HSM/550 Technical Reference

Document Revision 0.6

A Complete Reference to Using & Programming the Dallas High Speed Micro Development System HSM/550

HSM/550

LIMITED WARRANTY

The information in this manual is subject to change without notice and does not represent a commitment on the part of Systronix, Inc. Systronix, Inc. makes no warranty, express or implied, for the use or misuse of its products, which are provided with the understanding that you, the user, will determine fitness for a particular application. Systronix assumes no responsibility for any errors which may appear in this manual. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use without the written permission of Systronix, Inc.

Systronix reserves the right to revise this documentation and the software and hardware described herein or make any changes to the specifications of the product described herein at any time without obligation to notify any person of such revision or change.

TRADEMARKS

Systronix is a registered trademark of Systronix Inc, INT_EL and Intel are registered trademarks of Intel Corporation, Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

Systronix®, Inc.
555 South 300 East #21
Salt Lake City, UT 84111
TEL: 801-534-1017
FAX: 801-534-1019
Internet: www.systronix.com
email: info@systronix.com

printed June 25, 1999

CREDITS and a BIT of RAMBLING FROM the AUTHOR

This manual was created using Corel WordPerfect 8.0 on an NT4 workstation with 256 Mbytes DRAM. I remember our first real Systronix computer - an IBM PC/AT with a 6 MHz 286 and 128 Kbytes of DRAM (wow!), $3000 plus $700 for a 30 Mbyte hard disk. It had a high-resolution (720x400) Hercules graphic card with an amber AMDEK monitor. Schematics and the HSM/550 circuit board were created with Accel EDA and Scott Kendall. Prototype build and production managed by Jared Wyckoff. Network Services by Christopher Robin. Postscript output was obtained from an Apple LaserWriter IINT, PDF output using Acrobat 4.0.

- Bruce Boyes, Systronix, Inc.

HSM/550 Technical Reference

Systronix, Inc.
Complete Solutions for Rapid Development of Embedded Control Systems

Document Revisions

1. 0.1 First version for HSM/550 rev A prototype boards. Open or unfinished issues are identified by "????" preceding a description of the issue.

2. 0.2 Extensive additions, should be pretty much complete. Need to have an overall view of the board with location of all jumpers called out so customers don't need to spend hours reading that tiny text looking for them.

3. 0.3 Corrections from Scott proofreading previous revision. Changes to example description.

4. 0.4 Corrections to and cleanup of head styles and table of contents generation.

5. 0.5 Added DS87C550 and HSM550 errata. Made Quick Start a level 2 heading. Added short section on DS1284 clock and calendar.

6. 0.6 1999-06-25 replacing any "WP" fonts with generic ones. WP fonts can neither be embedded in a PDF file nor loaded onto a Mac.

Table of Contents

HSM/550

Early Production Release
1999 June 02

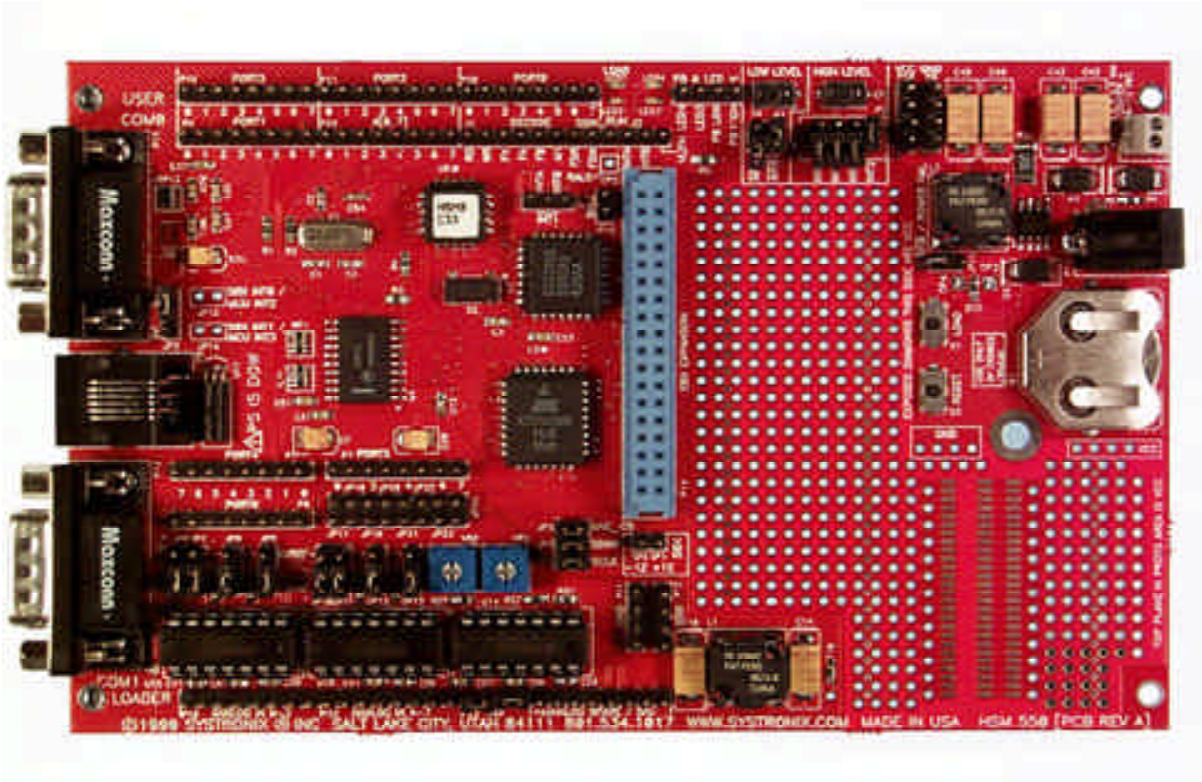
SYSTRONIX®

Dallas 87C550 High Speed Microcontroller Rapid Development & Prototyping System

The new HSM/550 is a ready-to-use DS87C550 development system.

Powerful Dallas High Speed Microcontroller core with 10-bit ADC and 8- or 16-bit PWM. Plus 12-bit DAC, analog buffers, clock & calendar, and more.

Notice the blue SBX expansion connector and generous prototype area.



Develop with the new 87C550 High Speed Microcontroller.

8 MIPs. 120 KBytes NVRAM. ADC, PWM, RS232, MicroLan, SBX. Prototype area.

Only $249.

Rev up your development with a warp-speed 8051 core, on-board ADC, DAC, PWM and more!

HSM/550 features eight channels of 10-bit ADC, one 12-bit DAC, four channels of PWM, and analog signal conditioning.

True zero wait-state performance. Easy program loading from a PC serial port. Dallas MicroLan/ iButton port for low-cost remote sensing & control.

Includes the new Systronix RAD51 8051 assembler and development environment! Device drivers and example programs with source code are included.

- All processor ports brought out to clearly-labelled headers.
- 120 KBytes NVRAM, zero wait states.
- Dual UARTs, dual RS232 ports
- Pushbuttons (high & low levels) and LEDs.
- Powerful serial loader& utility EPROM.
- 6-18 VDC power input, with low-noise, high-efficiency switching regulator.
- Generous prototyping area for SMT SOIC (wide and narrow), and through-hole DIP, SIP & ZIP
- Optional keypads and LCDs
- Standard 100mm by 160mm size
- Real technical support included!
- Latest info: www.systronix.com, or email to sales@systronix.com

What are Dallas High-Speed Microcontrollers?

Dallas High Speed Microcontrollers (HSMs) are high performance, low power, CMOS 8051 code-compatibles with a radical new processor core. HSMs complete an instruction cycle in only 4 clock periods instead of 12. Combine that 3X performance boost with clock speeds up to 33 MHz and you've got an 8.3 MIP CMOS controller!

The DS87C550 includes five external interrupts, an on-board watchdog timer, power-fail interrupt, dual UARTs, dual data pointers, built-in ADC, and power-conservation options. For data, contact Dallas Semiconductor at 972-371-4000 or www.dalsemi.com, or follow the links from www.systronix.com.

True 32 MHz Zero Wait-State Performance

The High Speed Micros require faster memory and I/O circuitry, and have much faster strobe slew rates. HSM/550 is rigorously designed to meet all manufacturer's timing requirements over worst case temperature and power variations, *with no "wait states"*. HSM/550 uses a 32 MHz crystal in order to utilize readily available 55 nsec SRAMS and to tolerate heavy loads on the address/data bus.

60 KBytes each of Code and Data plus 4 KByte I/O space

The High Speed Micros have the same 16-bit address space as 8051s, for up to 64 KBytes each of code and data. HSM/550 delivers with a full complement of memory: 60 KBytes each of code and data (both are nonvolatile), and a 4 KByte memory mapped I/O space.

I/O Options

The industry standard 8-bit SBX "mezzannine bus" interface is an easy way to plug on additional memory mapped I/O from dozens of vendors, or create your own with our SBX prototyping board. The Systronix SBX1 board supports parallel interface LCDs and matrix keypads, and 24 bits of rugged bidirectional digital I/O.

Smart Loader/Demonstrator EPROM

The powerful Systronix auto-bauding loader does much more than program HEX files into the development board's NVRAM. You can read and write all controller registers, internal data and external data memory, set stretch cycles,test interrupts, and more. You can peek and poke all memory-mapped I/O space - very handy for testing peripherals you've added. All of this can be done manually or via script files sent from an RS232 serial port of ANY computer - Wintel, Mac, Linux, SUN, whatever. All you need is a basic communications program.

Includes new Systronix RAD51 IDE and 8051 Assembler

HSM/550 includes the new Systronix RAD51 Integrated Development Environment (IDE) and 8051-family assembler. Requires Windows 95/98/NT.

How do I order?

Please refer to our *Product Matrix & Price List* and *Order Form* (available on our web site) for detailed option and ordering information. Our web site will always have the newest information on released products. Or call us!

SBX1 LCD and Keypad Option:

This feature-packed SBX board includes a 16-pin latching header for an LCD, a 4x5 keypad decoder, 24 bits of bidirectional I/O capable of sinking 150 mA, and a piezo buzzer. The digital I/O header is a standard Opto-22 type 25x2 for easy industrial I/O buffering.

Price (preliminary- subject to change):

HSM/550 $249 (single), $239 (2+), $??? price on higher quantities. SBX-1 LCD, keypad and I/O board: $??? SBX-Proto board with decoder/strobe PLD $???.

TECHNICAL DETAILS

Microcontroller Socketed PLCC68 Dallas DS87C550-33 MHz with a 32 MHz crystal. All ports are presented on labelled headers.

Memory 60 KBytes of code and 60 KBytes of data in NVRAM. 4 KByte I/O space. Replacable lithium battery.

Power Unregulated 6-18 VDC input from a 5.5x2.5 mm jack. Efficient switching regulator is reverse-polarity, short-circuit and over-temperature protected. 5V @ 500 mA available for user. Four layer board with isolated analog power plane and pi-filtered analog power.

Serial I/O Two RS232 serial I/O, one for each UART.

Analog I/O Eight channels of 10-bit ADC with op-amp buffers. Precision 2.048V reference. Input range of each channel is jumper selectable 0-2.048, -2.048 to +2.048, or 0 to 4.096. One channel of 12-bit analog output, op-amp buffered with a range of 0-4.096 volts (resolution of 1 mV per count). Two free opamps with adjustable gain of 1 to 11. Opamps are standard quad type in DIP sockets.

PWM and Capture/Compare Four channels of 8-bit PWM can be cascaded to form dual 16-bit PWM. Timer2 has multiple capture/compare modes.

Digital I/O The C550's ports 4, 5, and 6 are presented on labelled headers. SBX cards provide additional I/O options.

LEDs and Switches Two pushbuttons (low and high levels) and two LEDs are provided for experimentation. The pushbuttons can drive one or more of the C550's interrupts.

Clock & Calendar DS1284 with lithium battery (also provides NVRAM backup). Calendar or interval interrupts can be jumpered to the controller.

Expansion 8-bit SBX connector with up to 16 decoded addresses and two interrupts. Dallas MicroLan/iButton port for low-cost remote sensing & control. Memory mapped I/O space is decoded at FAXXH and FBXX, as are read and write strobes at FDXXH.

Easy Program Loading Serial program loading of HEX files initiated by on-card pushbutton. The auto-bauding serial loader is only active in LOAD mode. In RUN mode it is inactive, giving your program complete control of all controller resources.

Size Standard 100x160 mm single Eurocard size, hundreds of enclosures available (some stocked by Systronix) including RF shielded, NEMA rated, etc.

Environmental Commercial temperature 0 to 70 deg C.

Support & Warranty Friendly technical support. One year warranty against defects, and fast turn-around on repairs.

All systems include:

- Printed user manual & technical reference
- Wall cube power supply
- Systronix RAD51 assembler & IDE for Windows 95/98/NT
- Sample programs in assembly code and Keil C

SYSTRONIX®

555 South 300 East #21, Salt Lake City, Utah, USA 84111
Tel:+1-801-534-1017 Fax:+1-801-534-1019 www.systronix.com

revised 1999 jun 02 bab

HSM/550 - READ THIS FIRST

If you really hate reading manuals, then go ahead and hook up the board - HSM/550 COM1 is used for program loading (it's marked on the board silkscreen). Use a *straight-thru* serial cable (*not a null modem* cable). Run a simple terminal communications program such as Hyperterminal, set for 19200 8/N/1, XON-XOFF. Send HSM/550 a carriage return (0DH), usually by pressing the Enter key. HSM/550 should respond with a logon screen. If you get stuck, come back here and/or check the troubleshooting section or the FAQ at www.systronix.com.

Thank you for purchasing the **High Speed Microcontroller (HSM) Dallas DS87C550 Development System!** HSM/550 was developed in conjunction with Dallas Semiconductor and customers like you to be the best DS87C550 development board available. The current revision of the circuit board is "A" (in the lower right top corner of the board). HSM/550 is specifically designed to support the special I/O features of the DS87C550. In particular, HSM/550 has a low noise power supply with an additional PI-filter for the analog circuitry. HSM/550 is a four-layer circuit board with split analog and digital planes. We've added low noise analog buffer circuits to the DS87C550 inputs and the Linear Technology 12-bit DAC output.

Errata & Anomalies

HSM/550 PCB Rev A

SBX Card Covers Load/Reset Switches

Load/Reset switches unreachable if an SBX card is installed. Oops. Will be fixed in Rev B boards, they will be moved to the right edge of the board below the battery.

Write-only I/O Space Can't Verify

This is not exactly errata, it's more of an unavoidable quirk. The loader W command writes to memory or I/O space using the movx instruction. It also attempts to verify what was written by reading at the same address. This will find memory problems or some memory mapped I/O register problems. However, some memory mapped devices are write-only, or for other reasons don't allow you to read the same data you wrote. In that case you'll see a loader warning about a verify failure. That's the quirk - any I/O space verify failures may not be real, depending on your specific I/O device.

DS1284 shuts down if Lithium Battery > 3.7 Volts

This is not really an errata, it's just the way the DS1284 tells if it should enter backup mode. If Vbat is greater than 3.7 volts, the DS1284 concludes that Vbat > Vcc and it should enter battery backup mode. Therefore, if you replace the Lithium battery with source higher than 3.7 volts, or the battery

itself becomes 'overcharged', you will not be able to access the DS1284. We encountered this in initial prototypes of HSM/550 in which the power control circuitry charged the Lithium battery with 100 uA. This causes no harm, but after about 200 power-on hours, Vbat reached 3.9 volts and the DS1284 could not be accessed. Attempts to read a location such as FEXXH returned the value XX, which is indicative of reading a floating data bus.

DS87C550 Controller Die Rev A3

Revision A3 may be identified by the date/revision brand yywwA3, where yy and ww are the year and workweek of manufacture, respectively. This errata sheet is valid only when used in conjunction with the most current version of the data sheet available from Dallas Semiconductor via the Internet.

Changes from A2 to A3 were limited to manufacturability enhancements, and no changes to device functionality or errata were made.

1. The Stop mode current may be as high as 1 mA on some devices.

Work Around: None. This erratum will be fixed in future revisions of the device.

2. Extended (stretch) MOVX instructions do not operate correctly when the MD2, MD1, and MD0 bits are configured for 10 or more machine cycles. Lesser values will operate as described.

Work Around: None. This erratum will be fixed in future revisions of the device.

3. External interrupts (INT2, INT3, INT4, and INT5) will not cause the device to exit Idle mode.

Work Around: Use external interrupts 0 and 1 to exit Idle mode.

4. The capture/compare functions of Timer 2 will not operate in Idle mode

Work Around: None. This erratum will be fixed in the next revision of the device.

