



Table of Contents

- 1. Overview..... 2
- 2. Publishing Software Applications on a Springboard Module..... 2
- 3. Building Specialized Hardware on a Springboard Module 3
- 4. Programming a ROM Image onto Your Module..... 3
 - 4.1. Method 1: Using the Palm Debugger 3
 - 4.2. Method 2: Using the CardUpdaterMaker SDK..... 4
 - 4.3. Restoring the 8MB Flash Module..... 4
- 5. The Diagnostic Module Schematic..... 5
- 6. History..... 10

List of Acronyms	
CS0*	Chip Select 0. An asterisk denotes an active low signal.
CS1*	Chip Select 1. An asterisk denotes an active low signal.
HPG Tools	Handspring Palm OS GNU Tools
PCMCIA	Personal Computer Memory Card International Association. Also known as PC Card.
POSE	Palm OS Emulator
UART	Universal Asynchronous Receiver/Transmitter

1. Overview

Handspring provides various tools to assist you in the test and integration phase of your module development. These tools are described below. In this application note, we identify two broad categories of Springboard module development. In the first category, a developer may want to simply utilize existing Palm OS applications and databases on a Springboard module. A reference guide application is an example of a product in this first category. The second category includes specialized hardware that interfaces with the handheld through Springboard. VisorPhone, Handspring's cellular phone module, is an example of a product in this second category.

2. Publishing Software Applications on a Springboard Module

You can develop your software using the Handspring Palm OS GNU (HPG) Tools. Alternatively, you can use any of the Palm OS development tools (e.g. CodeWarrior, etc.) that produce valid application and database files. Handspring-specific tools are only needed to migrate your developed application to a Springboard memory module.

You can also use standard Palm OS tools to test and debug your software. The Palm OS Emulator (POSE) is available on the Handspring website. Keep in mind, however, that POSE does not support Springboard module-specific API calls. Also note that testing in POSE is not a 100% substitute for testing on the actual target product. If you want to test Handspring extension API calls, you should load the software into a Handspring handheld's internal memory and execute/debug it from there. The Palm OS Debugger supports assembly and source level debugging. This debugger is distributed with the HPG Tools.

After your code is debugged, you should test it on a Springboard module. To clarify this point, applications and databases residing on a Springboard module are executed directly out of module memory. The CPU fetches instructions and accesses data directly from the memory on the module rather than making a copy in internal memory. There are a few differences when operating in this environment. For example, databases are read-only as they are now physically located in ROM (or Flash). Also, applications should avoid hard coding card 0 (internal memory) when opening a database which may reside on card 1 (Springboard module memory). Various Palm OS calls to find the database *will* return the appropriate card number.

To test your application in the Springboard environment, you will need to create a final image of your software (a ROM image) using the Palm-MakeROM utility. This utility is part of the HPG Tools. Complete details on use of the Palm-MakeROM tool is described in the "*Handspring Development Tools Guide*." This guide is located in the documentation section of Handspring's developer website.

<http://www.handspring.com/developers/documentation.jhtml>

Working examples of the Palm-MakeROM tool are incorporated into the makefiles of sample projects such as the DiagRefModule SDK. All SDKs are located in the software section of Handspring's developer website:

http://www.handspring.com/developers/sw_dev.jhtml

Whether the software will eventually be hosted on a masked ROM, Flash, or OTP module, the ROM image can be tested prior to mass production by loading the software onto a Handspring 8MB Flash Module. This module contains memory addressed by chip select 0 (CS0*). This module is available on Handspring's website. Handspring's 8MB Flash Module comes pre-loaded with file management software (FileMover) so moving your software to the module for testing purposes is simple. We will also look at re-programming the entire module (as you would to distribute your software) later in this document.

3. Building Specialized Hardware on a Springboard Module

Developers who are building specialized Springboard modules will have a more complicated testing task than those publishing software on ROM modules. In addition to testing module-specific software as described above, the hardware should also be tested in a Handspring handheld.

