handspring

Springboard Module Testing

Application Note AN-01 Revision 2.00 Document No. 80-0124-00

Table of Contents

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 Asynchrnous 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 catagories of Springboard module development. In the first category, a developer may want to simply utilitize 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, etal.) 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 hardcoding 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 the 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/palmosswtechres.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.

More information on this module is available at: www.efig.com

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 punctiation 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 a 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;

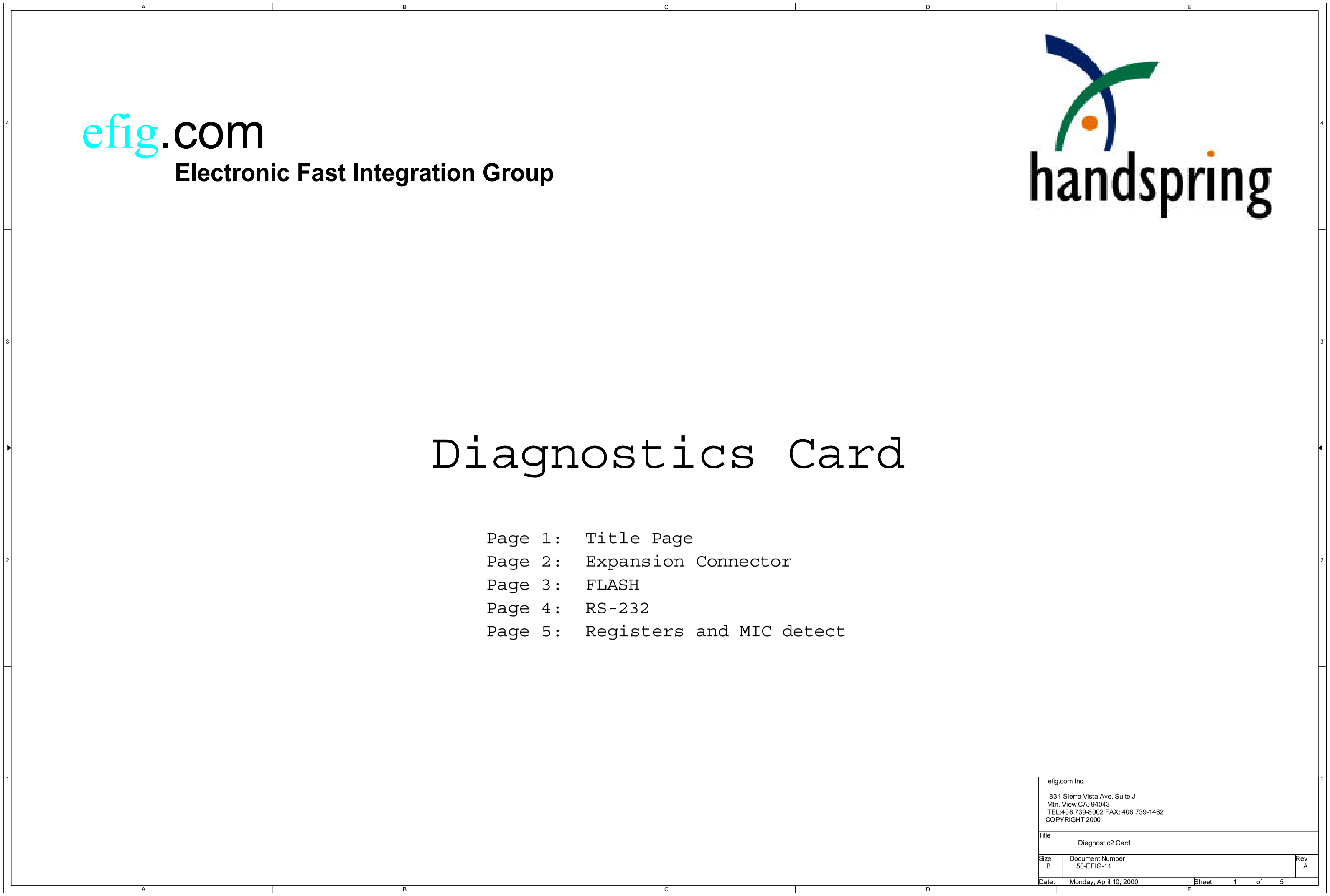
1. Download the SDK itself.
2. The HPG Tools themselves are also located on this page and are required to compile the SDK. The HPG Tools also require the Cygwin shell which is also located on this page.
3. The CardUpdaterMaker SDK requires a ROM image file. This image file is created by the Palm-MakeROM 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