5. The processor may experience erratic operation if Stop mode is initiated when the device is operating in 1 or 2 clock per machine cycle mode.

Work Around: Switch to 4 clock per machine cycle mode (CD1, CD0 = 10b) before entering Stop mode.

6. The analog-to-digital converter has an offset of approximately +5 LSb.

Work Around: None: This erratum will be fixed in the next revision of the device.

7. Serial port status bits RB8 (SCON0.2 and SCON1.2) and FE (SCON0.7 and SCON1.7) can not be modified for at least 1 serial port clock period after being set by internal hardware. The documentation implies that these bits can be modified at any time. This erratum applies to both serial ports.

Work Around: The incorporation of a software delay can ensure a successful modification of the above mentioned bits. These bits are modified by internal hardware at the same time the RI bit is set, prior to entering the serial port interrupt service routine. After entering the interrupt service routine, incorporate the following delay, based on the serial port baud rate, before modifying the bit(s):

$$\text{Delay} = \left[\frac{1}{\text{baud\_rate} \bullet 16} \right] \text{seconds}$$

This delay is most easily generated by incorporating a short software loop near the beginning of the serial port interrupt service routine. For example,

```
; Start of serial port interrupt service routine, after software
; has determined that the interrupt was generated by the RI flag.

      mov r0, #delay_count
   sp_delay:
      djnz r0, sp_delay
```

Where `delay_count` $= \text{int}\left[\frac{\text{Osc.Frequency}}{\text{baud\_rate} \bullet 192} \right]$

In general, this delay will be very short, having a minimal impact on software performance. For a microcontroller operating at 11.0592 MHz with its serial port configured for 19200 baud, the delay would be 3.25 ms, resulting in a delay_count of 03h.

8. SBUF0 and SBUF1 cannot be modified for at least 1 serial port clock period after their respective TI bits are set by internal hardware. The documentation implies that these bits can be modified at any time. This erratum applies to both serial ports.

Work Around: The incorporation of a software delay at the start of the interrupt service routine ensures a successful loading of the above mentioned registers. After entering the interrupt service routine, incorporate the following delay, based on the serial port baud rate, before modifying the TI bit:

$$\text{Delay} = \left[\frac{1}{\text{baud\_rate} \bullet 16} \right] \text{seconds}$$

This delay is most easily generated by incorporating a short software loop near the beginning of the serial port interrupt service routine. For example,

```
;Start of serial port interrupt service routine, after software
;has determined that the interrupt was generated by the TI flag.

      mov r0, #delay_count
   sp_delay:
      djnz r0, sp_delay
```

Where `delay_count` $= \text{int}\left[\frac{\text{Osc.Frequency}}{\text{baud\_rate} \bullet 192} \right]$

In general, this delay will be very short, having a minimal impact on software performance. For a microcontroller operating at 11.0592 MHz with its serial port configured for 19200 baud, the delay

would be 3.25 ms, resulting in a delay_count of 03h.

9. SCON0 and SCON1 cannot be modified while their respective transmitter circuitry is active. This is most likely to be an issue when attempting to clear the RI flag during full-duplex serial port activity. This erratum applies to both serial ports.

Work Around: This erratum can be overcome with two simple modifications. First, software should check the state of the RI after attempting to clear it. If no transmit activity was in progress, the bit will have been cleared and software can proceed. If the bit was not cleared, then the software should repeat the attempt until successful. The following code fragment illustrates the procedure:

```
ri_loop:
   clr  RI
   jb   RI, ri_loop
```

The second condition avoids the possibility of a receiver overrun while waiting to write to SCON. This is easily remedied by adding a character-pacing requirement to the incoming data stream. Specifically, the following time delay should be inserted between characters in the incoming data stream to avoid a receiver overrun during the preceding code example:

$$\text{Delay} = \left[\frac{1}{\text{baud\_rate}} \right] \text{seconds}$$

The same effect can be achieved by configuring the transmitter to send 2 stop bits per character.

Terminology in This Manual

We use a fixed pitch font to represent what you see or type on your PC:

```
C:\HSM550\ASM BLINK
```

☞ We use the pointing hand to call attention to something worthy of special note, such as a common pitfall or interesting feature of HSM/550.

What You Need to Use HSM/550

✓ To power up the HSM/550 board, you need a 6-12 VDC unregulated source such as the Systronix #5003 6VDC 800mA power cube or the heavy-duty #5007 12VDC 1000 mA power cube. If you use your own cube, be sure the center terminal of the 5.5 x 2.5 mm jack is positive 6 to 12 volts DC. The sleeve is negative. An AC power cube could damage HSM/550.

To connect HSM/550 to your PC serial port, you need a **straight-through cable from your PC** to a DB9 female, to mate with the DB9 male on HSM/550. Systronix #9210 serial adapter kit contains a 6 foot/ 2 meter DB9 extension cable, a DB9 to DB25 adapter, and DB25 and DB9 gender changers. DO NOT USE A NULL MODEM CABLE. **COM1 of HSM/550 is the loader serial port.**

✓ To download a program to HSM/550, and communicate with the HSM/550 serial loader, you need

any serial communications program such as Windows Hyperterminal. You can use *any* computer, not just a PC-compatible, as long as it supports RS232 communication.

✓ To assemble a program for use on HSM/550, you need an 8051 assembler such as the Systronix A51 assembler included with HSM/550. Or you can use any 8051 family development tool, running on any computer platform of your choice, as long as it generates a standard Intel HEX file. Binary file loading is not supported by HSM/550.

✓ To understand the High Speed Microcontroller family, you need the Dallas High Speed Microcontroller data sheets and application notes, available from www.systronix.com, www.dalsemi.com, or by calling Dallas Semiconductor at 972-371-4000. We've included the DS87C550 data sheet in PDF format on your HSM/550 disk.

Quick Start - Step by Step

What you need

If you've lost any of the accessories provided with HSM/550, replacements are available for a modest charge. Call us at 801-534-1017 or email to HSM550_sales@systronix.com.

1) HSM/550 board, with the power cube (included with all orders)
2) Serial cable to your PC (you provide this or order our serial adapter kit) and terminal software such as Windows95 HyperTerminal (you provide this).
3) Systronix A51 assembler (provided on HSM/550 disk or available at www.systronix.com)
4) Text editor which can save ASCII text. Windows Notepad or Wordpad are OK.
5) Clip leads to connect the LED test point to a processor port pin. Clip leads with a .025 square receptacle or a micro hook on each end are ideal. We'd like to include these with HSM/550 if we can find a good source at a reasonable price.
6) Jumpers to connect HSM/550 interrupts to the pushbuttons (provided with HSM/550)
7) DS87C550 data sheet (provided on HSM/550 disk or available at www.systronix.com or www.dalsemi.com)

Step-by-Step Procedure

1) HSM/550 example files are in the self-extracting archive HSM5_XMP.EXE.Copy this file to a working directory on your hard disk, such as HSM, and expand it by typing

```
C:\HSM\HSM5_XMP
```

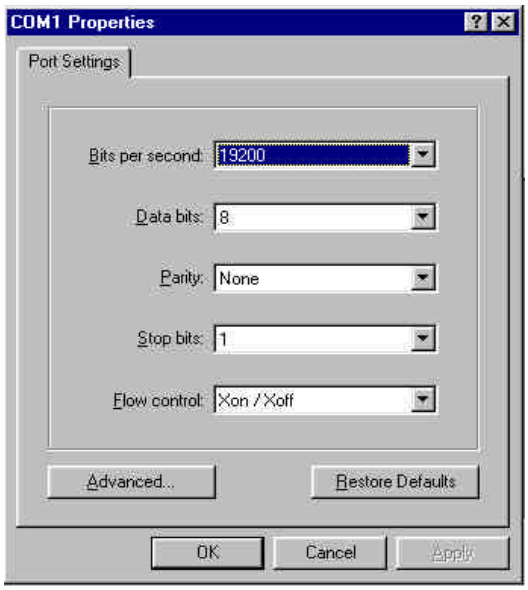
The files should expand without error messages. If you get a corrupted file error message, then the disk has probably been exposed to a strong magnetic field in shipment. Try reading it on another computer. The unzip in Norton File Manager or WinZip will also open this file. If all of this fails, your disk is toast. The quickest way to get new files is to load a new copy from the HSM files area of our web site at www.sysronix.com, or FTP it from ftp.systronix.com.

2) Connect the power cube to the HSM/550 P17 power jack. After application of power, the green RUN LED will be lit. Press the LOAD switch and hold it for more than 300 msec. The red LOAD LED should be on. If neither LED is lit, the board is not receiving power. Check for proper power polarity - the center of the P17 power jack is positive.

3) Connect a DB9 female straight (*not null modem*) cable from your PC serial port to HSM/550 COM1 (near the prototype area). HSM/550 expects TXD on pin 2 and RXD on pin 3 of the serial port cable. This sets up the PC as "DTE" (Data Teminal Equipment) and HSM/550 as "DCE" (Data Communications Equipment) in RS232 parlance. A kit of adapters and a DB9 extension

cable is available from Systronix at 801-534-1017 or www.systronix.com.

HSM/550 COM1 is the added UART of the Dallas 87C550, and is used by the serial loader for program loading. HSM/550 COM0 is the normal UART available in most every 8051, and is not used at all by the serial loader. HSM/550 must use COM1 for loading but you can use any available PC COM port.



4) Start a terminal program such as Hyperterminal, set the communication parameters for the baud rate you desire (typically 19.2 kbaud). Set other communication parameters to "direct" connection (not through a modem), 8 data bits, no parity, 1 stop bit, Xon/Xoff flow control. A Hyperterminal dialog box is shown to the right. Be sure to set the terminal software COM port to the correct port on your PC. Save these settings. Now open a connection in the terminal software, usually with a "connect" command or button.

5) Wait a second, to give the terminal program time to establish the connection to HSM/550. Now press and hold the LOAD button for more than half a second. When you release the button, the red LED should remain lit. If, instead, the green LED lights, you need to hold the LOAD button longer before releasing it. Now press your PC's ENTER key to give the loader a carriage return to use as its auto baud rate character.

☞ Note that your board may have a factory test program left in it, so don't be alarmed if it emits serial output in RUN mode, even before you load any program of your own.

If your PC serial port is connected through a straight cable (TXD and RXD are not swapped), and you press the LOAD button on HSM/550, and then press the ENTER key on your PC, you should see the loader prompt. It will look something like this. The exact appearance may vary with your specific loader version.

```
Unified HSM HEX LOADER Rev D.04 (4/08/99)
(C)1996-1999 SYSTRONIX, INC.
Type "?" for command help
```

6) If you enter a question mark for help, you should see something like this (the exact appearance will depend on your loader version):

```
550> ?
---------- Loader Commands ----------------------------------------
  L                          ;load intel HEX file
  D [start [end]]            ;dump HEX file
  V                          ;verify HEX file w/memory
  T                          ;toggle HEX file echo
  I                          ;toggle all external interrupt enables
  C [start [end]]            ;calc CRC-16
  E                          ;erase xdata to 0FFH
  F value [start [end]]      ;fill xdata with value
  G                          ;get port values
  P val0 val1 val2 val3      ;put valx to portx...
                             ;...except TXD1, RXD1, RD, WR of P1 & P3
  WX adr val                 ;write val to adr in xdata or I/O space
  WI adr val                 ;write val to adr in internal data space
  WR adr val                 ;write val to adr in SFR space
  RX adr                     ;read at adr in xdata or I/O space
  RI adr                     ;read at adr in internal data space
  RR adr                     ;read at adr in SFR space
  M                          ;Memory test with exhaustive patterns
  A                          ;Address line test
  X value                    ;show/change movx stretch value (0-7)
------------------------------------------------------------------------

Press any key to continue . . .
  WD val                     ;write val to system DAC (550 only)

  Memory: code=0000-EFFFH, xdata=0000-EFFFH, I/O=F000-FFFFH
  All values are entered and displayed in hexadecimal format
  memory errors display as {address}:{should be}/{actual}

Loader has detected DS87C550 processor
550>
```

7) Try sending one of the HEX files we've included to HSM/550. To send a HEX file to HSM/550, type an "L" for LOAD, and send the HEX file from your terminal software. If the loader receives the file with correct checksums, you will get an "OK" response. If there are load errors, the loader will tell you as they occur.

It's tempting to jump into a complex application right off the bat, but *please run one of the supplied sample programs first*. This will verify that your PC, cable, hardware and installed software are all working together

If you are using assembly code, we've included example files "HELLO33.HEX", "DS1820.HEX" and others, along with the .ASM source code. Check our web site www.systronix.com for other samples. For a quick test of your system, load the HELLO33.HEX file. This program continuously prints "hello from COMX" to both UARTs assuming a 33 MHz crystal. HELLO22 and HELLO14 are versions for 22.1184 and 14.7456 MHz, respectively.

There are many differences between the DS87C550 and other 8051s. In particular, common Special Function Registers (SFRs) and the interrupt vector table are different. "Why?" you ask, and rightly so. Dallas had to do this to be upwardly compatible with the Philips C552. Why did Philips do this in the first place? We don't know for certain but we suspect temporary insanity. The bottom line is that *typical 8051 programs will not run correctly on the DS87C550* so don't try loading generic 8051 programs, even simple "hello world" programs (the interrupt vectors are

different!) until you've modified them to work on the 87C550. Also, because of the added High Speed Micro features, Philips C552 programs may not execute as expected on the DS87C550.

8) To run your loaded file, move the HSM serial cable to COM0 (the default HSM/550 "user" serial port), and press the HSM/550 RESET button. The green RUN LED should light, and your sample program will emit a repeating message to your terminal software screen. Since the serial loader always uses HSM/550 COM1 and generic 8051 code will always use HSM/550 COM0 for serial I/O, you can leave both serial ports connected - COM1 to your PC, to load the program to be tested, and COM0 to another port on your PC, a different PC, serial modem, or other serial device which you are controlling.

9) That's it! You should be up and running. If you had problems, check your PC serial port connection and refer to the troubleshooting portion of this technical reference.

HSM/550 DETAILED DESCRIPTION

Philosophy and Purpose of HSM/550

The Systronix HSM/550 is the first development system for the Dallas DS87C550 High-Speed Micro with ADC and PWM. Like our HSM/KISS board, HSM/550 is designed from the ground specifically for the Dallas High Speed Microcontroller (HSM) family. There are some differences between generic 8051s and the HSMs. Just plopping an HSM into an 8051 socket will probably not give you the best performance, and will probably not even work at 22 Mhz or above. These differences are detailed in a separate section of this manual.

The analog I/O of the DS87C550 requires careful low-noise analog design. In addition to the built-in I/O of the DS87C550, we added a precision 12-bit DAC, a precision voltage reference, and low-noise CMOS opamps.

HSM/550's power supply is a low-noise, high-efficiency switching regulator. The frequency of the switcher is well above the input frequency range of the board's analog I/O. The switcher circuit is a proprietary Systronix design which has very low output impedance and very low ripple. It is actually quieter than many analog regulators! We isolated the analog and digital power systems with a low-impedance PI filter and split internal power and ground planes.

a. Easy to use eval/development board for the Dallas DS87C550.
b. *Full featured* - complete with 60 Kbytes each of code and data memory (we sell no boards with empty sockets as some vendors do to advertise a low price).
c. Analog rich: ADC, DAC, and precision reference. All analog I/O buffered. Multiple channels of PWM. Great for development of analog control loops.
d. Clock and calendar for time stamping or time-based control.
e. Designed for prototyping and development - generous through-hole and surface mount prototype areas, clearly labeled access to all controller I/O port pins.
f. SBX connector for additional I/O or prototyping expansion.
g. Easy to use - all you need to load a program is a PC with a serial port.
h. Built in auto-bauding serial loader for easy program loading and testing.
i. High Speed - designed for use at up to 33 MHz with no data "stretch" cycles. In order to run at full speed, the Dallas HSMs have special needs compared to generic 16 MHz 8051s. HSM/550 has been designed specifically for the HSM family. "Accept no substitutes".

HSM/550 Versions

HSM/550 is available only in a 33 MHz version. You can change to a slower crystal if you wish. Be sure to use a parallel resonant crystal designed for 18-20 pF load capacitors. Systronix stocks such crystals in 33.0000, 22.1184, 14.7456, and 11.0592 MHz versions.

Check our web site for special offers or special bundled versions of HSM/550. We also offer

aggressive quantity and educational discounts.