First, hardware developers need a means to monitor signals across the Springboard bus, and to test the module outside of the slot. While the connector core is the same as a 68-pin PCMCIA connector, keying features prevent PCMCIA cards from being inserted into a Springboard Expansion Slot. For testing purposes, an extender card is helpful. A list of vendors who provide Springboard extender cards are here:

<http://www.handspring.com/developers/palmooswtechres.jhtml>

If possible, your first generation hardware design (or separate test board) should be built using one of the flash types supported by Handspring's development tools. A complete listing of currently supported Flash components will be available on Handspring's website in a separate application note.

The ROM containing the card header information must use the CS0* signal for its chip select as described in the Springboard Development Guide. The ROM must also be 16-bit wide (8-bit wide devices are not supported).

Handspring has created a Springboard reference design called the Diagnostic Module. It contains AMD flash on CS0* and a UART on CS1*. The Diagnostic Module can be used by hardware developers to quickly emulate Springboard hardware and test the associated software. Software can be executed from the flash ROM on the Diagnostic Module. Springboard module hardware can be emulated on a PC and communicate with a Handspring handheld computer through a serial connector which is attached to the UART on the Diagnostic module. This provides a complete environment for developing and testing complex, hardware-specific Springboard software.

A more detailed description of the Diagnostic Module is provided in "*AN-03: Diagnostic Reference Module.*" The schematic for this reference design is attached.

4. Programming a ROM Image onto Your Module.

There are two ways to program the Flash memory on your module. The Palm Debugger is the first method that can be used to program a module, primarily for development and testing purposes. A second method is the Handspring supplied SDK called CardUpdaterMaker, the output of which can be used by end-customers. This SDK has the further advantage of supporting developer additions to supported Flash memory types.

4.1. Method 1: Using the Palm Debugger

The first method is to use the Palm Debugger, which supports Flash chips. This is primarily a method for developers to quickly program an image for development and testing purposes.

To overwrite the image on the 8MB Flash Module with your own, you'll need:

- A Handspring handheld and cradle (either serial or USB).
- Handspring's current release of the desktop software with HotSync software supporting USB transfer (even if you have the serial cradle).
- The Palm OS Debugger and Palm-MakeROM, which are distributed in the HPG Tools.

Here are step-by-step instructions for transferring your ROM image to the 8MB Flash Module for final test:

1. Create a ROM image using the Palm-MakeROM utility. Full details on the use of Palm-MakeROM can be found in the “*Handspring Development Tools Guide*.”
2. Copy your ROM image to the bin/device directory where you installed the Palm tools. If you used the default directory in the Setup program, this will be: c:\handspring\prc-tools\bin\device.
3. Launch the debugger on your desktop.
4. Enter debug mode on the handheld by using the following shortcut in the Find dialog:
 - Make sure that Hotsync is not running.
 - [shortcut] . 1
 - The shortcut character is drawn like a cursive lower case “L” character.
 - Two pen taps are needed to create a Graffiti period. The first pen tap indicates a punctuation character is coming. The second tap generates the period.
5. Type `att` in the debugger and press `Enter`. The debugger should respond with a welcome message. It should **not** include a timeout message. If a timeout message occurs, type `att` again, or soft reset your handheld and enter debug mode again.
6. Type `flash "yourRomImage" 28000000` (including the quotation marks), where `yourRomImage` is the name of your ROM image file created using the Palm-MakeROM tool.
7. Press `Enter`. The debugger will update a percentage complete measurement on the screen.
8. Type `reset` and press `Enter`, or press the reset button (soft-reset) on the back of the device.

4.2. Method 2: Using the CardUpdaterMaker SDK

The second method is to use a Handspring SDK called CardUpdaterMaker. This SDK is a GNU project which produces a user-friendly “Updater” application. The Updater is a standard Palm OS application which embeds your image file within it. To update Flash memory on a module, a user would simply install this application and launch it. Within the SDK is another project called FlashMgr. This is a library that CardUpdaterMaker uses to abstract the nuances of various Flash components. Flash components which are not currently supported by the SDK can be added here.