efig.com
Electronic Fast Integration Group

handspring

Diagnostics Card

Page 1: Title Page
Page 2: Expansion Connector
Page 3: FLASH
Page 4: RS-232
Page 5: Registers and MIC detect

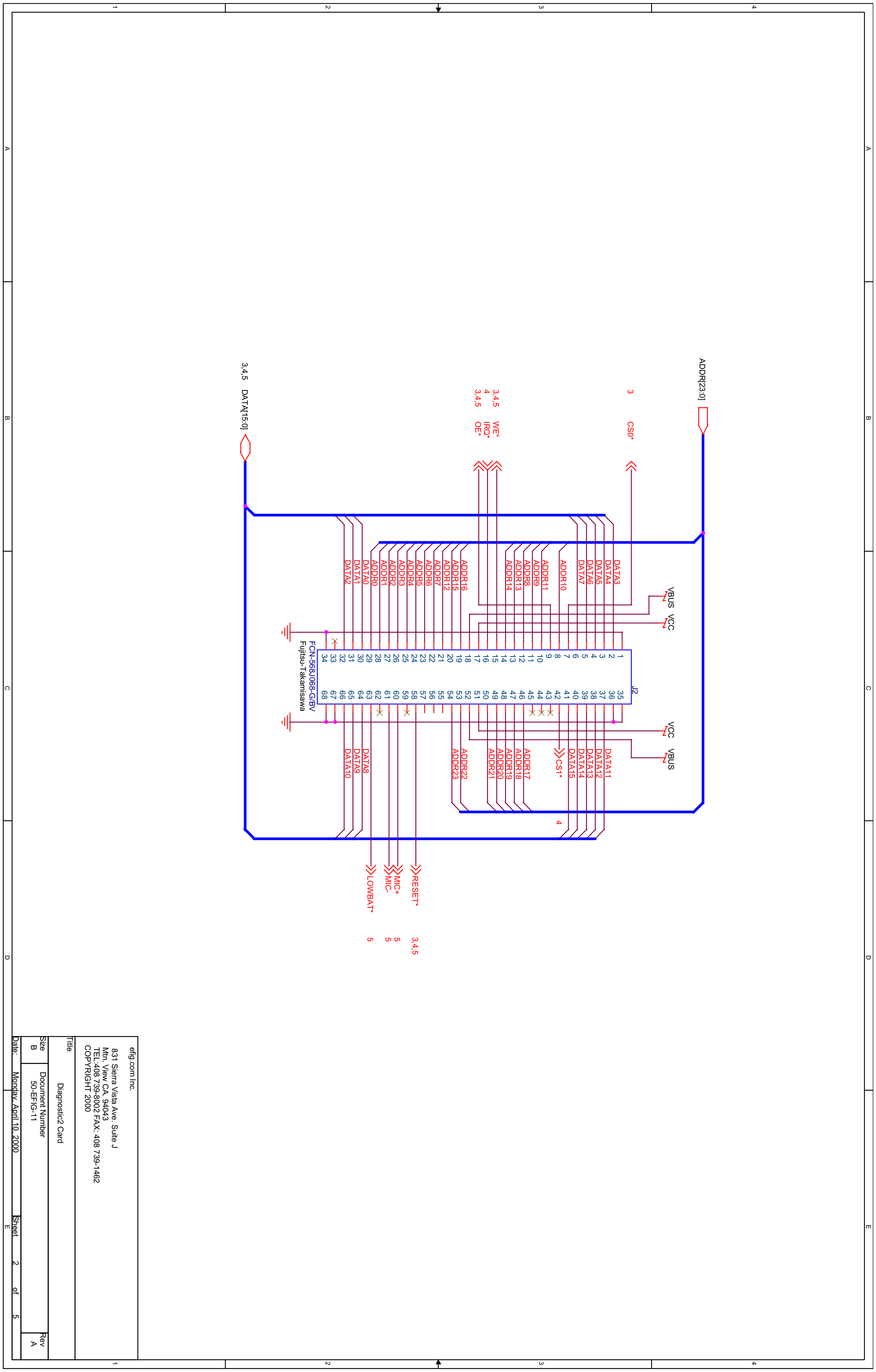
efig.com Inc.