HSM/550 Features

a. PLCC68 socket for DS87C550 High Speed Microcontroller or In Circuit Emulator.
b. 10-bit ADC, 8 channels, with opamp buffers and selectable input range 0-2.048, +/- 2.048, or 0-4.096 volts.
c. 12-bit DAC, 1 channel, range 0-4.095 volts (1 mV per count).
d. Two available non-inverting opamps with gain continuously adjustable from 1-11.
e. Precision 2.048 volt reference used by ADC and DAC.
f. Low noise and low-temperature coefficient LinCMOS opamps on all ADC inputs and the DAC output. Opamps are standard DIP14 quad packages in sockets. Feel free to try opamps of other manufacturers, or opamps which meet your specific design requirements.
g. 128Kx8 NVRAM with 3V CR2032 Lithium battery backup, divided into a 60 Kbyte code page and a 60 Kbyte data page. Due to the memory-mapped I/O space and the way in which the loader accesses code memory, the upper 4 Kbytes of the NVRAM are not available as either code or data. Backup retains code and data for at least five years (typical at room temperature) without system power. On board circuitry protects the NVRAM from invalid write cycles during power up and power failure conditions.
h. 4 Kbytes of memory-mapped I/O space (that's why each page of the NVRAM has 60 Kbytes usable out of of it's 64 Kbytes total).
i. On-board auto-bauding serial loader accepts a HEX file from UART1. Leaves UART0 untouched. Use any terminal program to transfer the file. Does not use DTR to trigger load mode.
j. Serial I/O: RS232 on UART0 and UART1.
k. Early power fail interrupt provides time for orderly system shutdown.
l. On-board high efficiency, low noise switching regulator, 6-12 VDC input, 5.5x2.5 mm power jack. The regulator can provide up to 500 mA at 5 volts for your use, provided that you use a suitable input power source for the board.
m. Push-button high drives one or more of INT2, INT3, INT4 OR INT5 (jumpers select).
n. Push-button low drives one or more of INT0 or INT1 (jumpers select).
o. LEDs to indicate load (red) or run (green) status.
p. Two amber LED test points - LED lights when point is driven low. Requires less than 1 mA of low sink capability on the driving source, so can be driven by any port pin.
q. Microcontroller supervisor and reset circuitry, with early Power Fail Warning interrupt to INT0 (can be isolated if you don't want PFW driving INTO).
r. Generous prototyping area for DIPs and also an *SMT proto area!* Proto area has a power plane on the front side of the board and a ground plane on the back. Multiple solder pads for power on the front and ground on the back make connecting your own chips quick and easy.
s. All processor signals brought out to headers, labeled on *both sides of the board* for easy wiring and probing.
t. Serial loader accepts Intel Hex files from any terminal communication program. The serial loader is active only in LOAD mode. In RUN mode it is mapped out of processor memory and does not use any processor resources. Our serial loader is completely non-intrusive.
u. Dallas MicroLan connector with on-board DS1820 temperature sensor.
v. On-board DS1284 clock and calendar with battery backup. The DS1284 can generate periodic- or calendar- based interrupts.
w. Standard Euroboard 160x100 mm size, conforms to ANSI/IEEE Std 1101/1987, IEC 297-3-1984, and other international standards. HSM/550 fits Systronix enclosures and many others.

Loader and PLD Options and Updates

We are planning to add additional features in future releases, and we welcome your input. Please contact us if you would like a special version of either the loader or any of the PLDs. We can design and produce hardware built to your specifications, and have done so for a variety of small and large companies.

If you wish to use the loader or PLD in your own products, we can provide programmed parts or a license to create your own, at a very reasonable price. We have, in fact, done so for several customers. The serial loader and memory control PLD, and their associated circuitry are copyrighted by Systronix, Inc, with all rights reserved. Purchasing HSM/550 does not give you the right to duplicate either device.

Systronix Web Site & Forum

Our web site (www.systronix.com) is the main repository for new HSM/550 example code and documentation. There are links from the web site to an ftp file area. You can also join a forum of Systronix users. Our web site has information about the forum..

Getting Technical Support

Our technical support is included with your purchase of HSM/550. We believe good support begins with good written documentation (starting with what you're reading right now). If you can't find the answer in our documentation, then try the FAQ on the web at www.systronix.com, send e-mail to support@systronix.com, call us at 801-534-1017, or Fax us at 801-534-1019. When you contact us, please tell us about any errors or weaknesses in the documentation so that we can improve it in the next revision.

If you can, please contact us by e-mail first. You can attach a file of source code and captured output (use MIME encoding if possible) to your message. If you send us an example of a problem please make the example as simple as possible, and include any necessary I/O driver "include" files if you have modified them. *Please send us ASCII text or PDF files rather than word processor files*. Due to the risk of viruses, we cannot accept word processor files containing macros. We try to answer all e-mail within one or two business days.

Support does not include designing or writing code for you or debugging your application. We'll be happy to give you some general suggestions. Beyond that, if you need an application developed, we offer that service as well.

Please feel free to contact us with any unusual questions about programming HSM/550. We can probably help you approach your needs in the most efficient way. That's why we're here! Customers consistently give us high marks for courteous, competent technical support. But, we're only human and sometimes we make mistakes. We listen to our customers (we've gotten some of our best product ideas from them!) so tell us how we're doing for you.

Installing and Using the DOS A51 Assembler

We recommend you use the new RAD51 Windows IDE and Assembler. However, if you will be using the A51 DOS assembler, install it now. Just copy the file A51.EXE from the diskette into the desired directory on your hard disk.

Invoke A51 With a Batch File

Instead of typing a long command line to invoke A51, use our batch file ASM.BAT. Just invoke the batch file followed by the .ASM file which you wish to assemble:

```
C:\HSM\A51 HELLO
```

Here's the contents of ASM.BAT. "%1" is the filename you pass to the batch file on the command line.

```
Echo off
Echo assembling %1.asm with A51
a51 %1.asm -o %1.hex -l %1.prn
```

Detailed documentation for the DOS assembler A51 is available in a PDF file on your HSM/550 disk.

Installing and Using the Windows RAD51 IDE & Assembler

RAD51 requires 32-bit Windows such as Windows 95, 98 or NT 4.0 or later. It is not compatible with Windows 3.X and also may not work correctly with NT 3.X. RAD51 installs with it's own "setup.exe" install program. Online documentation for the assembler is included.

HSM/550 LOADER/DEMONSTRATOR

It's tempting to jump into a complex application right off the bat, but *please run one of the supplied sample programs first*. This will verify that your PC, cable, hardware and installed software are all working together, and that you understand the complete download process.

Most of the problems people have getting started are serial cable or PC serial communications related. So be sure your PC serial port is properly connected and configured. We test every HSM/550, and it's serial port is quite rugged, so it is unlikely that the HSM/550 serial port is the problem. Refer to the troubleshooting section for more tips.

Remember: use a straight through serial cable (not a null modem cable), and connect to HSM/550 COM1, not COM0. Then send the loader a carriage return (^M, 13 decimal, 0D Hex, or 0x0D, usually the 'enter' key). More than half of the support calls we receive are related to these simple issues, usually because a new user didn't even read the "quick start" section of this manual. Obviously, you're not making that mistake - give yourself two points.

What is the "Smart Autobauding Loader/Demonstrator EPROM?

That's a rather ungainly name, but it was the best we could think of. We're not an ad agency, just a bunch of engineers. The loader provides Intel HEX file transfer, memory fill, HSM Port read and write, SFR access, I/O space peek and poke, and much more. I/O space "peek and poke" is really just I/O space "read and write" but "peek and poke" somehow sounds like you're doing more. *You can use any terminal emulator or communications program which supports standard RS232 serial I/O.* You don't need a "Wintel" PC.

The HSM/550 serial loader/demonstrator uses the HSM second serial port (COM1 on HSM/550). The serial loader resides in a 27C256 EPROM or equivalent in socket U2. Jumper JP2 must be in the "EXT" position to drive the HSM's EA (External Access) pin low. We ship HSM/550 with this jumper in the EXT position (so we can test the loader).

Autobauding Loader: Supported Baud Rates

The Loader/Demonstrator syncs to any standard baud rate from 600 to 57,600 baud with a 33 MHz crystal. Note that with a 33 MHz crystal, 38,400 baud error is greater than 3% and may not be reliable. Here's a table of supported baud rates. Crystals on the HSM/550 and in your PC are not perfect, especially at temperature extremes, so you may find that your PC does better or worse than the table indicates. For example, one new NT workstation here at Systronix World HeadQuarters (WHQ) does fine with a 33 MHz HSM/550 at 38400 baud, while another identical workstation occasionally receives garbled characters.

HSM/550 Loader/Demonstrator Baud Rates								
Crystal	600	1200	2400	4800	9600	19200	38400	57600
33.000 MHz	Y	Y	Y	Y	Y	Y	error >3%	Y
22.1184 MHz	300	Y	Y	Y	Y	Y	Y	N
14.7456 MHz	150	Y	Y	Y	Y	Y	?	N
11.0592 MHz	150	Y	Y	Y	Y	Y	N	N

Serial Loader 100% Non-intrusive in RUN Mode

The serial loader is active only in LOAD mode. Our proprietary (meaning we racked our brains for a while to figure out how to make it all work) memory control PLD provides this function. In RUN mode, the serial loader is mapped out of processor memory and does not use *any* processor resources. In RUN mode the serial loader EPROM is placed in a low-power, inactive state. **In RUN mode our serial loader is completely non-intrusive and will not interfere with your application program in any way**. This is a very important capability, and one of the nicest features of HSM/550.

The serial loader includes other functions such as memory test, reading and writing to controller ports, and so forth. It is not a monitor, so it cannot be accessed from your application program, and cannot set breakpoints, single-step your program, and other typical monitor functions. Monitors which do this should be part of your executable program. Such monitors are available from a number of sources, some free. If we ever get a lot of time on our hands we may develop one ourselves.

HSM/550 Memory Map in Load and Run Mode

HSM/550 uses a single 128Kx8 SRAM with nonvolatile control logic and a battery backup to make this a 128Kx8 NVRAM (NonVolatile RAM) device. The PLD in location U5 splits the NVRAM into two pages. Page0 is the lower 64 Kbytes and Page1 is the upper 64 Kbytes. The 4 Kbyte memory-mapped I/O space disables the upper 4 Kbytes of each NVRAM page.

While executing out of the serial loader, the code page of the NVRAM is mapped into the data space of the controller. This enables the loader to write to what will be the code page (in RUN mode) of the NVRAM. It also prevents the controller from accessing any external data memory while in LOAD mode. Therefore the loader firmware is written to use only internal data memory. The table *HSM/550 Serial Loader Modes and NVRAM Memory Access* describes the relationship between load and run modes and NVRAM page access.

HSM/550 Serial Loader Modes and NVRAM Memory Access					
Reset Type:	Code Space	Data Space	LED	Controller	Hardware
Power-On Reset	NVRAM page0 (user program)	NVRAM page1 (user data)	green	hard reset	hard reset
Reset/Run Push-button	NVRAM page0 (user program)	NVRAM (user data)	green	hard reset	hard reset
Load Push-button	Serial loader in ROM	NVRAM page0 (temporarily mapped into data space for program load & verify)	red	hard reset	hard reset

Reset/Run Mode: Power on reset resets the controller and causes execution to begin in the code page of the NVRAM. The RESET push-button also resets the controller and causes execution to begin in the code page of the NVRAM.

Load/Program Mode: Pressing and holding the LOAD push-button resets the controller as the reset button does. But if you continue to hold the load button down for a half second or so, execution starts in the serial loader.

Using the Serial Loader & Demonstrator

Serial Loader Syntax, Command Entry, and Special Characters

All loader values must be provided in hexadecimal format, without any special characters. For example, 0ff , ff, FF, and 0FF are valid values. 0xff and 0FFH are not and will be considered errors by the loader. All loader input is case-insensitive.

Data ranges are not tested, so you can enter ‘99999' for an address or even a byte value. The least significant digits (either two or four, depending on whether an 8- or 16- bit value) are used, others are ignored. This keeps the loader code fast and clean but doesn’t warn you if you enter nonsensical or over-large values.

All commands are one or two characters with optional arguments. Arguments must be separated by one or more tab or space characters. The entire command line must be completed with a carriage return

Control-C (^C, 03 hex) will restart the loader. A backspace key (08 hex) causes a destructive backspace. The loader always uses xon (^Q - 11 hex) and xoff (^S - 13 hex) flow control. The flow control is accepted as input from the host PC - in other words the PC can pause the loader’s output. The loader does not emit flow control characters to the PC, so the PC can send a stream of data at the selected baud rate and the loader is guaranteed to keep up. As you use a slower and slower crystal, the fastest baud rate supported also decreases (9600 baud max at 1.8432 MHz for example), so the maximum incoming data rate is self-limiting.

Serial Loader & Demonstrator - Tips & Tricks

Automate Your Testing with 'Script' Files

In load mode, you can of course send a series of keyboard commands to the loader. On HSM/550 this could include setting up the DS87C550's analog to digital converter and reading the output. "OK", you say, "maybe that's cool but where's the tricky part?"

Here it is: you can prepare an ASCII text file with these commands, one per line. Then send the file to HSM/550. In Hyperterminal it's Transfer->Send Text File. You will need to set the File->Properties->Settings->ASCII Setup->Line Delay to 500 or 1000 msec, to give the loader time to process the command. If your communications software supports macros or scripts, so much the better. We've included some sample files we use with Hyperterminal. More are on the web site.

This is a simple and powerful way to test register settings before you take the time to create a program. It's much easier to debug a DS87C550 configuration problem at the loader command line, than within a program. We use this approach a lot in our own test and debug work here. That's why we took the trouble to add these features to the loader - we thought you'd find them useful too.

Serial Loader Command Description

? - loader on-line help

This command causes the loader to emit several lines of command help.

L - Load Intel HEX file

This command tells the loader to await reception of an Intel HEX file, and upon it's reception, to load it into NVRAM page 0. In RUN mode, NVRAM page 0 then becomes the code page and will cause your HEX file to execute. Errors are reported as they occur. The incoming file is not echoed to the serial port unless the T command has been given prior to the L command. If the load is error-free, the loader will issue an "OK" response. The loader always uses x-on/x-off flow control when echoing the file (i.e. you can pace the output from the loader). Give the loader the L command, then send a HEX file as standard ASCII text - do not use protocols such as zmodem. In HyperTerminal this means using the "Transfer->Send Text File" menus.

D [start [end]] - dump intel HEX file

Send the contents of NVRAM page 0 (the RUN mode code page) to the serial port, in HEX file format, beginning at address {start} and proceeding through address {end}. If no addresses are provided, dumps all of page 0.

V - verify Intel HEX file

Tells the loader to await reception of an Intel HEX file to be compared to those addresses in NVRAM page 0. After you enter the V command, send a HEX file just as if you were using the load command.

T - toggle Intel HEX file echo

Causes the loader to echo all incoming HEX files back out the serial port so that you can see it as it's being received. The echo persists until you enter the T command again or reset the loader.

I - toggle interrupt test

This is not just a loader command, it is also interrupt vector code contained within the loader. If you trigger an external interrupt while the loader is active, and interrupt test is enabled, the loader will emit a message to HSM/550 COM1. This is useful for verifying that your interrupt hardware is working correctly, or for testing the pushbuttons on HSM/550. All interrupts are assigned the same priority level. This feature is only available in LOAD mode and does not in any way affect your program's interrupts in RUN mode.

An interesting test is to jumper-enable multiple interrupts. When you press the button, all jumpered interrupts will be asserted at the exact same time. This is a good test case for interrupt handling code. Serial loader interrupt code handles each interrupt in order of its priority. No interrupts are lost. Try it! Then change the interrupt priority with an appropriate WR command and try it again.

C [start [end]] - calculate CRC-16

Calculate a 16-bit Cyclic Redundancy Code over the range of NVRAM page 0 addresses. If no addresses are provided, calculates over all of the page.

E - erase NVRAM page 0

Erase all of NVRAM page 0 to the value FF, except for I/O space. NVRAM page 0 is xdata space in LOAD mode, but becomes code space in RUN mode.

F value [start [end]] - fill NVRAM page 0

Fill NVRAM page 0, with the value provided, except for I/O space. If no addresses are provided, fills all of the page. NVRAM page 0 is xdata space in LOAD mode, but becomes code space in RUN mode. You must provide a fill value or the loader will emit an error message without performing any filling.

G - get microcontroller port values and display them

Left to right, the loader will display Port 0 , Port 1, Port 2, and Port 3. If you use the HSM/550 pushbuttons to drive some of the Interrupt pins, you should see the change in values with the G command.

P val0 val1 val2 val3 - put valueX to microcontroller portX

Puts the values you provide to the ports, except for port pins which are used for TXD1, RXD1, RD, WR of Port1 & Port3.

WX address value - write the byte to the address in xdata space

NVRAM page 0 is xdata space in LOAD mode, but becomes code space in RUN mode. Xdata space also include I/O space at F000 and above. This command lets you set individual bytes of NVRAM page 0 memory, and write to any location in I/O space.