There are several components that may be helpful when using the CardUpdaterMaker SDK. All of these files are located on the software section of Handspring’s developer website:

http://www.handspring.com/developers/sw_dev.jhtml

Here are some step-by-step instructions to get working right away;



1. Download the SDK itself.
2. The HPG Tools themselves are also located on this page and are required to compile the SDK. The HPG Tools also require the Cygwin shell which is also located on this page.
3. The CardUpdaterMaker SDK requires a ROM image file. This image file is created by the Palm-MakeROM utility which is distributed with the HPG Tools.
4. To see a working sample of Palm-MakeROM within a project makefile, download the DiagRefModule SDK.

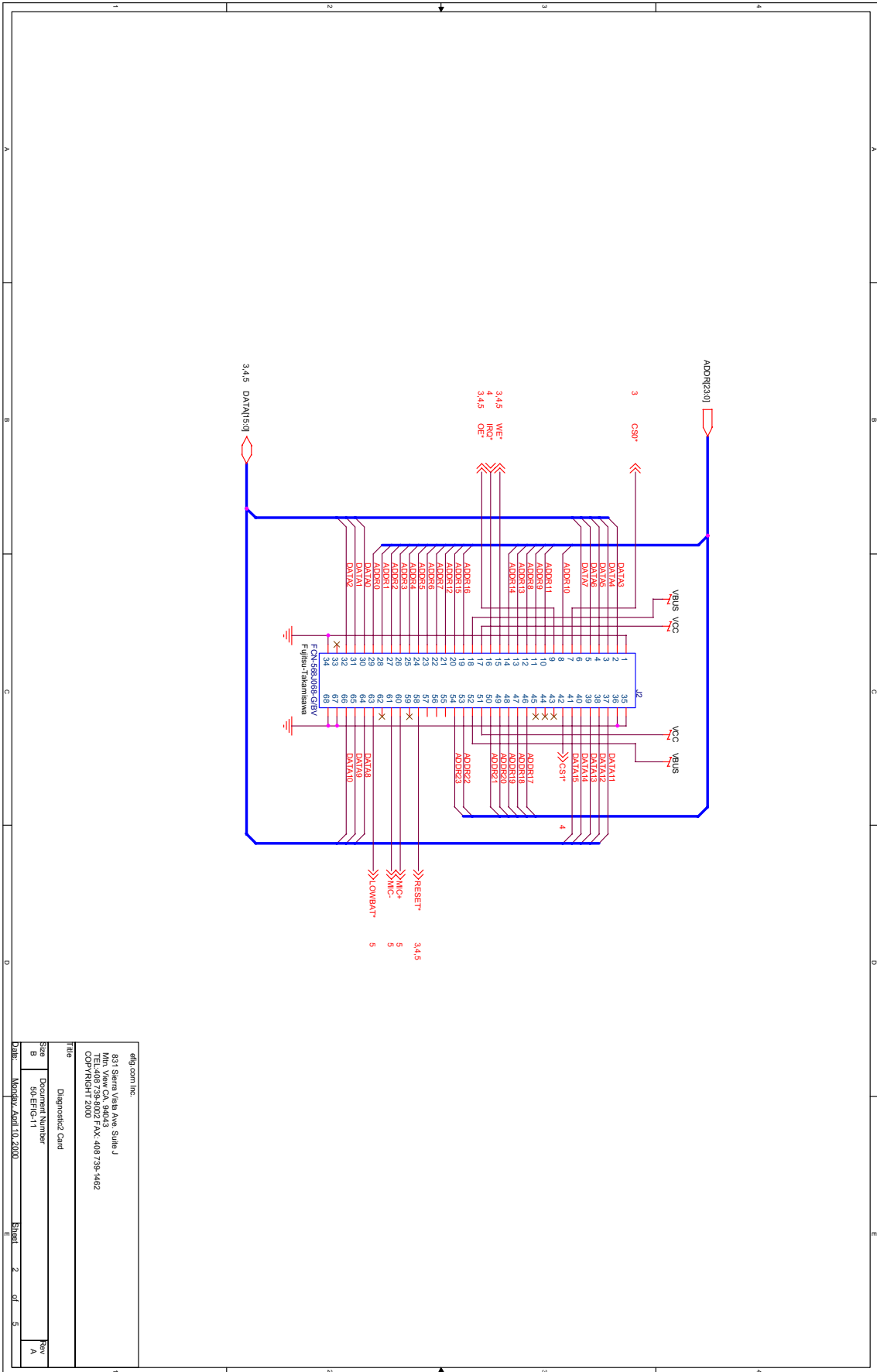
4.3. Restoring the 8MB Flash Module

There is a utility to restore the original image of the 8MB Flash Module. It is located on the Customer Service section of the Handspring website at:

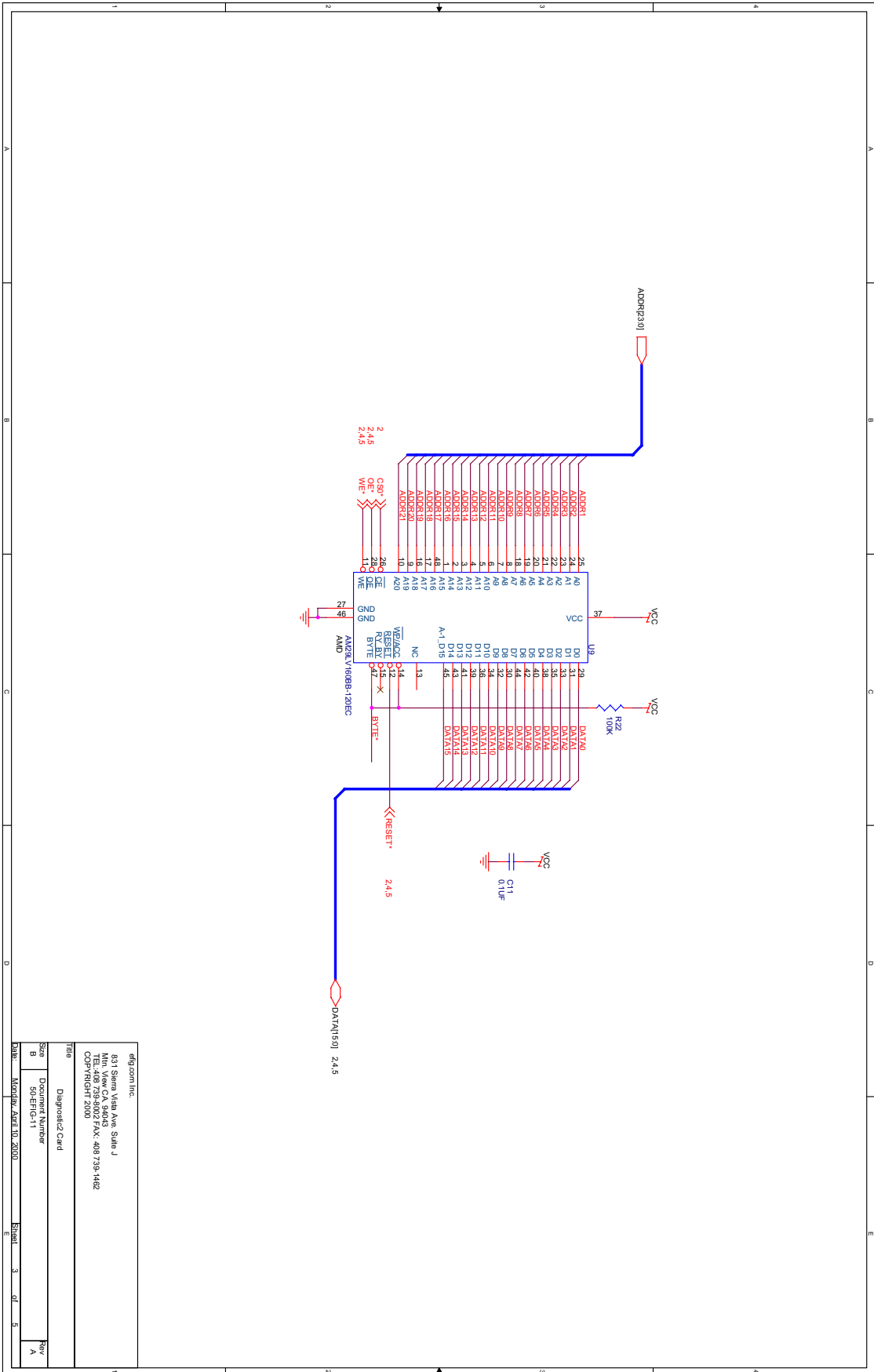
<http://www.handspring.com/support/softwareup.jhtml>