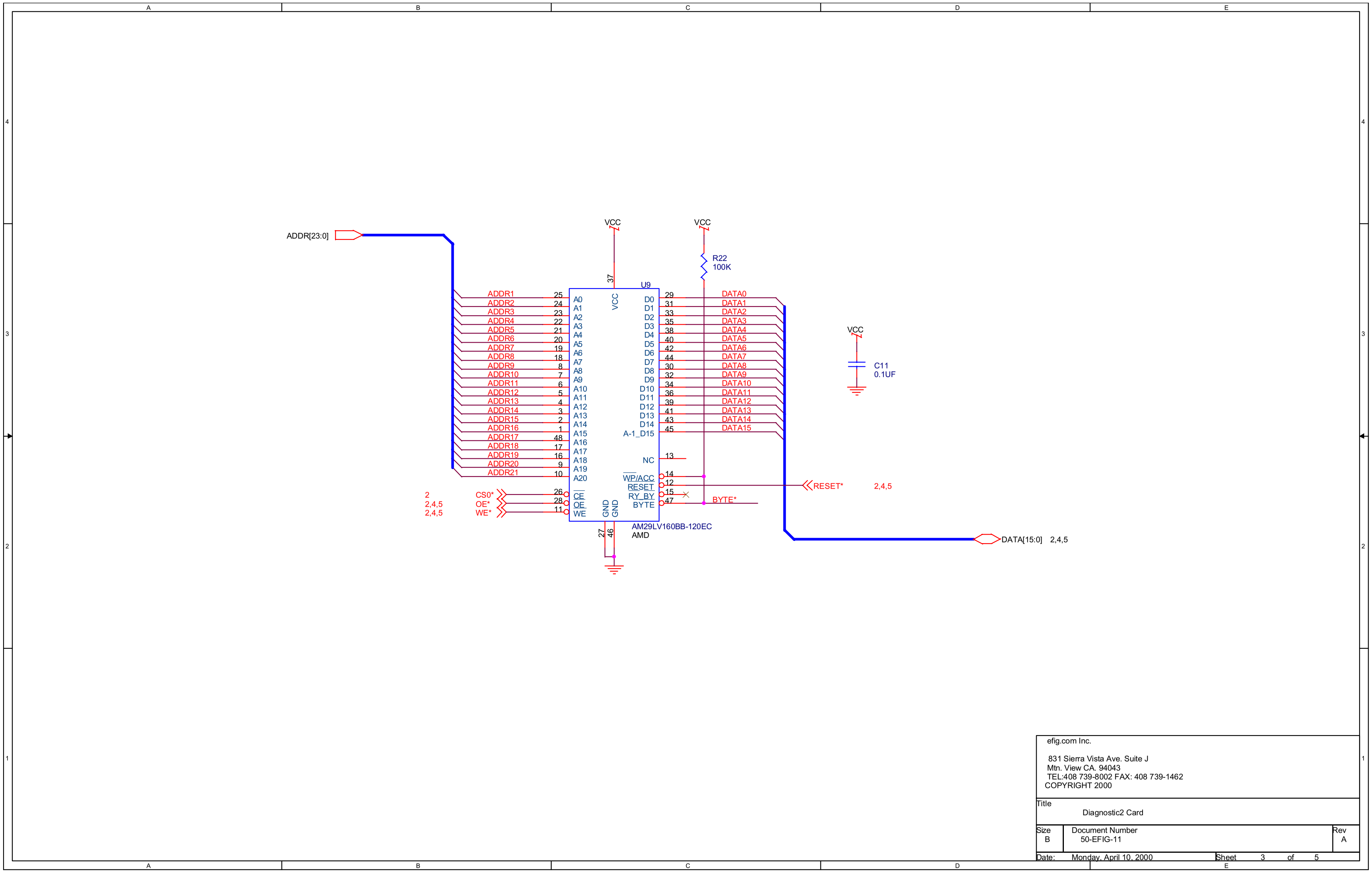
831 Sierra Vista Ave. Suite J
Mtn. View CA. 94043
TEL:408 739-8002 FAX: 408 739-1462
COPYRIGHT 2000

