



Table of Contents

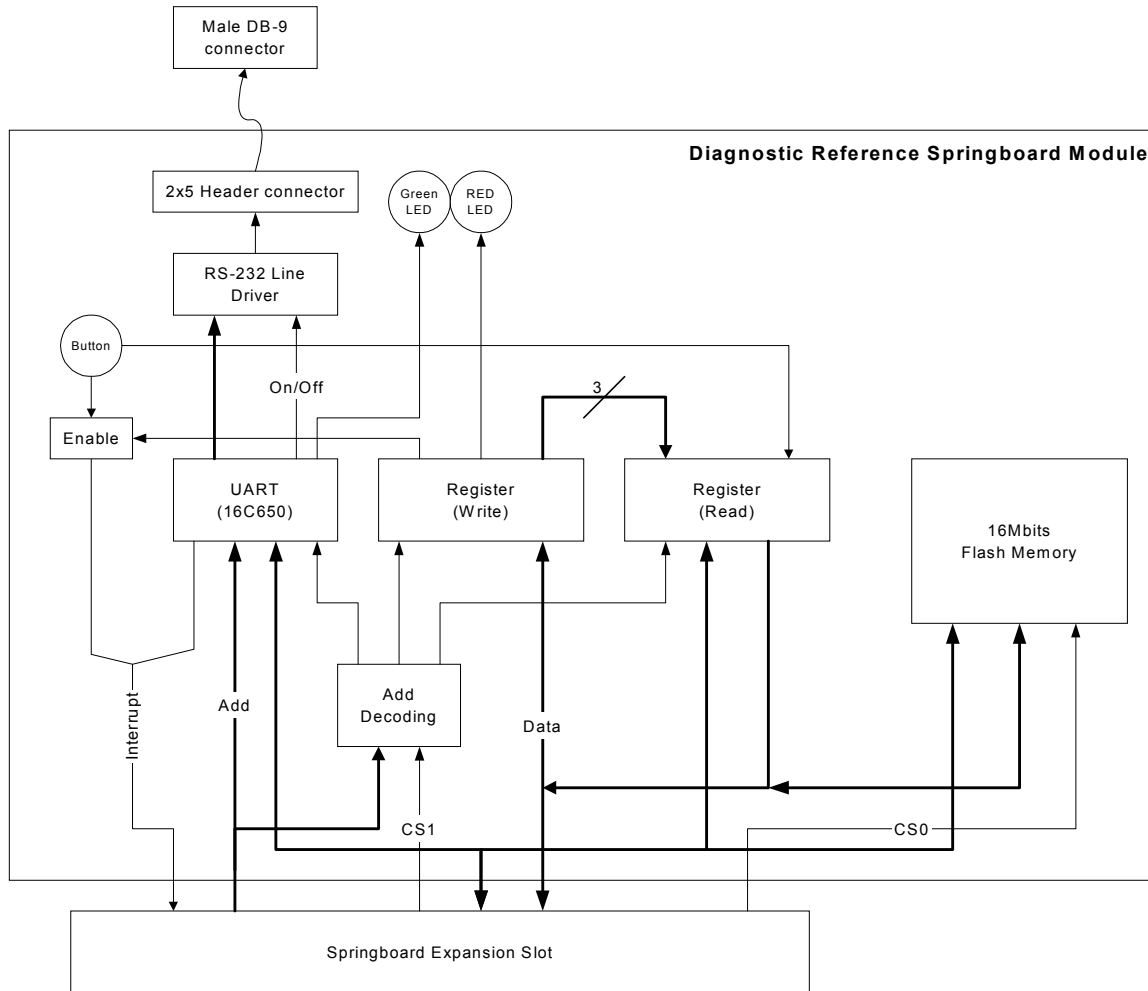
1.	Overview	1
2.	Hardware Description.....	2
3.	Software Description.....	5
3.1.	Project Builds	6
3.2.	Applications Built.....	7
3.3.	Applications Execute	8
4.	History	9

1. Overview

The Diagnostic Module is a hardware and software reference design kit for hardware and software developers to use as a reference when developing Springboard modules. This diagnostic module provides a detailed example of how to properly interface with the Springboard expansion slot and how to develop appropriate Springboard application software.