This command gives you the ability to manually write to any peripherals you've installed in the prototype area as memory-mapped I/O. This is very useful in debugging your hardware.

RX address - read the byte from the address in xdata space

NVRAM page 0 is xdata space in LOAD mode, but becomes code space in RUN mode. Xdata space also include I/O space at F000 and above. This command lets you read individual bytes of NVRAM page 0 memory, including any location in I/O space.

This command gives you the ability to manually read any peripherals you've installed in the prototype area as memory-mapped I/O. In combination with the W command, you can manually configure and test all your I/O devices.

WR address value - write the byte to the SFR address
This command lets you write any Special Function Register. Timed Access Registers are supported, and the loader performs that special access for you. There are no restrictions, so you can, if you wish, clobber the serial I/O used to access the loader. If this happens just reset the HSM/550.


DS87C550 SFR locations and bit mappings are different from 8051 SFRs and from SFRs of other High Speed Micros. Double check your SFR address from the DS87C550 data sheet!

RR address - read the byte at the SFR address
This command lets you read any Special Function Register. Timed Access Registers are supported, and the loader performs that special access for you.

WI address value - write the byte to the internal data RAM address
This command lets you write any internal data RAM location.

RI address - read the byte at the internal data RAM address
This command lets you read any internal data RAM location.

M - exhaustive test of NVRAM page 0
NVRAM page 0 is xdata space in LOAD mode, but becomes code space in RUN mode. This command does not access I/O space. This command writes a series of patterns to memory and reads them back. Errors are reported as they occur. This tests all bits if memory with ones and zeros, and will detect shorted address or data lines. It doesn't help you find which lines are shorted, however.

A - address line test
This command does not access I/O space. This command write a series of patterns to memory and reads them back at locations calculated by walking a 1 across the value of the memory address. Errors are reported as they occur. This test is good for detecting shorted address lines. For example, if you get errors at locations 1000 and 4000, then A12 and A14 are shorted together.

X {value} - change movx stretch cycle value
This command changes the value of the controller's movx stretch cycle register. The movx stretch cycle is similar to "wait states" in a microprocessor memory cycle. After a reset, the value defaults to 1. It can be anything from 0 (no stretch cycles) to 7 (for the slowest write cycle).

Any change to the value persists until the next board reset. It applies to all movx cycles, including memory test and I/O access. HSM/550 is designed to support a stretch cycle value of 0 in all write accesses to on-board data memory. If you add peripherals to the prototype area, you may need to use a slower write cycle when you access them.


Connect an oscilloscope to the WR strobe (P3.6, on the Port3 header just below the processor). Run the A (address line) test or M (memory) test. Adjust the 'scope for a convenient display of pulse width. Now enter the command "X 0" to change the stretch cycles to zero. Run the memory test again, watch the oscilloscope, and you can immediately observe the difference in write pulse width.

WD - write value to DAC (HSM/550 only)
This command is only available on systems which have a DAC, the HSM/550. Meaningful values are 0-FFF.

S - signature
If you don't read this manual, you won't learn about this command, since it's not in the on-line loader help. This command emits the signature of the authors of the serial loader. It is located at the

top of code memory in the serial loader, so it is a good test of the integrity of the loader address and data lines.

Other new features

See the readme file on disk, or the loader help screen for information on new features which your loader may contain.

HSM/550 HARDWARE

Controller or ICE Pod Installation

A PLCC68 part has one chamferred corner which must be aligned with the matching socket corner. Device leads are relatively fragile so be careful to line up the part in the socket precisely before pressing it in place. Refer to the schematics and the Dallas data sheets for processor pinout details and signal names.

If you are using an In Circuit Emulator (ICE), turn off the board and ICE power. Insert the ICE pod, then turn on the ICE and board power at the same time. We recommend using a power strip to do this. If you power one and not the other, you could cause CMOS components in the ICE or the board to latch up, possibly destructively. We've seen this erase PLDs and damage the ICE. That's when we started using a power strip to turn both the board and the ICE on and off simultaneously.

Power Supply

The power input jack is 5.5 x 2.5 mm, with center terminal positive and the sleeve negative. The voltage regulator will accept supplies up to an absolute maximum of 24 VDC.. Power supply input voltages above 6 VDC are acceptable, and up to a point (about 9V DC) will make the regulator more efficient. The regulator will simply shut down if it is short-circuited or overheats. The regulator will start to drop out at about 5.5 volts typical. Reset circuitry puts the board in a safe reset state if VCC drops to 4.5 volts.

If you provide your own power supply be sure it is a DC voltage of at least 6 volts. Some wall cube power supplies are actually AC transformers and may damage the board. HSM/550 ships with a 12 VDC 1000 mA power cube. With an input voltage of 7 VDC, HSM/550 draws 180 mA typical, and with 6 VDC input, 210 mA. Current use is lower at higher input voltages because of the power conversion of the switching regulator (Power = Voltage X Current).

If you will be driving the board from a 12 VDC vehicle electrical system, the input voltage will typically be 13.8 VDC. HSM/550 includes reverse-polarity and overvoltage protection. The overvoltage typically clamps at 24 VDC.

HSM/550 is designed to be operated from a 6 volt or greater battery. All the on-board circuitry is relatively efficient. The DS87C550 has numerous power conservation modes. We may offer a special low-power version with 'zero-power' PLDs, let us know if you're interested in this.

I/O Mapping

HSM/550 uses memory-mapped I/O to support on-board memory-mapped peripherals, the SBX connector, and your own prototype area devices. PLD U10 is the memory-mapped I/O decoder. HSM/550 reserves 4 Kbytes of external data space for memory-mapped I/O at address FXXXH - FFFFH. The PLD uses address lines A6-A11 plus a decode of FXXX (A12-A15) from PLD U5. The result of all this is that I/O space is decoded into some 256 byte blocks such as FEXX, and some 128 byte blocks such as FC8X-FCFX. Some peripherals such as the DS1284 clock and calendar need a 256-byte I/O space while others such as the SBX connector require less.

The serial loader RX and WX commands will read and write this I/O space. As far as the controller is concerned there is no difference between *external data memory* and *external data I/O*. Hence the term “memory-mapped I/O”. On the other hand, serial loader *program space* commands such as erase and fill commands, and memory and address tests will not write into this reserved I/O space.

The HSM/550 PLD U5 provides a decode of FXXXH asserted low. You can use this FXXXH signal as an input into an additional PLD which will decode the I/O space further. HSM/550 PLD U10 used this decode plus A6-A11, ALE and RD and WR to provide the I/O decodes shown in the table. **Header J3 provides access to all the memory mapped I/O signals which are available for your custom use.**

HSM/550 MEMORY MAPPED I/O TABLE					
Address Range (hexadecimal)	signal name at U10 and/or J3	Active level	Signal location	Bytes	Description
F000-FFFF	FXXX_L	LOW	J3-3	4096	address decode at FXXX portions used by HSM/550 (described in this table) also available for your use
FF00-FFFF	N/A	N/A	N/A	256	reserved by Systronix or decode it yourself
FE00-FEFF	CLK_CS_L	LOW	U10-15 U7-20	256	DS1284 chip select at FEXX used by HSM/550
FD00-FDFF and WR	FDXX_WR_L	LOW	J3-8	256	write strobe at FDXX available for your use
FD00-FDFF and RD	FDXX_RD_L	LOW	J3-7	256	read strobe at FDXX available for your use
FC80-FCFF	MCS1_L	LOW	J3-6 P14-20	128	SBX chip select 1 or available for your use
FC00-FC7F	MCS0_L	LOW	P14-22	128	SBX chip select 0 or available for your use
FB00-FBFF	FBXX_L	LOW	J3-5	256	address decode FBXX available for your use
FA00-FAFF	FAXX_L	LOW	J3-4	256	address decode FAXX available for your use
external data READ	RD_L	LOW	J3-1	64K	read strobe available for your use
external data WRITE	WR_L	LOW	J3-2	64K	write strobe available for your use

External Memory or Peripheral Devices

The demultiplexed low-order address lines A0-A7 are brought out to header P24. AD0-AD7 (MCU Port 0) are on header P20. The un-multiplexed high order address lines A8-A15(MCU Port 2) are on header P21. These signals are all used to access memory on HSM/550. You can easily add external memory or peripheral devices to the prototype area, address them with A0-A15, and strobe them with WR, RD and PSEN.

Recommended Peripheral Addressing

Many peripheral devices use 8 address/data lines plus chip select and read and write strobes. If all eight address lines are used inside the peripheral, it will have a 256-byte address space. Therefore,

it's convenient to only decode the upper byte of the HSM's address, into 256-byte I/O blocks at F0XX, F1XX, F2XX,... FFXX.

You will want to decode your device chip select into this HSM/550 address space at F000H - FFFFH to avoid conflicts with the HSM/550 NVRAM. We've provided the address decodes in the table above. If you wish, you can further decode all 16 address bits down to a single byte address. Be aware that adding more decoding slows down the timing of your chip select and will probably require stretch cycles when accessing that device. The decodes we've provided are done in high speed PLDs and do not require additional stretch cycles.

Create your own decoder PLD or ask about stock devices available from us.

Protecting Processor Pins from Static or Under-voltage

If you will be accessing machinery or equipment attached by cables to the address and data busses of the HSM controller, you will need to buffer them or at least protect them against static or under voltage conditions before they leave HSM/550. **Do not wire controller pins directly to a cable which drives a printer or other device.** Page 60 of the 1993 *Dallas Soft Microcontroller Data Book* shows a combination of schottky diodes and 1K ohm resistors to protect processor port pins from negative voltages. Other application notes on pin protection are available from Dallas Semiconductor and Systronix.

Alternatively, ICs are available with active clamping and current limiting. One such device is the Texas Instruments TL7726 hex clamp. As the name implies, it has six inputs which provide protection for six I/O lines. It is intended to be used with an external current limiting resistor on each input. Data sheets and application notes are available at the Texas Instruments web site, www.ti.com. Systronix stocks the TL7726 in both DIP8 and SOIC8 packages.

Testing HSM/550 After Adding Peripherals

The loader M and A commands provide memory and address line tests, respectively. It's a good idea to run these after you add peripherals to your HSM/550 to verify that you haven't shorted address or data lines to power, ground, or each other. Of course if the shorts are really bad, the test won't run at all. So check your work as you go and run the test as you add each component.

HSM/550 Voltage Monitor, Reset and NVRAM Control

On board circuitry protects the NVRAM from invalid write cycles during power up and power failure conditions. Board reset occurs at Vcc-10% or 4.50 volts nominal. An early Power Fail Warning interrupt occurs when the unregulated power input drops below 6 volts. The NVRAM is forced into low-power data retention mode when Vcc drops by 5% and the controller is reset. It is necessary that the system software not write to SRAM after the PFW interrupt. The NVRAM data is maintained until its backup supply drops below 2 volts.

Lithium Battery

The CR2032 lithium battery is sized to provide typically five years or more of current. The lithium battery has a temperature range of 0 to 70 C. Below 0 C the battery will freeze and NVRAM contents will be lost. When the battery warms up it will again function and you can load a new program. Although the battery will provide backup down to almost 2.0 volts, it is a good idea to

replace the battery if its voltage at room temperature drops below 2.7 volts. Any good grade of battery is acceptable. Lithium CR2032 batteries are available from Systronix, at most camera stores, department stores and even many drugstores.

Test points TP3 and TP4 are provided to measure the voltage across a 100 ohm resistor in series with the lithium battery. With no external power source, this voltage is typically less than 50 nV, or a current of 500 pA to maintain the NVRAM. With power on, a current of typically 300 nA flows into the lithium battery from the board's power supply. This is normal and does not harm the battery.

Note that the battery must be installed in order to access the DS1284 clock and calendar.

Voltage Monitors

The DS87C550 specifies 4.00/4.13/4.25 V minimum operating Vcc. The HSM (and DS5000 family) power fail warning occurs at 4.25/4.38/4.50 min/typ/max. Typical 5V components such as SRAMs are specified with 10% Vcc tolerance, or 4.50 to 5.50 volts operating voltage

In order for HSM/550 to function reliably, all of its components must be operating properly, not just the controller. Therefore, system operation must stop when Vcc is less than 4.5 volts. HSM/550 uses a "5%" supervisor chip whose reset threshold is 4.50 volts minimum. A 5% supervisor resets the entire HSM/550 system at 4.50/4.65/4.75 volts min/typ/max. A 10% Vcc reset supervisor would allow more time to operate as power is failing but marginally violates the specification of the SRAM and other chips. HSM/550 write-protects the NVRAM and resets the controller when Vcc drops to the reset threshhold of the supervisor chip. To prevent a reset during a NVRAM write cycle, the controller must stop writing to NVRAM before Vcc decreases to the reset threshhold. Note that read cycles while power is failing are actually OK and will not corrupt NVRAM contents.

PFW Interrupt and System Shutdown Time

To provide time for system shutdown, we would like an early power failure interrupt. Ideally we should monitor the voltage prior to the on-board regulator. The power failure warning interrupt built into the 87C550 may never be triggered since its threshold (4.25/4.38/4.50 min/typ/max) is below a 5% Vcc reset threshold and identical to a 10% threshold. Therefore HSM/550's reset circuitry will be triggered before the 87C550's internal reset circuitry. The 87C550's built-in PFW interrupt has another limitation - it can only monitor Vcc since it is internally connected to the controller's power pin. The supervisor used in HSM/550 monitors the unregulated DC input and emits a PFW interrupt when the unregulated power input falls below about 6 volts (depending on load). This provides plenty of time for shutdown, since at this early state, regulated Vcc is still at its nominal 5 volt value.

PFW Interrupt Routine Tips

The exact amount of time available between the PFW interrupt and a controller reset depends on several factors: component tolerances, temperature, power supply capacitance (both pre- and post-regulator), and the total current consumption of your system. To be absolutely safe, your PFW interrupt code should not perform any critical writes to the NVRAM. Finally, verify that your routine completes well before the worst-case Vcc reset threshold. You can use an oscilloscope to monitor the time at which your PFW routine drives an I/O device and the time at which the controller reset signal occurs.

Interrupts and Timer/Counter Inputs

The Dallas 87C550 has 16 interrupt sources with three priority levels. There are six external interrupts and ten internal ones. Of the external interrupts, interrupt 0 is the highest priority. When the external interrupt pins are not being used in their special modes, they can be used as general purpose I/O pins. For detailed interrupt information, and all internal interrupt details, please go to the DS87C550 data sheet.

External Interrupt Pins

Interrupts used by HSM/550 peripherals

INT0 is used by the HSM/550 power failure warning (PFW) signal, if JP25 is installed. INT1 is used by the DS1284 clock and calendar INTA or INTB outputs if JP1 is installed. INT2 and INT3 are routed through the normally connected jumpers JP15 and JP14, respectively. If there is no SBX card present or it does not use interrupts, INT2 and INT3 won't be driven by any part of the SBX hardware. INT4 and INT5 are not used by HSM/550 peripherals.

INT0/P3.2

(Active low, edge or level sensitive). External Power Failure Warning when the HSM/550 unregulated power input drops below approximately 6 volts. (Do not confuse this with the 87C550 internal PFW interrupt.) This should be the highest priority interrupt in your program. Jumper JP25 is provided to control the PFW-to-INT0 connection. You must install a jumper to enable the external PFW interrupt.

INT0 is also tied to jumper block P3. By default, it is not connected to the Low Level pushbutton switch S4. By placing jumpers on P3 you can connect INTO and/or INT1 to the pushbutton. The ability to drive more than one interrupt input simultaneously helps debug interrupt conflicts.

INT1/P3.3

(Active low, edge or level sensitive). Interrupt output from DS1284 clock and calendar. Jumper JP1 is provided to control which if any interrupt from the DS1284 is connected to the DS87C550. The silkscreen on the board identifies one end as INTA and the other as INTB. The center terminal of the 3-contact jumper goes to the DS87C550 INT1 input. If no jumper is present, no interrupt from the DS1284 is connected.

INT1 is also tied to jumper block P3. By default, it is not connected to the Low Level pushbutton switch S4. By placing jumpers on P3 you can connect INTO and/or INT1 to the pushbutton. The ability to drive more than one interrupt input simultaneously helps debug interrupt conflicts.

INT2/3/4/5

(Edge sensitive: rising, falling or both) **INT2 and INT3** are tied through JP15 (SBX Interrupt 0, DS87C550 INT2) and JP14 (SBX Interrupt 1, DS87C550 INT3). These jumpers are "normally closed" by a trace on the bottom side of the board. To disable this, cut the trace. You can then install a jumper block and a jumper if you later decide to re-enable them.