Title
Diagnostic2 Card

Size B
Document Number 50-EFIG-11
Rev A

Date: Monday, April 10, 2000
Sheet 1 of 5


ADDR[23:0]
3,4,5 DATA[15:0]
3 CS0*
3,4,5 WE*
4 IRQ*
3,4,5 OE*
VBUS
VCC
VCC
VBUS
J2
FCN-568J068-G/BV
Fujitsu-Takamisawa
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
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
DATA3
DATA4
DATA5
DATA6
DATA7
ADDR10
ADDR11
ADDR9
ADDR8
ADDR13
ADDR14
ADDR16
ADDR15
ADDR12
ADDR7
ADDR6
ADDR5
ADDR4
ADDR3
ADDR2
ADDR1
ADDR0
DATA0
DATA1
DATA2
DATA11
DATA12
DATA13
DATA14
DATA15
CS1*
4
ADDR17
ADDR18
ADDR19
ADDR20
ADDR21
ADDR22
ADDR23
RESET* 3,4,5
MIC+ 5
MIC- 5
LOWBAT* 5
DATA8
DATA9
DATA10
A
B
C
D
E
1
2
3
4
efig.com Inc.
831 Sierra Vista Ave. Suite J
Mtn. 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 2 of 5



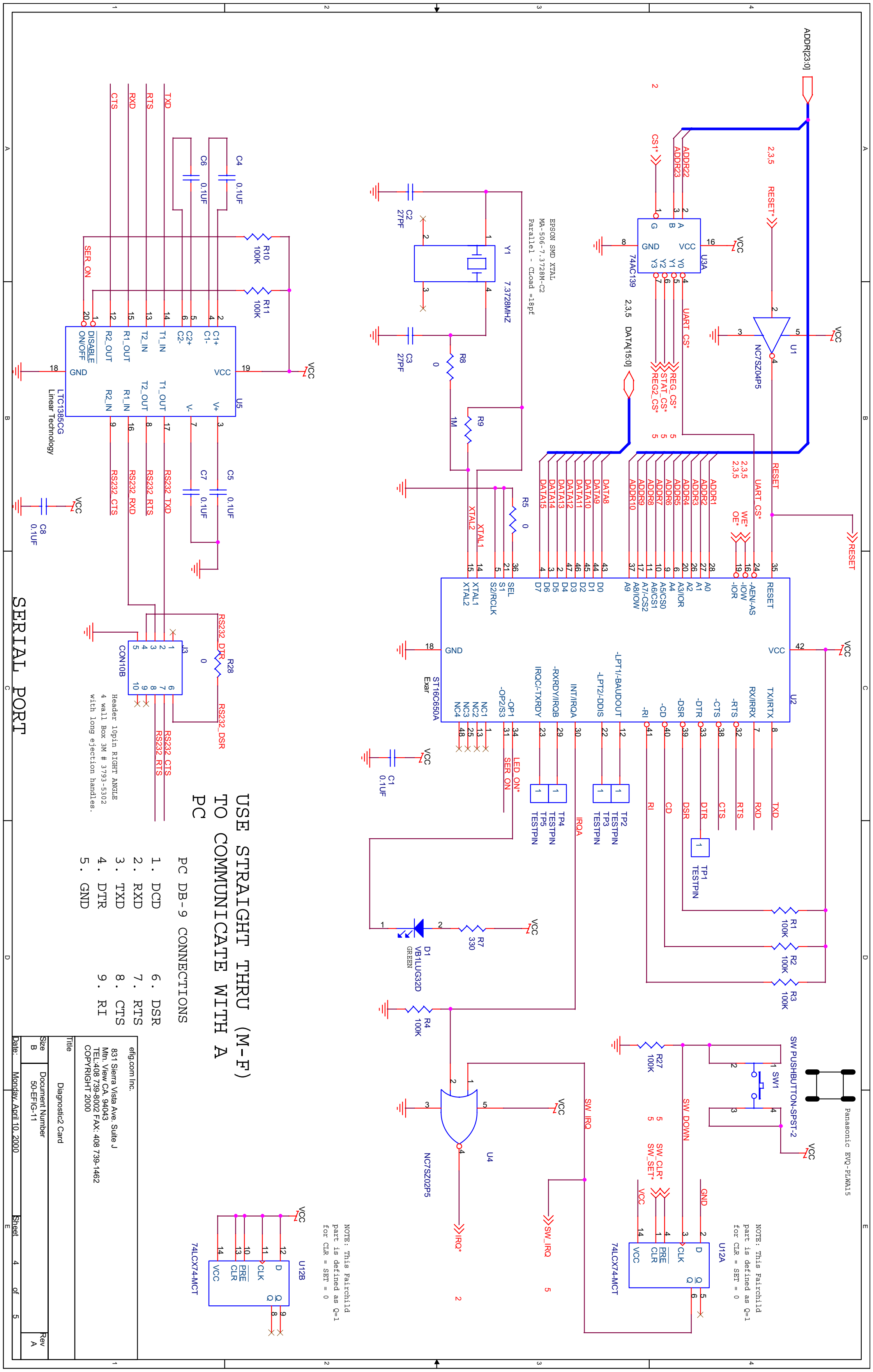
efig.com Inc.

