mode.
- Less than 2.5 mA while key(s) is pressed.

Filters Key Bounce

- 15 - 25 milliseconds.

VII. Key Code Matrix

	X0	X1	X2	X3	X4	X5	X6	X7
Y0	! 1	@ 2	# 3	Z	$ 4	% 5	^ 6	& 7
Y1	CMMD	Q	W	E	R	T	Y	~ `
Y2	X	A	S	D	F	G	H	Space 1
Y3	CAPS LK	TAB	CTRL					
Y4			FN	ALT				
Y5					C	V	B	N
Y6	_ -	+ =	BACK SP	*Special Function One*	8	9	0	Space 2
Y7	{ [	{]	| \	*Special Function Two*	U	I	O	P
Y8	" '	ENTER	*Special Function Three*		J	K	L	: ;
Y9	? /	*Scroll Up* (up arrw)	*Special Function Four*		M	< ,	> .	DONE
Y10	DEL	(lt arrw)	*Scroll Dn* (dn arrw)	(rt arrw)				
Y11	SHIFT(L)	SHIFT(R)						