2. Hardware Description

The block diagram below shows the main components and interfaces of the Diagnostic Reference Module.



The Diagnostic Module's electrical hardware consists of a Flash memory device, a serial UART device, registers, and latches. These components are accessed via the Springboard slot to demonstrate the basics of the Springboard interface. The Diagnostic Module consists of:

- Address decoding logic to demonstrate how to map multiple devices and functions.
- Flash memory device to demonstrate how to interface memory to the Springboard expansion bus to store module-resident software applications.
- 16650 UART device to demonstrate how to interface and communicate with memory-mapped I/O devices on the Springboard expansion bus. The UART provides a demonstration of basic serial communication protocol with external devices such as a standard communications port on a PC or most RS-232 compatible communications ports.
- Registers to control the red LED and the interrupt enable of the push-button. This demonstrates glue logic interfaces, such as address decoding and register read-write controls.
- Latches to demonstrate how to implement control status such as microphone signal detect and software-controlled interrupt readings.

The address decoding is listed in the table below. All of the I/O devices below are mapped to the chip select 1, which starts at address location 0x29000000. Only Address lines A23, A22, WE* and OE* are used to decode the devices' enable/select.

Springboard signals					Device accessed
A23	A22	WE*	OE*	Address location	Device
0	0	X	X	0x290x xxxx	16650 UART
0	1	0	1	0x294x xxxx	Address Latch Write
0	1	1	0	0x294x xxxx	Address Latch Read
1	0	1	0	0x298x xxxx	Status Latch (read only)
1	1	X	X	0x29Cx xxxx	GPIO Register

The GPIO Register implements controls for the Red LED and controls for the push-button interrupt circuitry. It also serves as a register read-back for data lines D7-D5. The address and data it controls are listed in the table below.

GPIO Register (Address 0x29CX XXXX)	
Data Bit #	Definition
0	Red LED (0 = off, 1 = on)
1 - 2	Interrupt button enable/clear (01 = clear; 10 = set)
3	Unused
4 - 6	Set GPIO (4 - 6) (Readable from Status Register)
7	Unused

The SerialDiag application SB (store byte) command can be used to exercise and demonstrate the above controls.

SB 29C00000 0	Turns red LED off
SB 29C00000 1	Turns red LED on
SB 29C00000 2	Clears the interrupt from the push-button. (If the interrupt has been set, it must be cleared.)
SB 29C00000 4	Sets the interrupt

3. Software Description

The Diagnostic Module's software consists of a set of applications that demonstrate various features of the Springboard application. The software classification types and descriptions are listed below:

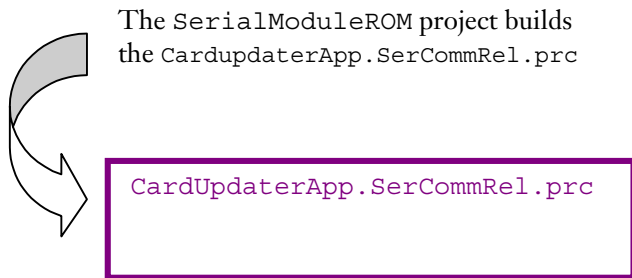
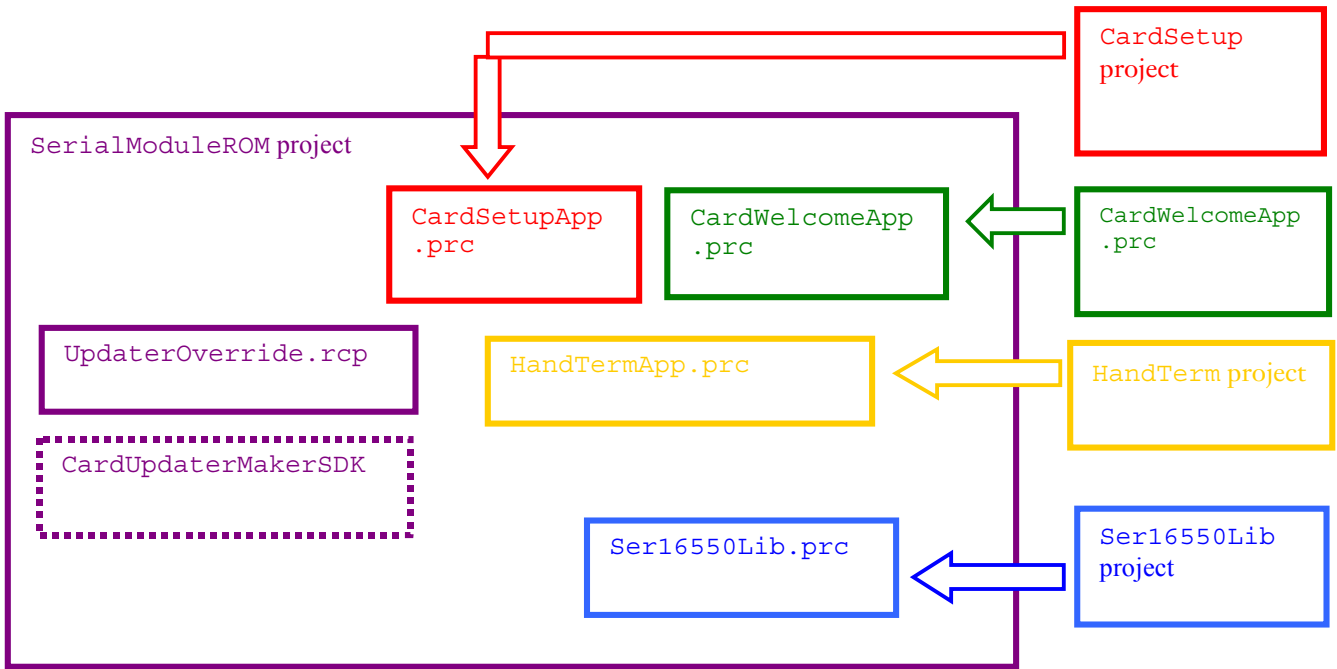
- The optional Card Welcome application that provides a welcome greeting and preference configurations.
- The required Card Setup application that facilitates the plug-and-play paradigm by installing the necessary software hooks, registering interrupt handlers and other event handlers, and installing necessary libraries and databases.
- The Hand Terminal "main" application sample provides the application's user interface and the functional processing. It is the main application of the Diagnostic module. The Hand Terminal application implements the basic features of terminal communication software.
- The 16550 UART serial library provides all the standard serial library API replacements. This library maps the serial functions into the hardware controls of the UART device.
- The Serial Diagnostic utilities provide statistical performance monitoring and data viewing. (This feature is enabled through conditional compilation directives.)
- The Card Updater application to build all the above applications and combine them into resource data to be programmed onto the card's Flash memory.

3.1. Project Builds

The SerialModuleROM project is a top-level project that builds all the individual card-resident applications into a card updater application that programs the card applications onto the card's Flash memory.