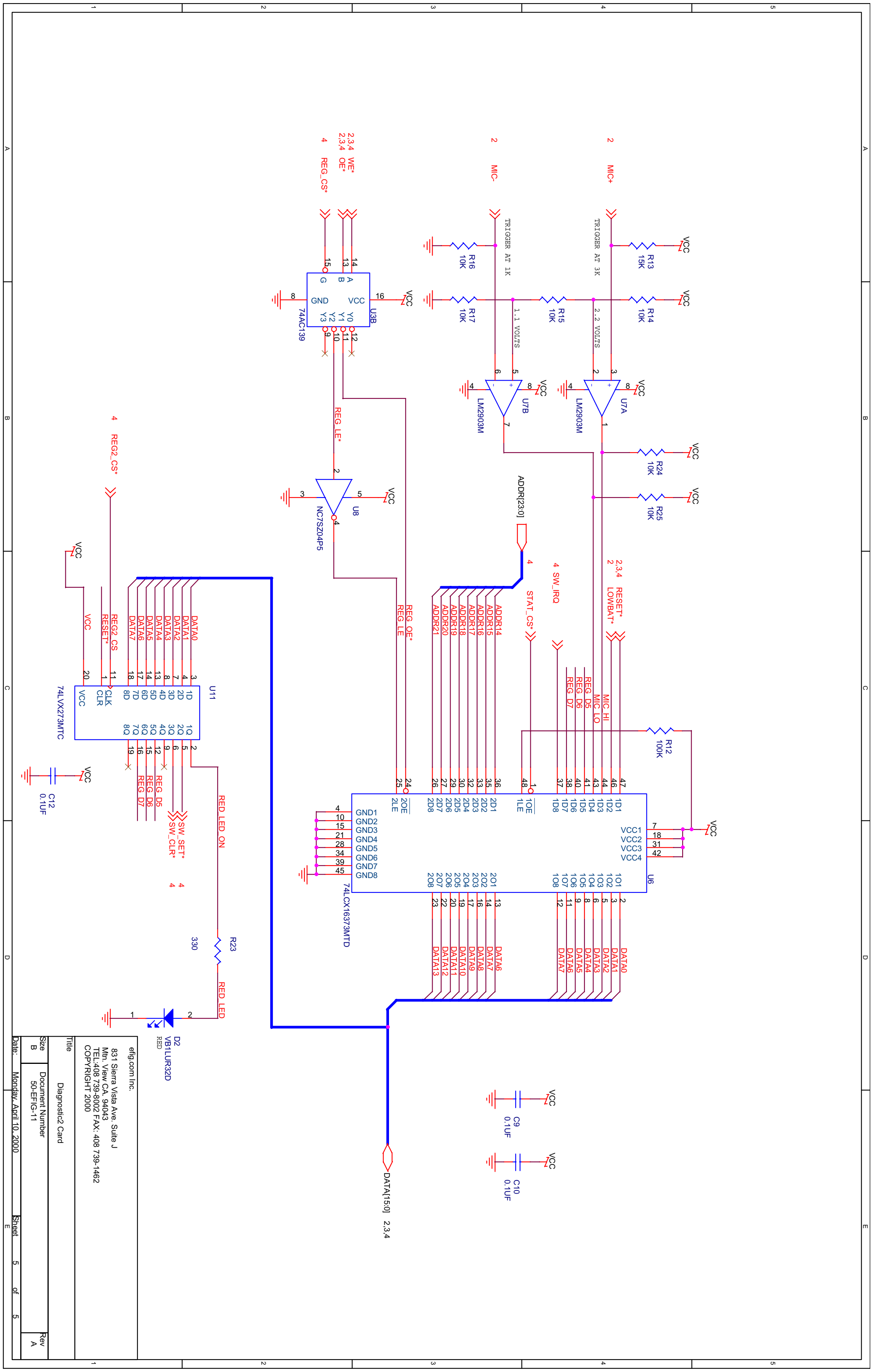
831 Sierra Vista Ave. Suite J
Mtn. View CA. 94043
TEL:408 739-8002 FAX: 408 739-1462
COPYRIGHT 2000