All **INT2/3/4/5** are tied to jumper block P2. By default, none of these inputs are connected to the High Level pushbutton switch S3. By placing jumpers on P2 you can connect one or more of the interrupt inputs to the pushbutton. The ability to drive more than one interrupt input simultaneously helps debug interrupt conflicts. If you aren't using the SBX interrupts (many SBX cards don't), there's no conflict with this pushbutton.

Analog Inputs

The DS87C550 has eight analog inputs, a single 10-bit analog-to-digital converter, and internal or external reference options. The HSM/550 board supplements this with eight op-amp analog buffers and a precision 2.048 volt reference.

Analog Input Buffers & Input Range Jumpers

The analog buffers are implemented in three quad Texas Instruments TLC27M4 opamps. The opamps are standard pinout DIP14 parts, so you can easily change to another opamp.

VREF for the ADC is either 2.5V (internal reference) or 2.048 volts (using HSM/550 external reference).

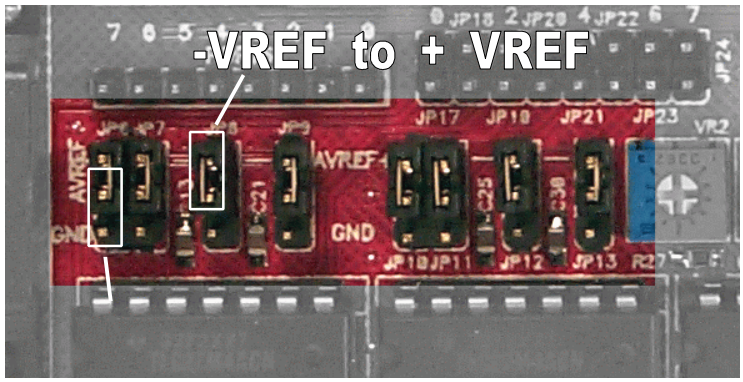


Figure 7. Analog Input Range Jumpers. The upper outline selects -Vref to +Vref. The lower outline selects 0 to 2xVref. No jumper present selects a range of 0 to +VREF volts.

The analog buffers have jumper-selected input ranges of -2.048 to +2.048, 0 to 2.048, or 0 to +4.096 volts (external ADC reference). The output of the analog buffers is designed to be always positive, since the input range of the ADC is limited to positive inputs with respect to ground.

When using the DS87C550 internal reference, the input range of the ADC changes but the range of the input buffers does not. This is because the internal reference from the DS87C550 is not brought outside the controller. However, you could supply an AVREF+ of 2.5 volts by dividing analog VCC by two. This would give buffer input ranges of -2.5 to +2.5, 0 to 2.5, or 0 to 5. We don't recommend doing this because using a reference of VCC/2 is much less accurate and stable than a precision reference such as the 2.048 reference supplied on HSM/550. How much? VCC/2 creates about a 5-bit reference which degrades the ADC to that same level.

The analog input buffer input ranges can be individually set for each input channel with jumpers JP6-JP13.

ADC Reference Selection & Reference Jumpers

The 10-bit ADC built into the DS87C550 can use either a reference internal to the DS87C550 or an external reference. This is controlled by the ADRS bit in SFR PWMADR at address D6H. The internal reference is 2.5 volts and is the default after a reset.

Setting ADRS to a '1' selects the external reference. In addition, the external reference jumpers must be set on header P23. Or take the jumpers off P23 and create your own reference in the prototype area, and then wire it to P23-2 and P23-3.

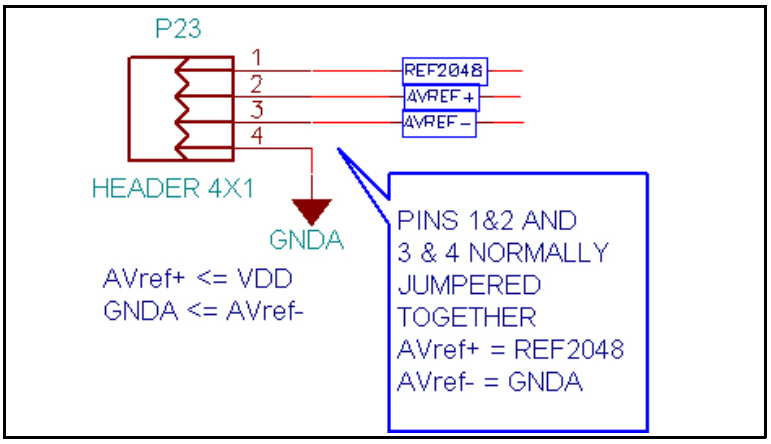


Figure 8. Header P23, ADC reference selection.

Analog I/O ADC and DAC Counts

External ADC reference

You may notice that, when using the HSM/550 external reference, the analog voltage ranges are multiples of 2048 millivolts which is 2^{11}. This is because the DAC reference is 2048 millivolts. When operating on a 5V supply, it is best to keep the maximum analog input a couple of diode drops below VCC, hence the maximum analog I/O value of 4.096 volts. This all works out rather conveniently: the DAC output is 1 mV per count (4096 millivolts is 2^{12}), the ADC input is also an even number of millivolts per count. This makes conversion of the ADC and DAC values to voltage very simple. For example, a DAC count of 1024 is 1.024 volts. With an analog input range of 0-4.096, and using the ADC in 10-bit mode, each ADC count is 4 millivolts. Therefore an ADC value of 512 is (512 x 4mV) = 2.048 volts.

Internal ADC reference

When using the internal ADC reference, the ADC range becomes 0-2.5 volts. The analog input buffer ranges are not affected by the internal ADC range. See the notes above on the analog input ranges. The range of the DAC is unaffected by the ADC reference, and remains 0 to 4.095 volts (2^{12}-1 mV).

■Analog Outputs

87C550 PWM Outputs

The DS87C550 contains four independent 8-bit pulse-width modulators. For complete details, see the DS87C550 data sheet PDF file and click on the Digital I/O bookmark for Port 6.

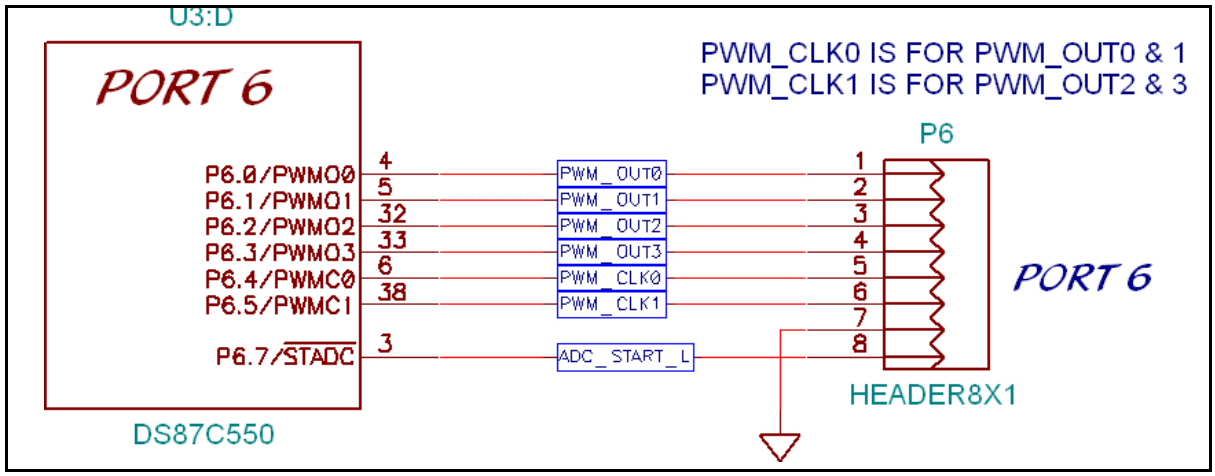


Figure 9. Header P6, PWM signals from MCU Port 6.

The PWM outputs and the PWM clock signals are presented on header P6. If you don't need PWM functions, you can use these Port 6 pins as generic 8051 I/O pins. Note that the ADC_START input is also a Port 6 I/O, although it has nothing to do with the PWM signals.

HSM/550 12-bit DAC

HSM/550 includes a high-quality Linear Technology LTC1451 12-bit DAC. The output of the DAC is buffered and presented on P8. Header P8 also provides access to the 'spare' opamp inputs and outputs.

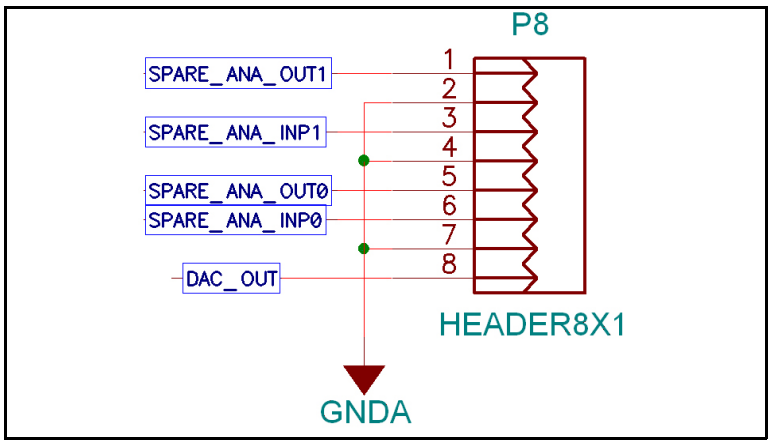


Figure 10. Header P8, DAC output and available analog opamp input & output.

Designer opamp preferences are about as personal as political choices. There's no real 'best'. We chose the TLC27M4 because it has good single-supply operation, is low power, has low input current and low temperature drift. It drives very well to within a few millivolts of ground and up to 4 volts or so. Heavily loaded it will drive to 3.5 volts, so be aware of this limitation. All the "rail-to-rail" opamps we've investigated have other shortcomings which we felt made them a less desirable choice in this application. But if you disagree, feel free to install your favorite opamp in the sockets we provided.

Available OpAmp Gain Stages

HSM/550 includes two available opamps wired as non-inverting amplifers. VR1 and VR2 set the gain from 1 to 11. VR1 sets the gain for SPARE_ANA_OUT0 and VR2 for SPARE_ANA_OUT1. Clockwise rotation of the pot increases the gain. Header P8 also provides access to these opamp inputs and outputs.

Dallas MicroLan/iButton

HSM/550 includes a simple interface to the Dallas One Wire network (DOW net), also called MicroLan. This is similar to the iButton protocol, with the difference that DOW devices are addressable and designed to allow multiple devices to share the same network wire. HSM/550 includes a Dallas DS1820 temperature sensor. An indexed PDF file data sheet is available from Systronix.

DS1820 Temperature Sensor

The DOW net connector is an RJ11 type. You can use standard telephone-type RJ connectors and cable, available from a variety of distributors.

Caution: The DOW_DQ pin is tied directly to the DS87C550 P3.4. Exposure to static discharges could permanently destroy that pin on the processor. You may want to add your own static protection to the DOW net.

We have included a simple DS1820 test program in HEX form (no source code). It supports reading the DS1820. It does not support reading multiple devices on a network.

A full-featured DOW network driver with source code and technical support will be available “soon” from Systronix as a commercial product.

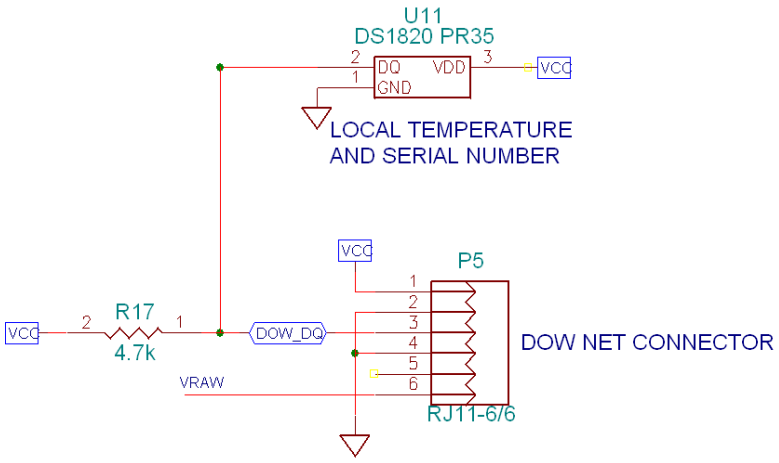


Figure 11. Header P5 and U11, DOW net

DS1284 Clock and Calendar

The DS1284 is a clock and calendar chip which operates on its own crystal, independently of the DS87C550 controller. The DS1284 is backed up by the same lithium battery which preserves HSM/550 code and data memory. The DS1284 is memory mapped at address FEXX.

The DS1284 can generate periodic or calendar-based interrupts which can be jumpered to the DS87C550 interrupt input INT1. The DS1284 also can generate an accurate 1 KHz square wave, available on J2. The DS1284 includes some NVRAM which is independent of all other HSM/550

memory. This is a handy place to store constants or parameters which you do not want to be changed by frequent new program loads. The HSM/550 NVRAM is never cleared or filled by the loader.

We've included a sample program, 550_ECC.ASM, which reads and displays the time since reset.

The complete DS1284 data sheet is included as a PDF file, so we won't attempt to go into detail on using it here. We've added a detailed index so you can quickly locate specific areas of interest.

SBX Mezzanine Connector

The SBX connector is a well-established, simple mezzanine bus connector. It's address space is limited, but the interface is simple and low cost. A variety of SBX boards are available from numerous vendors: if you are looking for some, start with a Web search.

Systronix SBX1 LCD and Keypad Board

This SBX card is an easy way to add the most common digital I/O to your system. SBX1 includes a 4x4 or 4x5 matric keypad decoder/debouncer, a parallel interface LCD connector, LCD backlight control, a buzzer, 24 bits of bidirectional I/O, and a standard Opto-22 style 25x2 I/O header.

Systronix SBX-P Prototyping Board

Coming soon - this prototyping board is still in the concept stage. E-mail your suggestions to sbxp@systronix.com.

Using Other SBX Boards

You "should" be able to simply plug them onto HSM/550, write the necessary firmware and go. The I/O space read and write commands (RX and WX) in the HSM/550 loader are very helpful in debugging your SBX interface. Test out your idea manually or by sending a text file to the loader, then write the same algorithm into a program.

TP1 and TP2 are included to tie +/- 12V to the SBX connector if your card requires it. You will need to provide your own +/- 12V supply, there is none on HSM/550.

Using and Modifying HSM/550 I/O and the Prototype Area

HSM/550 is notable chiefly for its analog I/O. We've also provided some basic I/O such as dual RS232 ports, but you can easily modify these or add more to suit your needs.

LED Test Points

T2 and T3, if driven low, will cause LEDs D5 and D6, respectively, to light. Only 1 mA of sink capability is needed, so these test points can be driven by a controller port pin.

Adding other Serial I/O

There wasn't room on HSM/550 to provide the means to change the serial I/O easily. If you need additional serial I/O or another type such as RS485, we recommend adding a UART such as the Philips 2691 (available from Systronix) to the prototype area. Or plug on one of the available SBX cards with additional serial I/O. Or use an available converter (from companies such as Black Box at www.blackbox.com) on the existing HSM/550 serial ouput.

S3 and S4 Pushbuttons

All interrupts are disconnected from the pushbuttons unless the P2 or P3 jumpers have shorting blocks installed. If you aren't using the switches S3 and S4 for interrupts (or even if you are) you can connect them to devices in the prototype area. Refer to the interrupt section for details. Some interrupts may be driven by HSM/550 devices - if you do wire them also into the switches and other prototype circuitry, the HSM/550 devices will then also be driving your circuitry.

VCC and GND in Prototype Area

We've provided marked headers adjacent to the prototype area for easy VCC (5 volts) and ground access. In addition, the top board layer in the prototype area is VCC. All the exposed shiny diamonds on the top of the board in the prototype area are solder-plated VCC attachment points.

The bottom board layer in the prototype area is ground. All the exposed shiny diamonds on the bottom of the board in the prototype area are solder-plated ground attachment points.

DALLAS HIGH SPEED MICROCONTROLLERS

The Dallas High Speed Microcontrollers (HSMs) are supersets of the venerable 8051 microcontroller. Well over 100 million 8051s have been shipped, and they are in use in everything from DNA replicators to blood glucose meters. The HSM family is code-compatible with the 8051 instruction set. The HSM core architecture has been redesigned to offer faster execution while consuming less power.

High Speed Microcontroller Data Sheets

A data book is available from Dallas Semiconductor. Call them at 972-371-4000, or better yet, get the data sheets today in PDF form at www.dalsemi.com. There is also a HSM data sheet link from our web site at www.systronix.com. Acrobat reader is free at www.adobe.com. Datasheets from Systronix require Acrobat 4.0 or later to display all the Acrobat features such as highlighting.