5. The Diagnostic Module Schematic

 Electronic Fast Integration Group															
<h1>Diagnostics Card</h1>															
<p>Page 1: Title Page Page 2: Expansion Connector Page 3: FLASH Page 4: RS-232 Page 5: Registers and MIC detect</p>															
		<table border="1"><tr><td colspan="3">efig.com Inc. 831 Sierra Vista Ave. Suite J Men. View CA. 94043 TEL: 408 739-8002 FAX: 408 739-1462 COPYRIGHT 2000</td></tr><tr><td colspan="3">Title: Diagnostic2 Card</td></tr><tr><td>Size: B</td><td>Document Number: 50-EFIG-11</td><td>Rev: A</td></tr><tr><td colspan="2">Date: Monday, April 10, 2000</td><td>Sheet: 1 of 5</td></tr></table>		efig.com Inc. 831 Sierra Vista Ave. Suite J Men. View CA. 94043 TEL: 408 739-8002 FAX: 408 739-1462 COPYRIGHT 2000			Title: Diagnostic2 Card			Size: B	Document Number: 50-EFIG-11	Rev: A	Date: Monday, April 10, 2000		Sheet: 1 of 5
efig.com Inc. 831 Sierra Vista Ave. Suite J Men. View CA. 94043 TEL: 408 739-8002 FAX: 408 739-1462 COPYRIGHT 2000															
Title: Diagnostic2 Card															
Size: B	Document Number: 50-EFIG-11	Rev: A													
Date: Monday, April 10, 2000		Sheet: 1 of 5													