Title
Diagnostic2 Card

Size	Document Number	Rev
B	50-EFIG-11	A

Date: Monday, April 10, 2000 Sheet 3 of 5


ADDR[23:0]
RESET*
2,3,5
CS1*
ADDR22
ADDR23
U3A
74AC139
G
A
B
GND
VCC
Y0
Y1
Y2
Y3
UART_CS*
REG_CS*
STAT_CS*
REG2_CS*
U1
NC7SZ04P5
DATA[15:0]
RESET
UART_CS*
WE*
OE*
ADDR1
ADDR2
ADDR3
ADDR4
ADDR5
ADDR6
ADDR7
ADDR8
ADDR9
ADDR10
DATA8
DATA9
DATA10
DATA11
DATA12
DATA13
DATA14
DATA15
U2
ST16C650A
Exar
RESET
-AEN/-AS
-IOW
-IOR
A0
A1
A2
A3/IOR
A4
A5/CS0
A6/CS1
A7-CS2
A8/IOW
A9
D0
D1
D2
D3
D4
D5
D6
D7
SEL
S1
S2/RCLK
XTAL1
XTAL2
GND
VCC
TX/IRTX
RX/IRRX
-RTS
-CTS
-DTR
-DSR
-CD
-RI
-LPT1/-BAUDOUT
-LPT2/-DDIS
INT/IRQA
-RXRDY/IRQB
IRQC/-TXRDY
-OP1
-OP2/S3
NC1
NC2
NC3
NC4
TXD
RXD
RTS
CTS
DTR
DSR
CD
RI
IRQA
LED_ON*
SER_ON
TP1 TESTPIN
TP2 TESTPIN
TP3 TESTPIN
TP4 TESTPIN
TP5 TESTPIN
R1 100K
R2 100K
R3 100K
R4 100K
R5 0
R7 330
R8 0
R9 1M
D1 VB1LUG32D GREEN
C1 0.1UF
C2 27PF
C3 27PF
Y1 7.3728MHZ
EPSON SMD XTAL
MA-506-7.3728M-C2
Parallel - CLoad =18pf
U4
NC7SZ02P5
IRQ*
SW_IRQ
SW_DOWN
SW_CLR*
SW_SET*
SW PUSHBUTTON-SPST-2
SW1
Panasonic EVQ-PLHA15
R27 100K
U12A
U12B
74LCX74-MCT
D
CLK
PRE
CLR
Q
VCC
GND
NOTE: This Fairchild part is defined as Q=1 for CLR = SET = 0
U5
LTC1386CG
Linear Technology
C1+
C1-
C2+
C2-
T1_IN
T2_IN
R1_OUT
R2_OUT
DISABLE
ON/OFF
V+
V-
T1_OUT
T2_OUT
R1_IN
R2_IN
C4 0.1UF
C5 0.1UF
C6 0.1UF
C7 0.1UF
C8 0.1UF
R10 100K
R11 100K
R28 0
RS232_TXD
RS232_RTS
RS232_RXD
RS232_CTS
RS232_DTR