What's Different About the HSM Family

Port 0

No Port 0 pullups are required or advised with the High Speed Microcontroller Family. Port0 has special drivers which can tolerate up to 100 pF and meet the required high speed timing when Port 0 is used as the multiplexed address/data bus AD[7..0]. In the case of address to ALE setup, this means charging or discharging this much capacitance in 11 nsec at 22.1184 Mhz. Adding Port 0 pullups significantly degrades signal integrity on Port 0, causing more overshoot and undershoot, and more noise during ALE transitions.

Memory Timing

There is more to memory timing than this manual has space to discuss. Overall memory system timing is a combination of memory device (both SRAM and EPROM), processor speed, bus loading, address latch, and decode circuit timing. These calculations can be quite complex and can trade off component speed versus cost and availability.

Memory timing application notes are available at the Dallas Semiconductor web site at www.dalsemi.com, and there are also links to these from www.systronix.com. Suffice it to say that at frequencies above 16 Mhz, a faster Port 0 demultiplexing latch is required. HSM/550 uses an AC573. This device can produce large switching currents into capacitive loads. Low-value series termination resistors are advisable on its outputs to avoid excessive overshoot. For noise immunity, this latch should use CMOS input thresholds, never TTL.

To provide code access which complies with data sheet specifications, fast memory devices and fast control circuitry are also required. There is no provision to add wait states or stretch cycles to code access in the Dallas HSM family. HSM/550 uses a 10 nsec PLD to control memory access. All HSM/550 circuitry has been designed to provide reliable worst-case performance at full operating speeds.

Data access (both memory and I/O devices) can be slowed down with stretch cycles. All HSM/550 circuitry has been designed to provide reliable worst-case data access at full operating speeds with no stretch cycles added.

Strobes

Due to the fast transitions on strobes, termination may be required, depending on your design particulars. HSM/550 provides special termination of ALE for low emitted noise and reliable operation during Port 0 transitions.

Instruction Timing

The High Speed Microcontroller Family owes its performance improvements to a redesigned 8051 core. This core maintains code compatibility with the 8051 family while cutting the number of clock cycles per instruction cycle from 12 to 4. This gives a 3X increase in instruction cycles executed per second, assuming all instructions take the same number of instruction cycles. But they don't, so the performance difference is really between 2X and 3X for a typical instruction mix. Detailed instruction timing is available in the 80C320 family data book.

Power Supply and Reset Circuitry

Never, ever use a capacitor and resistor for reset of a HSM family processor. It's not even a good idea on a generic 8051. HSM/550 uses a precision reset control chip with proprietary Systronix circuitry to provide reliable reset control, NVRAM protection during power-up and power-down, and Power Failure Warning (PFW) interrupt generation.

Power supply accuracy and regulation with temperature and load variation is also critical. Low cost regulators with 2% or better tolerance are widely available, and are used in all Systronix products. We also use high-quality low-ESR electrolytic capacitors for bulk power supply bypassing. Bypass capacitors at the processor, memory, and high-speed support chips must be high-quality tantalum and multi-layer ceramic devices.

Additional Features

The High Speed Microcontroller Family has additional peripherals or memory, depending on the family member. Most members include a second UART, watchdog timer, and additional external interrupts. Some include 1 KByte of on-board movx memory.

TROUBLESHOOTING & DEVELOPMENT TIPS

No Serial Communication between PC and HSM/550

This is the most common problem. The serial port on HSM/550 is very simple and robust - it's very hard to damage it. If your PC isn't talking to HSM/550, its most likely that the problem is in your PC or the cable between your PC and HSM/550. The serial port on HSM/550 does not use hardware flow control, so presence or absence of handshaking signals from your PC has no effect on HSM/550.

1. Often, people connect HSM/550 to a previously unused serial port on their PC.
   a. If possible, use a PC COM port which you know has recently been operating correctly with another known good serial device such as a modem or printer.
   b. If you have a serial mouse or pointing device and you know it works, swap your mouse COM port to the unused port, and put HSM/550 on the former mouse port. You will have to change your PC's configuration in order to do this.
2. On a DOS or Windows 3.X PC, another I/O driver may be loading and interfering with the PC serial port you're trying to use.
   a. Check your PC's setup
   b. Try another PC
   c. Try using another, known good COM port on your PC.
3. You must use a straight-through cable (not a null modem cable).
   a. This means pin 2 is your PC's RXD (HSM/550 TXD) and pin 3 is the PC's TXD (HSM/550 RXD). Ground is pin 5. The pins are usually numbered, molded into the plastic inside the DB9 shell (it's very tiny print!). The schematics contain a detailed pinout of the serial connector.
   b. A null modem cable has TXD and RXD swapped within the cable to permit connecting 2 PCs together, as if there were a modem between them (hence the name "null modem"). You can't use a null modem cable with HSM/550.
4. If you have an oscilloscope or logic probe, connect it to RXD1, Port1.6 This is labeled on the header P4-6. This signal is the serial input to the controller, it is a CMOS level, and is not the RS232 voltage level. At 19.2 kbaud, bits are 52 usec wide, so set the oscilloscope time base to 50 to 100 usec per division. Now in load mode, when you press your PC's Enter key, you should see periodic characters on RXD1. This is your PC sending carriage returns to HSM/550. HSM should respond with a burst of characters on Port1.7, header P4-7, the controller TXD1 output. These characters are the loader printing the opening prompt to HSM/550 COM1. This verifies that the loader is receiving and sending serial information from/to the RS232 level translator.
   a. You must push the load pushbutton and hold it for more than 500 msec to trigger load mode. Each time you push the button in this manner, and send the loader a carriage return character (0DH), the loader should emit a brief burst of serial characters, the opening prompt. If you hold down the enter key the loader will continue to emit loader prompts. If TXD1 stays high, then for some reason the loader is not starting up.

   i. Check the board power at a VCC and GND test point. Is it getting 5 volts +/- 5% (4.75 to 5.25 volts DC)?
   ii. Is the ALE strobe oscillating? This indicates the processor is operating. If not, is the crystal seated in its socket? Absence of ALE indicates an inoperative controller chip, most likely due to a power or crystal problem.
5. Next check the RS232 level translator: carefully probe pin 14 of U16, the SP312A serial input from your PC. Pin 15 of U16 is the serial output from HSM/550. BE CAREFUL! If you short the charge pump pins to data pins with a scope probe, you could permanently damage the SP312A device and/or the HSM/550 controller. This is because RS232 voltage levels exceed safe CMOS or TTL levels.
   a. If serial input and output on the PC side of the SP312A are toggling between at least +5V and -5V with the cable connected, then RS232 serial data is getting out of the HSM/550 board and the problem is in your PC's serial cable or serial port, or the configuration of your PC terminal software.
   b. If RS232 output from the PC is toggling and the TTL signal to the controller is not, then the SP312A level translator is damaged. Likewise, if serial data is toggling from the controller but not appearing on the output of the level translator, it may be damaged or your PC or cable may be clashing with the RS232 signal. Remember that you will not see any output from the loader unless it receives a carriage return.

In any case, PLEASE CONTACT US before you return your board. We have almost a 100% success rate in solving problems like this over the phone. We do charge a minimum service fee on boards which are returned without authorization and check out to be operating fine. So please save yourself and us time and aggravation and call us first. We'll do our best to get you up and running right away. If you need to return your board, you need an RMA (return merchandise authorization). If you have a major credit card we can ship a replacement board at once without first receiving your old board back for evaluation.

Internet FAQ

There is a FAQ (Frequently Asked Questions) on our web site. It is updated on a regular basis.

Start Simple

Start with a simple program, get it working, and then add complexity in modules. Try to add new functions as subroutines which you can call or not call to easily isolate suspected problems.

Learning Assembly Code and Embedded Programming

If you find a simple way to learn assembly code or embedded programming, let us know! We get asked "how do I learn ..." quite frequently. Maybe in a few years there will really be such a science as Computer Science, but at the moment, starting a discussion about "good design" is a good way to start an argument. This manual is not intended to teach you assembly code or good programming principles. Here are a few good books. Some of these may be hard to find - good luck - we don't stock them for that reason. If your local big chain bookstore won't order them, try a college bookstore or a smaller, independently owned bookstore, or search for the title on the Internet.

The Art of Embedded System Programming by Jack Ganssle, published by Academic Press. This one should be easy to order at your local bookstore. This book focuses on philosophy and general good programming hygiene. It's not specific to any processor. Lots of true and humorous anecdotes, very readable. He also publishes a delightful free e-mail newsletter. To subscribe, send a message to

majordomo@ganssle.com, with the words "subscribe embedded your-email-address" in the body. Of course, substitute your actual email address for the text ‘your-email-address’.

The Microcontroller Idea Book by Jan Axelson. YES we do stock this one. It’s a good overview of microntroller interfacing ideas, intended for hobbyists or those who are new to embedded systems. Lots of schematics and sample code (available on disk) for BASIC-52 controllers.

C and the 8051 by Thomas Schultz. As the title suggests, Professor Schultz assumes you will be using C, so most of the source code is in C. But there is a lot of good information applicable to any language or assembly code. Available at major bookstores and www.amazon.com.

Exception Handling

All embedded applications should have run-time *exception handling*. Good exception handling is one hallmark of a good programmer. Exception handling is a difficult topic, and highly application dependent, so it is hard to make specific recommendations. The serial loader uses exception handling to deal with bad command lines, bad hex files, and similar situations.

If you are using BCI51 Pro, please, please take advantage of the BCI51 ONERR instruction to provide run-time error handling. Print the error code to the serial port so that you at least know what error occurred.

Quick Diagnosis Table

HSM/550 QUICK DIAGNOSIS TABLE

SYMPTOM	EXPLANATION	SOLUTION
RUN and LOAD LEDs are both off	no power to HSM/550	check power cube - center is positive VCC-GND short in prototype area could be causing regulator shutdown.
No prompt from loader on PC	no serial communications from HSM/550 to your PC	press and hold LOAD button, RED LOAD LED should be lit to confirm LOAD mode. serial cable must be straight-through is PC serial comm software properly configured? You must send a carriage return (0DH) to the loader to establish communications. Start with 19200 baud.
I added parts to prototype area, now operation of the loader is erratic	Overloaded or shorted address or data lines	If loader won’t even run, beep out address and data lines - look for shorts to each other or to power or ground in the circuitry you’ve added.
I put in a faster crystal, now the board doesn’t work reliably	33 MHz is the fastest crystal.	Don’t do this! HSM/550 will not run faster than 33 MHz.
I put in a slower crystal, now the board doesn’t work reliably	Crystal may be wrong cut or load capacitance for HSM/550.	The 8051 requires a parallel resonant, fundamental mode crystal with about 18 pF load capacitance. Not all crystals can generate all baud rates.

HSM/550 QUICK DIAGNOSIS TABLE		
SYMPTOM	**EXPLANATION**	**SOLUTION**
Loader starts up, displays help, but I get HEX file load errors	Loader EPROM address and data lines are OK, but not NVRAM address and data lines	Run loader M and A commands to test memory and address lines. If M command fails, run A command and beep out the address or data lines of circuitry you've added.
A simple "hello world" 8051 program I have doesn't run on HSM/550.	DS87C550 registers and interrupt vectors differ from those in a generic 8051 and even other High Speed Micros such as the DS80C320.	Modify the program to use the SFR and interrupt vectors of the DS87C550. Check the bookmarks in the PDF data sheet to quickly find areas of interest.
Serial I/O is garbled	Baud rate mismatch or PC serial comm program got "out of sync"	HyperTerminal in particular seems prone to lose sync with incoming data if it ever gets a partial character or data at the wrong speed. Close HyperTerminal and restart it. Get a free new version of Hyperterminal PE over the Internet.
I didn't use HSM/550 for a long time and my program is no longer in memory	Lithium battery ran down and you lost your program in NVRAM.	The Lithium battery should be good for five years at room temperature. High moisture or conductive atmosphere such as salt spray will increase battery leakage. Replace with a CR2032 3V battery when the voltage is less than 2.5 volts. Don't overload the Lithium battery output in the prototype area.
BCI51 program runs then appears to hang	Run time error may have occurred (divide by zero, math overflow, etc)	Use ONERR to print the error value to the serial port, or blink an LED with a pattern so that you know when a runtime error occurred.

Warranty

HSM/550 is warranteed against defects in manufacturing for a period of one year. This does not include damage due to static, driving controller I/O pins beyond their design limits (see the DS87C550 data sheet for pin specifications) or other damage caused by improper use. Systronix is not liable for latent bugs, if any, in micro-controllers. In the case of a defective controller or processor (such as the infamous Intel floating point bug) we will honor all manufacturer's warranties and make every reasonable effort to keep your system up and running.

PLEASE CONTACT US before you return your board. We do charge a minimum service fee on boards which are returned without authorization and check out to be operating fine. So please save us both some possible aggravation and contact us first. We'll do our best to get you up and running right away. If you need to return your board, you need an RMA (return merchandise authorization). If you

have a major credit card we can ship a replacement board at once without first receiving your old board back for evaluation.

HSM 550 DEVELOPMENT / EVAL BOARD – REVISIONS

REV OR ECO	DATE	BY	DESCRIPTION
0.1	99 JAN 05 99 JAN 20	BAB BAB	START OF SCH CAPTURE, BASED ON ALC.SCH AND HSM/KISS ADDING 1284 CLOCK AND CALENDAR. ONE RS232 CHIP. NO CS REGISTER. LITHIUM BATTERY.
0.2	99 JAN 24	BAB	DELETING DIGITAL I/O, OTHER CHANGES AFTER REVIEW ON 22 JAN. TRY ADDING 2ND SBX?
0.3	99 FEB 02	wsk	REFORMATED MADE READY FOR FIRST LAYOUT.
0.4	99 FEB 19	wsk	SHEET 3: Added C51,C54; SHEET 4: Added C52, C53; SHEET 5: Renamed nets LED0_L & LED1_L to LED3_L & LED4_L; SHEET 7: Changed orientation of VR1,VR2; SHEET 8: Added note at P23; SHEET 10: Changed note at JP15; connected SBX connector ground pins to GND net; SHEET 11: Added TP3, TP4, R25;
0.5	99 APR 06	wsk	SHEET 3: Swapped signals connected to pins 1 & 11 of U10. This returns the signals to their original assignments.
0.6	99 APR 09	wsk	SHEET 4: Added P24.
0.7	99 APR 15	wsk	SHEET 7: Added C55. SHEET 8: Added C56.
0.8	99 APR 19	wsk	SHEET 11: Connected LOADPB_L signal to the LOAD pushbutton. ALL SHEETS: Added legal notice.
0.9	99 APR 21	wsk	SHEET 3: Corrected C5 value; corrected R6 value; added TP5. SHEET 6: Changed analog power supply note. SHEET 7: Added analog power supply note. SHEET 9: Added RP13 A & RP13D (unused).
A	99 APR 21	wsk	Released to Manufacturing.
A.01 A.02	99 MAY 03 99 JUN 21	wsk bab	SHEET 10: Corrected source reference for RST_H. Sheet 11: added 2N4403 to load circuit to reduce battery charging from MAX703 MR

TABLE OF CONTENTS

TO DO:

Add silk screen to the back of PCB for ADC range jumpers.