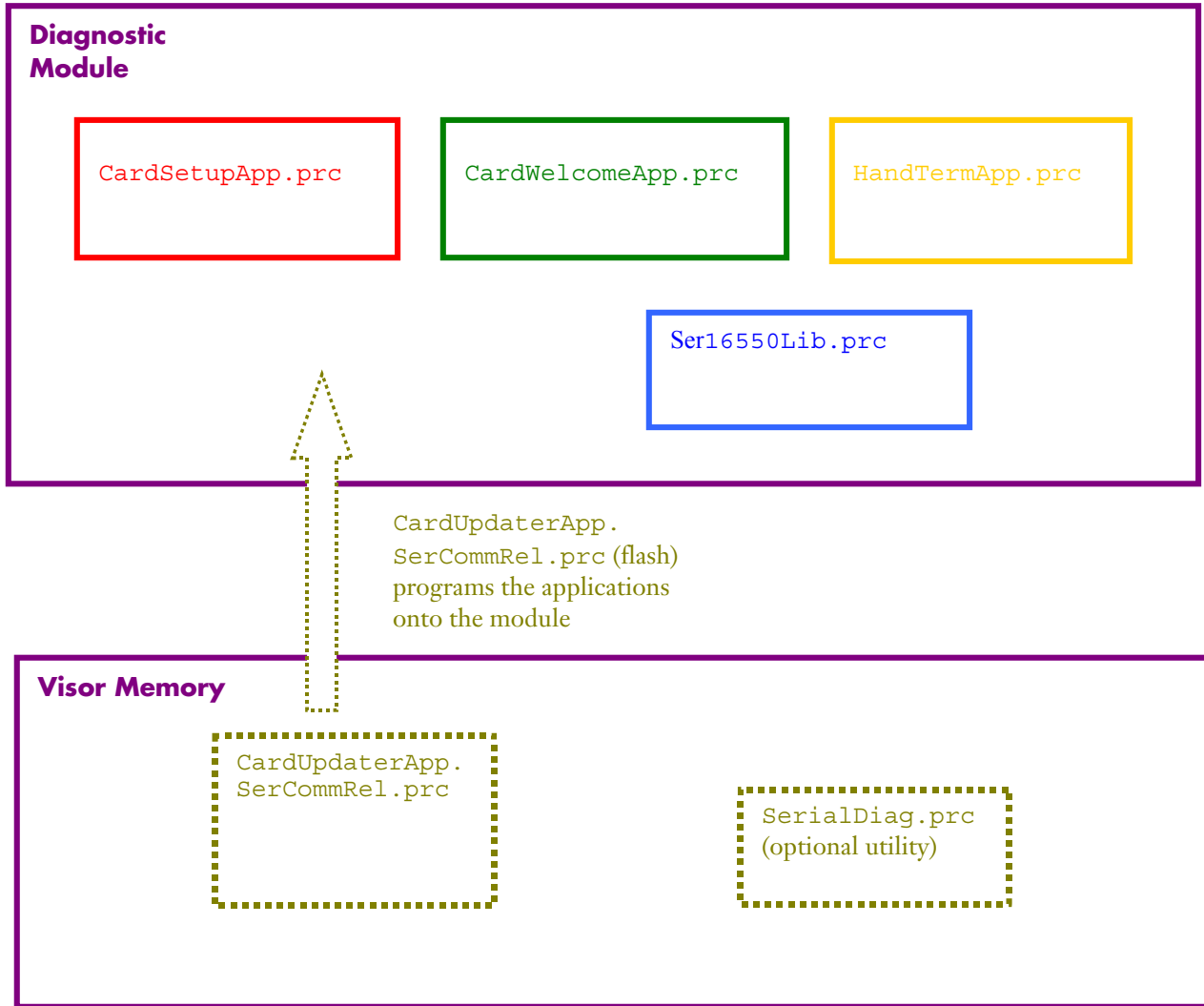
- The CardSetup project builds the CardSetupApp.prc.
- The CardWelcome project builds the CardWelcomeApp.prc.
- The HandTerm project builds the HandTermApp.prc.
- The Ser16550Lib project builds the Ser16550Lib.prc.
- The SerialDiag application is a Visor resident application and can be built independently.

The below diagrams illustrate the Diagnostic Module's software projects and the applications they build.