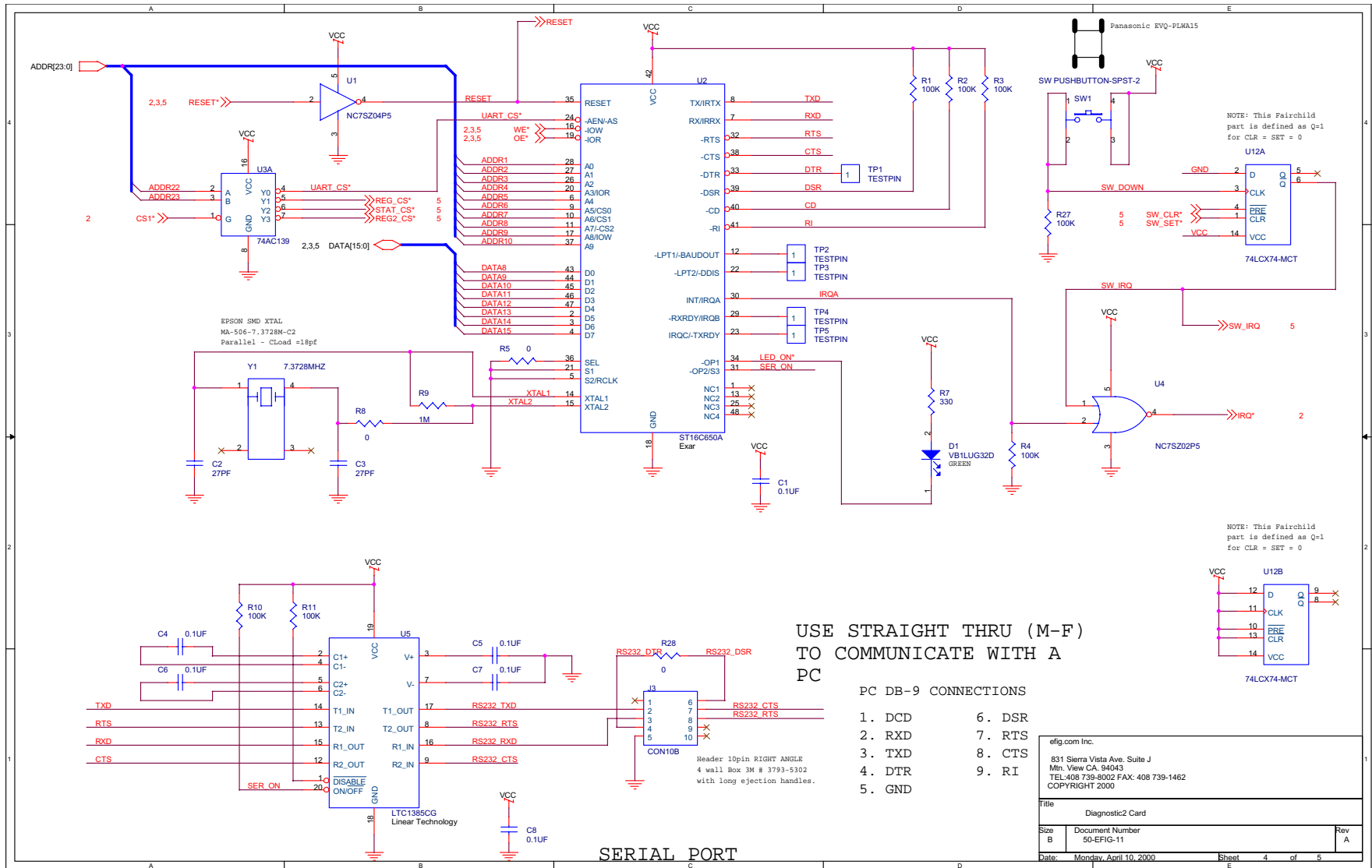
efig.com,inc		831 Sierra Vista Ave, Suite J	
10000 E. Mississippi Ave		Scottsdale, AZ 85258	
TEL: 480.739.8002 FAX: 480.739.1482		COPYRIGHT 2000	
Title: Diagnostic Card			
Size: B	Document Number: 58152-1	Rev: A	
Date: Modified: April 10, 2000	Sheet: 2	of: 5	