SYSTRONIX
555 SOUTH 300 EAST
SALT LAKE CITY, UT, USA 84111
TEL: +1-801-534-1017 FAX: -1019
WWW.SYSTRONIX.COM INFO@SYSTRONIX.COM

Title	HSM550 DEVELOPMENT/EVAL BOARD REVISIONS	
Size: B	Number	Rev: A.01
Date: Mon Jun 21, 1999		Drawn by: B Boyes
Filename: hsm550a02.sch		Sheet 1 of 11

HEADERS

HEADER	FUNCTION	COMMENTS	PAGE
P1	MCU PORT 4	SEE MCU PORT TABLE	
P2, P3	SEE JUMPER TABLE		
P4	MCU PORT 1	SEE MCU PORT TABLE	
P5	DOW NET CONNECTOR	PIN 1: VCC PIN 2: GNDD PIN 3: DOW DQ PIN 4: GNDD PIN 5: NO CONNECT PIN 6: VRAW	9
P6, P7	MCU PORTS 6 & 5	SEE MCU PORT TABLE	
P8	DAC OUT, SPARE OP AMP I/O	PIN 1: SPARE ANALOG OUT 1 PIN 2: GNDA PIN 3: SPARE ANALOG IN 1 PIN 4: GNDA PIN 5: SPARE ANALOG OUT 0 PIN 6: SPARE ANALOG IN 0 PIN 7: GNDA PIN 8: DAC OUTPUT	7
P9	DIGITAL GROUND TEST POINTS	PINS 1-4: GNDD	11
P10	VCC TEST POINTS	PINS 1-4: VCC	11
P11	ANALOG GROUND TEST POINTS	PINS 1-4: GNDA	8
P12	VDD TEST POINTS	PINS 1-4: VCC	8
P13	COM1: PROGRAM LOAD CONNECTOR	PIN 1: DCD (ON) PIN 2: TxD1 PIN 3: RxD1 PIN 4: DTR (NO CONNECT) PIN 5: GNDD PIN 6: DSR (ON) PIN 7: RTS (NO CONNECT) PIN 8: CTS (ON) PIN 9: RI (NO CONNECT)	9
P14	SBX CONNECTOR	PIN 1: SBX +12 PIN 2: SBX -12 PINS 3,17,35: GNDD PIN 4,18,36: VCC PIN 5: RST_H PIN 7: A2 PIN 9: A1 PIN 11: A0 PIN 12: SBX INTERRUPT 1 PIN 13: WR_L PIN 14: SBX INTERRUPT 0 PIN 15: RD_L PIN 19: AD7 PIN 20: SBX CHIP SELECT 1 PIN 21: AD6 PIN 22: SBX CHIP SELECT 0 PIN 23: AD5 PIN 25: AD4 PIN 27: AD3 PIN 29: AD2 PIN 30: OPT0 PIN 31: AD1 PIN 33: AD0 PINS 6,8,10,16,24,26,28,32,34 NO CONNECT	10
P15	COM0: MAIN APPLICATION CONNECTOR	PIN 1: DCD (ON) PIN 2: TxD0 PIN 3: RxD0 PIN 4: DTR (NO CONNECT) PIN 5: GNDD PIN 6: DSR (ON) PIN 7: RTS (NO CONNECT) PIN 8: CTS (ON) PIN 9: RI (NO CONNECT)	9
P16	POWER INPUT TERMINAL		11
P17	POWER INPUT JACK	CENTER: VRAW (6-12 VDC) SHELL: GNDD	11

HEADERS (cont.)

HEADER	FUNCTION	COMMENTS	PAGE
P18	ANALOG INPUTS 0 - 3	PIN 1: ANALOG IN 0 PIN 2: GNDA PIN 3: ANALOG IN 1 PIN 4: GNDA PIN 5: ANALOG IN 2 PIN 6: GNDA PIN 7: ANALOG IN 3 PIN 8: GNDA	6
P19	MCU PORT 3		5
P20	MCU PORT 0		4
P21	MCU PORT 2		4
P22	ANALOG INPUTS 4-7	PIN 1: ANALOG IN 4 PIN 2: GNDA PIN 3: ANALOG IN 5 PIN 4: GNDA	6
P23	ANALOG REFERENCES	PIN 1: REF2048 PIN 2: AVREF+ PIN 3: AVREF- PIN 4: GNDA	8
P24	DEMULTIPLEXED ADDRESSES A0 THROUGH A7	PINS 1-8: A0 - A7	4
J2	REAL TIME CLOCK	PIN 1: CLK_SQW PIN 2: CLK_INTA PIN 3: CLK_INTB PIN 4: VLITH	4
J3	ADDRESS DECODER	PIN 1: RD_L PIN 2: WR_L PIN 3: FXXX_L PIN 4: FAXX_L PIN 5: FBXX_L PIN 6: MCS1_L PIN 7: FDXX_RD_L PIN 8: FDXX_WR_L	3
J4	PUSHBUTTONS & LEDS	PIN 1: PUSHBUTTON HIGH PIN 2: PUSHBUTTON LOW PIN 3: LED 3_L PIN 4: LED_4_L	4

MCU PORTS

MCU PORT 0 P20 PAGE 4	PIN 1: P0.0 / AD0 PIN 2: P0.1 / AD1 PIN 3: P0.2 / AD2 PIN 4: P0.3 / AD3 PIN 5: P0.4 / AD4 PIN 6: P0.5 / AD5 PIN 7: P0.6 / AD6 PIN 8: P0.7 / AD7	MCU PORT 4 P1 PAGE 5	PIN 1: P4.0 / CMSR0 PIN 2: P4.1 / CMSR1 PIN 3: P4.2 / CMSR2 PIN 4: P4.3 / CMSR3 PIN 5: P4.4 / CMSR4 (DAC_CS) PIN 6: P4.5 / CMSR5 (SDINP) PIN 7: P4.6 / CMT0 PIN 8: P4.7 / CMT1 (SCLK)
MCU PORT 1 P4 PAGE 5	PIN 1: P1.0 / CT0 / INT2 PIN 2: P1.1 / CT1 / INT3 PIN 3: P1.2 / CT2 / INT4 PIN 4: P1.3 / CT3 / INT5 PIN 5: P1.4 / T2 PIN 6: P1.5 / T2EX PIN 7: P1.6 / RxD1 PIN 8: P1.7 / TxD1	MCU PORT 5 P7 PAGE 8	PIN 1: P5.0 / ADC0 PIN 2: P5.1 / ADC1 PIN 3: P5.2 / ADC2 PIN 4: P5.3 / ADC3 PIN 5: P5.4 / ADC4 PIN 6: P4.5 / ADC5 PIN 7: P5.6 / ADC6 PIN 8: P5.7 / ADC7
MCU PORT 2 P21 PAGE 4	PIN 1: P2.0 / A8 PIN 2: P2.1 / A9 PIN 3: P2.2 / A10 PIN 4: P2.3 / A11 PIN 5: P2.4 / A12 PIN 6: P2.5 / A13 PIN 7: P2.6 / A14 PIN 8: P2.7 / A15	MCU PORT 6 P6 PAGE 5	PIN 1: P6.0 / PWMO0 PIN 2: P6.1 / PWMO1 PIN 3: P6.2 / PWM02 PIN 4: P6.3 / PWMO3 PIN 5: P6.4 / PWMO4 PIN 6: P6.5 / PWMO5 PIN 7: GNDD PIN 8: P6.7 / STADC
MCU PORT 3 P19 PAGE 5	PIN 1: P3.0 / RxD0 PIN 2: P3.1 / TxD0 PIN 3: P3.2 / INT0 PIN 4: P3.3 / INT1 PIN 5: P3.4 / T0 (DOW_DQ) PIN 6: P3.5 / T1 (232_SHDN_L) PIN 7: P3.6 / WR_L PIN 8: P3.7 / RD_L		

JUMPERS

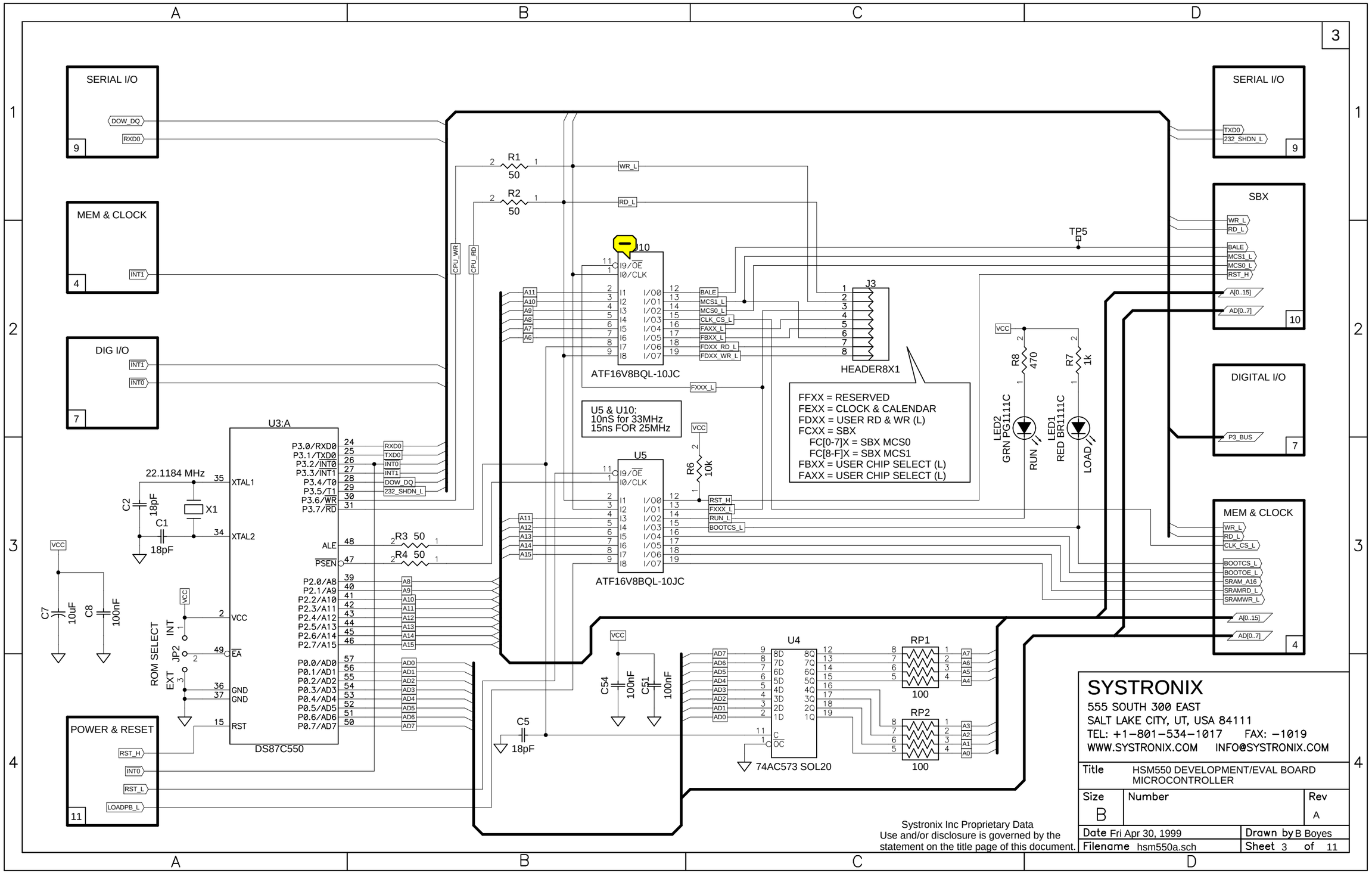
JUMPER	NC?*	POSITIONS	COMMENTS	PAGE
JP1	NO	1-2 2-3	CLK_INTA TO MCU INT1 (TYPICAL) CLK_INTB TO MCU INT1	4
JP2	NO	1-2 2-3	MCU USES INTERNAL ROM MCU USE EXTERNAL ROM	1
JP3	YES	1-2	DAC CHIP SELECT	5
JP4	YES	1-2	SERIAL DATA TO DAC	
JP5	YES	1-2	SERIAL CLOCK TO DAC	
JP6 JP7 JP8 JP9 JP10 JP11 JP12 JP13	NO	 NO JUMPER 1-2 2-3	ANALOG INPUT RANGE SECLECTION AVref- to AVref + -(AVref+ + AVref-) TO AVref+ 0 to 2AVref+	6
JP14	YES	1-2	SBX_INT1 TO MCU INT3	10
JP15	YES	1-2	SBX_INT0 TO MCU INT2	
JP16	YES	1-2	BUFFERED ALE FOR USE WITH SYSTRONIX OR CUSTOM SBX CARDS WHICH REQUIRE ADDRESS/DATA DEMUX.	
JP17 JP18 JP19 JP20 JP21 JP22 JP23 JP24	YES	1-2	ANALOG BUFFER OUTPUTS TO ADC INPUTS.	8
JP25	YES	1-2	POWER FAIL ALARM TO INT0	11
P2	NO	1-2 3-4 5-6 7-8	PUSHBUTTON HIGH TO MCU INT2 PUSHBUTTON HIGH TO MCU INT3 PUSHBUTTON HIGH TO MCU INT4 PUSHBUTTON HIGH TO MCU INT5	5
P3	NO	1-2 3-4	PUSHBUTTON HIGH TO MCU INT2 PUSHBUTTON HIGH TO MCU INT3	

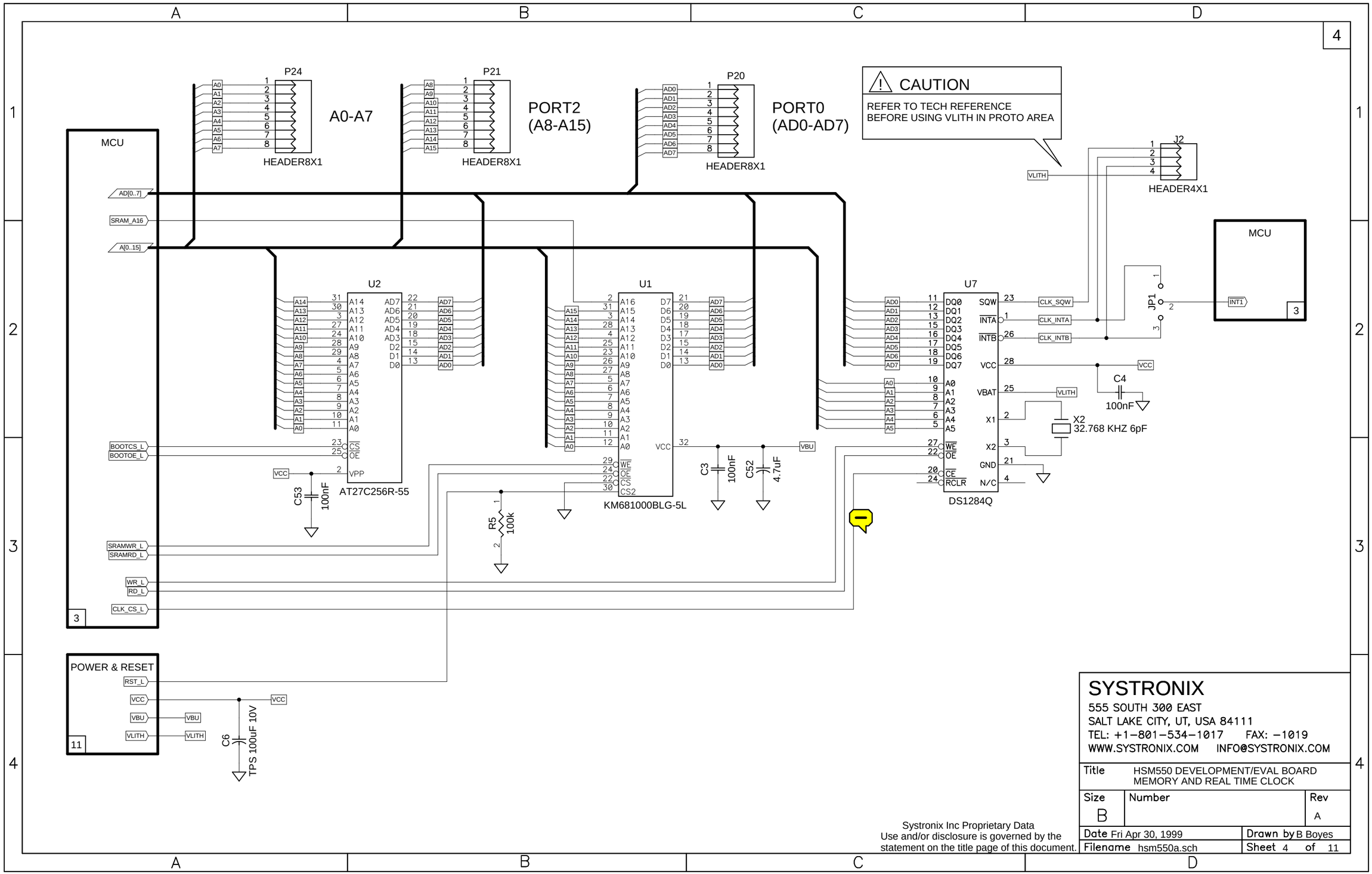
* 'NC' JUMPERS ARE NORMALLY CLOSED. A TRACE CONNECTS PINS 1 & 2 ON THE BACK OF THE BOARD. THIS TRACE CAN BE CUT IF DESIRED AND REJUMPERED ON THE FRONT LATER TO RECONNECT IT.