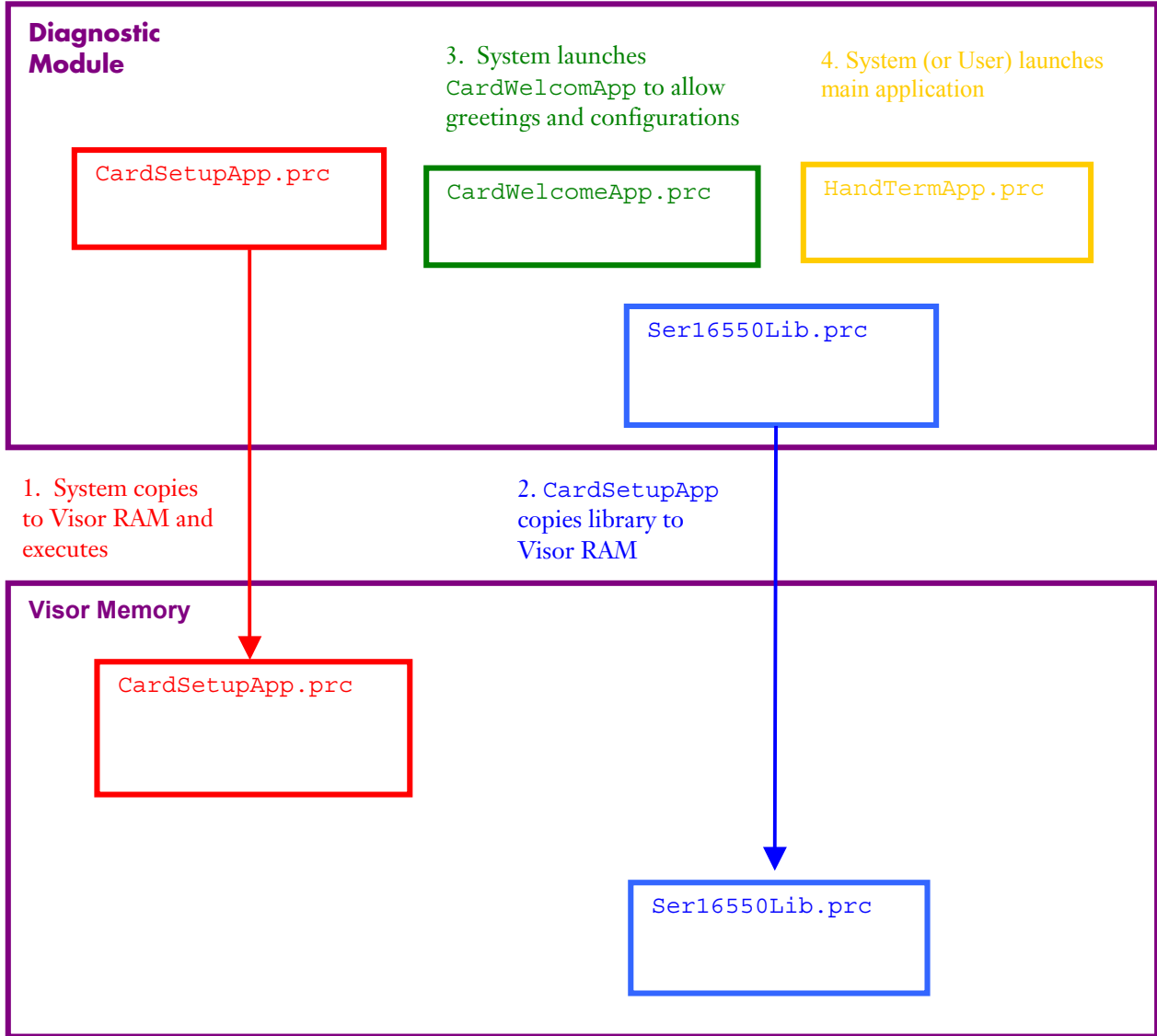
3.2. Applications Built

The diagram below shows the resident storage of the applications. The `CardUpdaterApp.SerCommRel.prc` is used to program the card-resident applications onto the card's Flash memory. After this operation is completed, the `CardUpdaterApp.SerCommRel.prc` is no longer needed, and can be deleted. The `SerialDiagApp.prc` is an optional monitor utility.



3.3. Applications Execute

The diagram below shows the application's execution relationship with respect to the Visor, to the card, and to each other.



4. History

Date	Revision #	Description of changes
11 Dec 00	1.01	Minor revision to show support for 16Mbit flash.
4 Apr 00	1.00	Initial release.

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