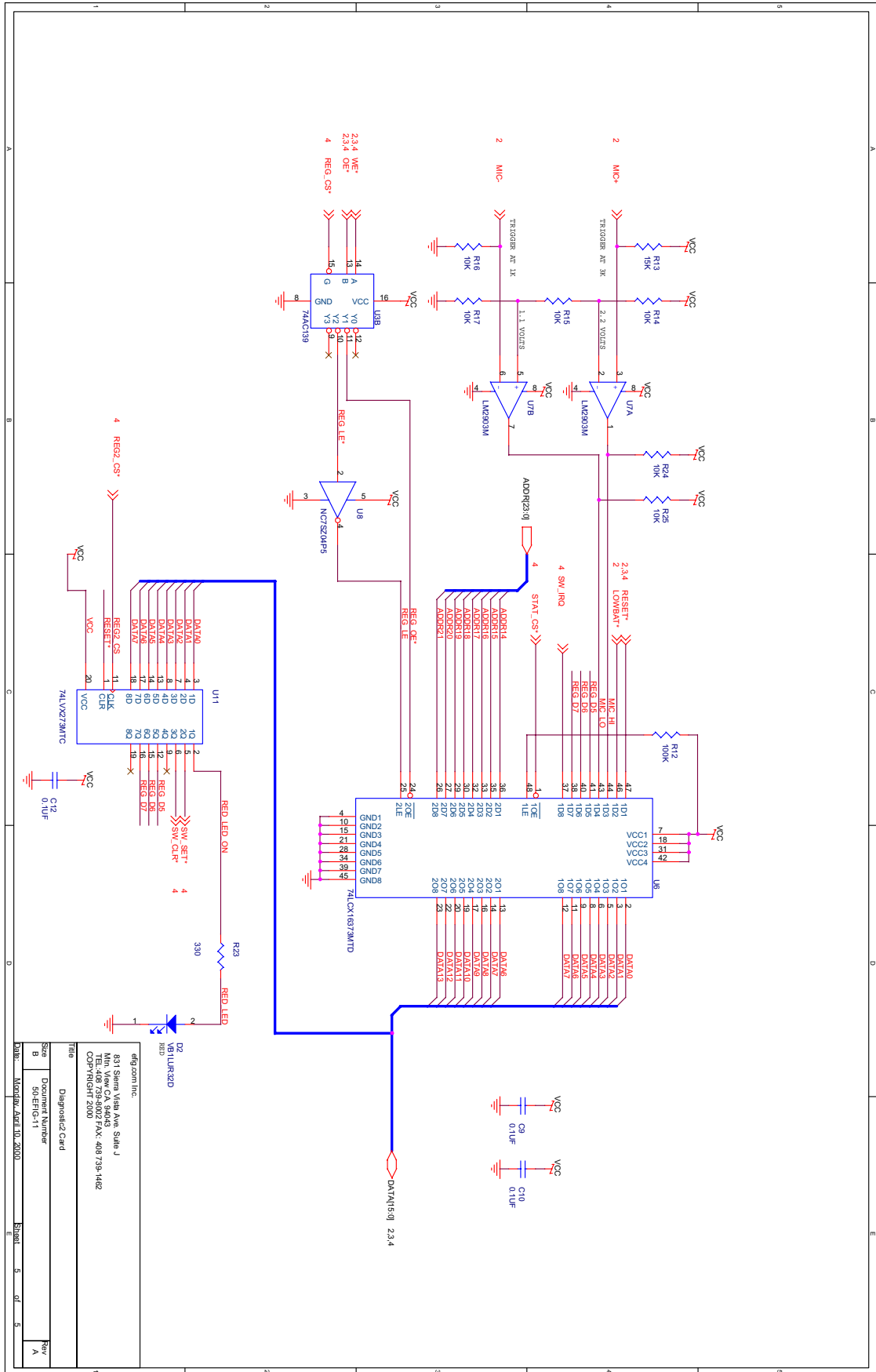


efig.com Inc.
 831 Sierra Vista Ave. Suite J
 Mil. View, CA, 94043
 TEL: 408-739-8002 FAX: 408-739-1482
 CDP-1605011 2000

Title		Diagnostic Card	
Size	Document Number	Sheet	Rev
B	50-ENIG-11	3	A

Date: Monday, April 10, 2000





Title		Diagnostic Card	
Size	B	Document Number	50-FIG-11
Date:	March, April, 2000	Sheet	5 of 5

efig.com Inc.
 831 Sierra Vista Ave. Suite J
 Milpitas, CA 95035
 TEL: 408 738-8022 FAX: 408 738-1482
 CDP-766871 2000

6. History

Date	Revision #	Description of changes
1 Mar 02	2.01	Update and reformatted. Removed reference to efig.com. Company is no longer selling the reference module.
21 Dec 00	2.00	Updated and reformatted. Correction: PCMCIA extender cards are not compatible with Springboard keying features. Added: CardUpdaterMaker SDK information.
10 Sep 99	1.00	Initial release.

Handspring™, Visor™, Springboard™, and the Handspring and Springboard logos are trademarks or registered trademarks of Handspring, Inc. © 2000, 2001, 2002 Handspring, Inc.