RS232_DSR
J3
CON10B
Header 10pin RIGHT ANGLE
4 wall Box 3M # 3793-5302
with long ejection handles.
SERIAL PORT
USE STRAIGHT THRU (M-F)
TO COMMUNICATE WITH A
PC
PC DB-9 CONNECTIONS
1. DCD 6. DSR
2. RXD 7. RTS
3. TXD 8. CTS
4. DTR 9. RI
5. GND
efg.com Inc.
831 Sierra Vista Ave. Suite J
Mtn. View CA. 94043
TEL 408 739-8002 FAX: 408 739-1462
COPYRIGHT 2000
Title
Diagnostic2 Card
Size
B
Document Number
50-EFG-11
Rev
A
Date: Monday, April 10, 2000
Sheet 4 of 5


WE*
OE*
REG_CS*
2,3,4
2,3,4
4
MIC-
MIC+
TRIGGER AT 1K
TRIGGER AT 3K
1.1 VOLTS
2.2 VOLTS
R13 15K
R14 10K
R15 10K
R16 10K
R17 10K
R24 10K
R25 10K
R12 100K
VCC
U3B
74AC139
A
B
G
GND
VCC
Y0
Y1
Y2
Y3
U7A
U7B
LM2903M
U8
NC7SZ04P5
REG LE*
REG2_CS*
ADDR[23:0]
RESET*
LOWBAT*
SW_IRQ
STAT_CS*
REG_OE*
REG_LE
ADDR14
ADDR15
ADDR16
ADDR17
ADDR18
ADDR19
ADDR20
ADDR21
MIC HI
MIC LO
REG D5
REG D6
REG D7
U6
74LCX16373MTD
1OE
1LE
2OE
2LE
GND1
GND2
GND3
GND4
GND5
GND6
GND7
GND8
VCC1
VCC2
VCC3
VCC4
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
DATA8
DATA9
DATA10
DATA11
DATA12
DATA13
DATA[15:0]
2,3,4
U11
74LVX273MTC
CLK
CLR
VCC
REG2_CS
RED LED ON
SW_SET*
SW_CLR*
R23
330
RED LED
D2
VB1LUR32D
RED
C9 0.1UF
C10 0.1UF
C12 0.1UF
efig.com Inc.
831 Sierra Vista Ave. Suite J
Mtn. 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 5 of 5

6. History

Date	Revision #	Description of changes
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.