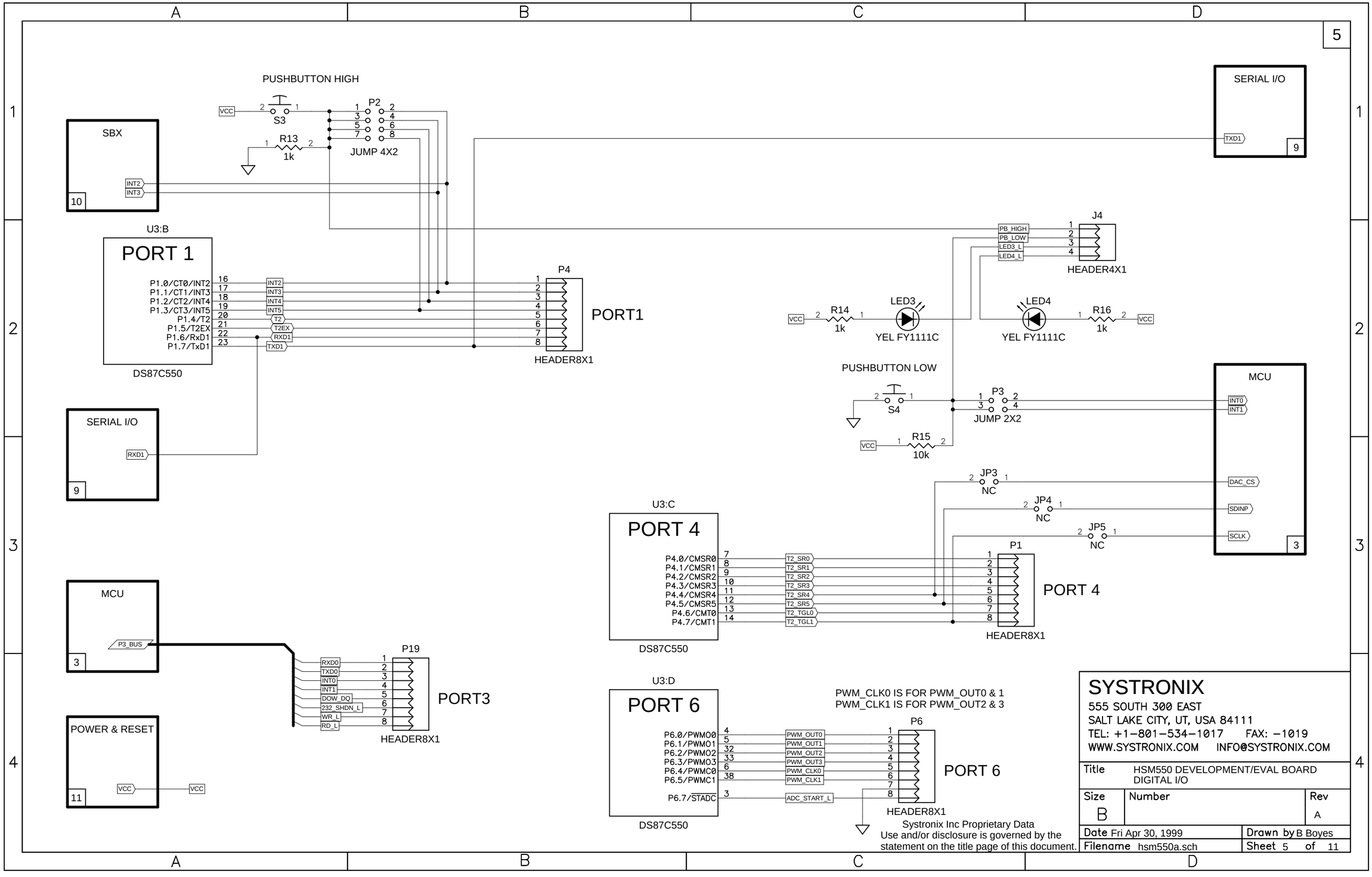
60 KBYTES CODE
60 KBYTES DATA
4 KBYTES I/O SPACE

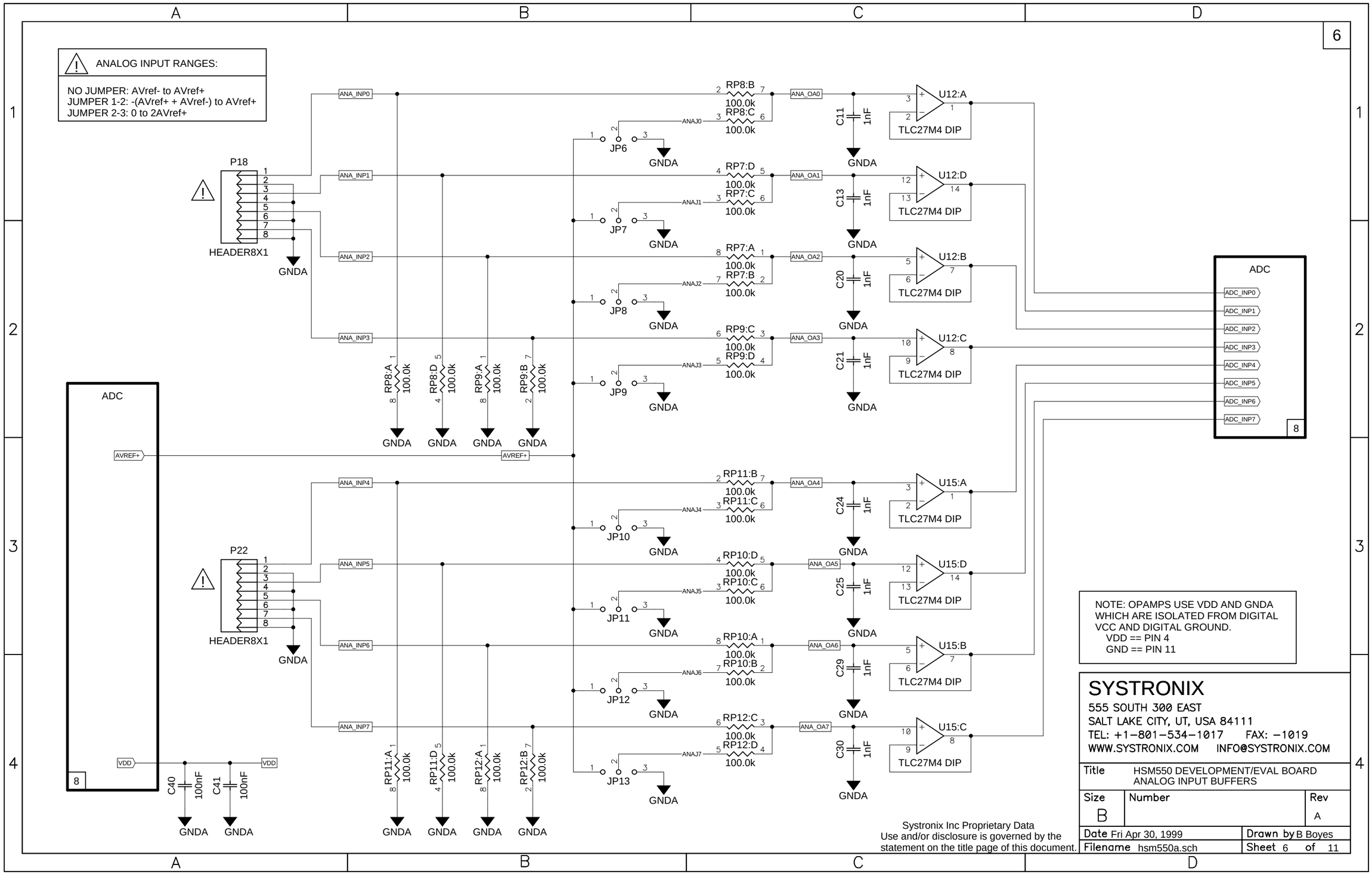
SYSTRONIX
555 SOUTH 300 EAST
SALT LAKE CITY, UT, USA 84111
TEL: +1-801-534-1017 FAX: -1019
WWW.SYSTRONIX.COM INFO@SYSTRONIX.COM

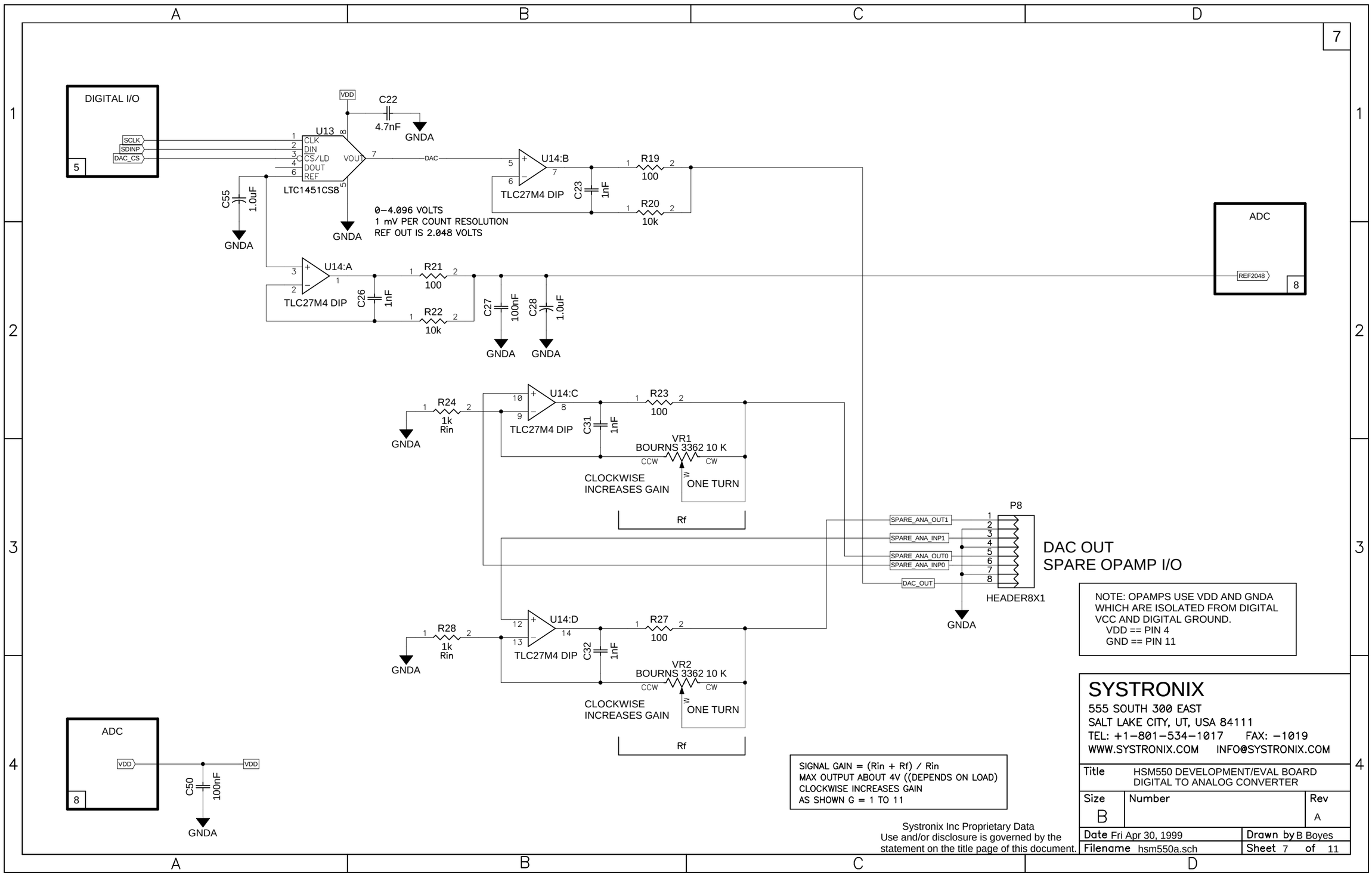
Title	HSM550 DEVELOPMENT/EVAL BOARD NOTES	
Size B	Number	Rev A
Date Fri Apr 30, 1999		Drawn by B Boyes
Filename hsm550a.sch		Sheet 2 of 11

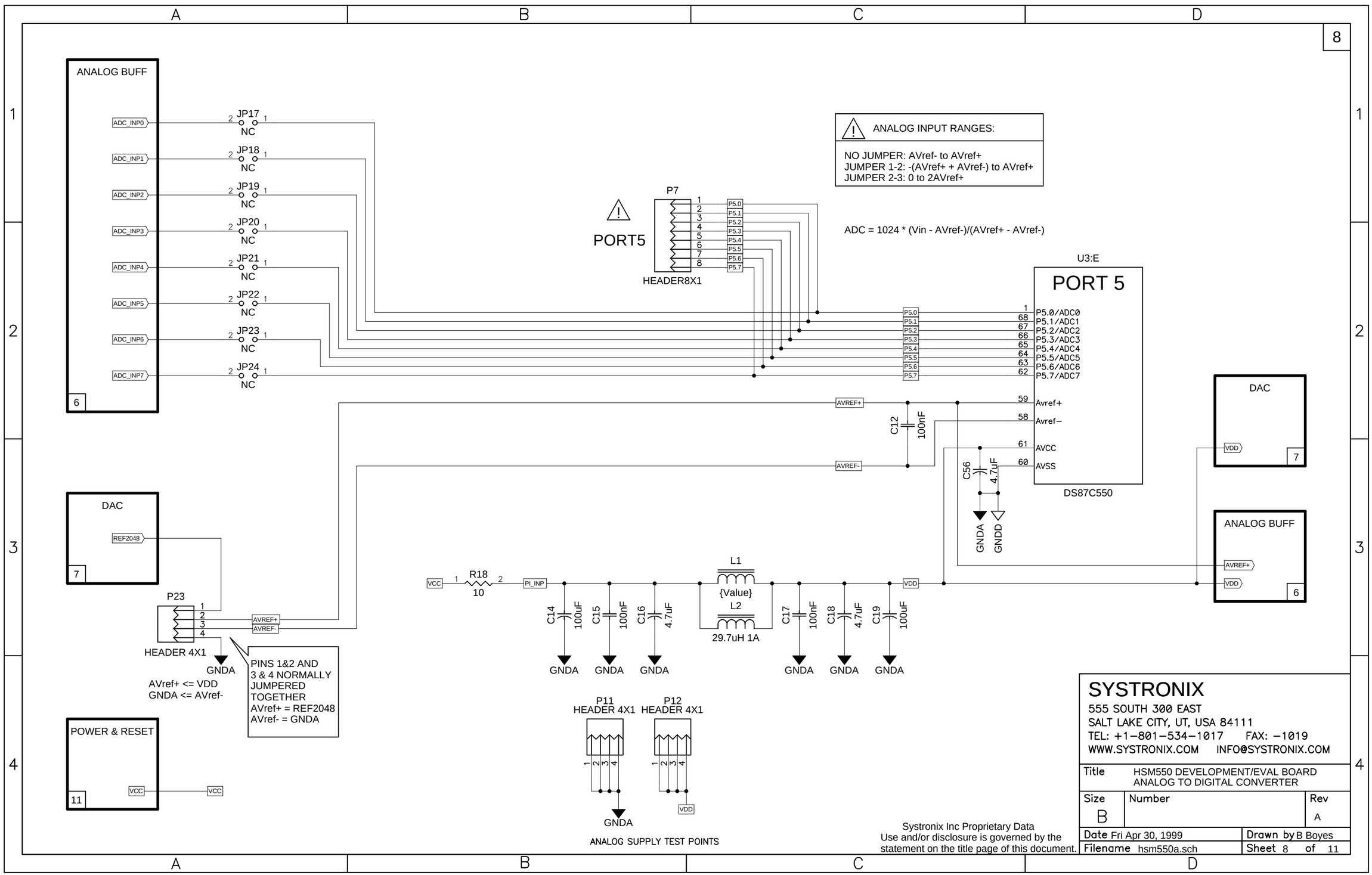

SERIAL I/O
DOW_DQ
RXD0
9
MEM & CLOCK
INT1
4
DIG I/O
INT1
INT0
7
POWER & RESET
RST_H
INT0
RST_L
LOADPB_L
11
SERIAL I/O
TXD0
232_SHDN_L
9
SBX
WR_L
RD_L
BALE
MCS1_L
MCS0_L
RST_H
A[0..15]
AD[0..7]
10
DIGITAL I/O
P3_BUS
7
MEM & CLOCK
WR_L
RD_L
CLK_CS_L
BOOTCS_L
BOOTOE_L
SRAM_A16
SRAMRD_L
SRAMWR_L
A[0..15]
AD[0..7]
4
R1 50
R2 50
R3 50
R4 50
R6 10k
R7 1k
R8 470
WR_L
RD_L
CPU_WR
CPU_RD
U10
ATF16V8BQL-10JC
BALE
MCS1_L
MCS0_L
CLK_CS_L
FAXX_L
FBXX_L
FDXX_RD_L
FDXX_WR_L
FXXX_L
J3
HEADER8X1
TP5
FFXX = RESERVED
FEXX = CLOCK & CALENDAR
FDXX = USER RD & WR (L)
FCXX = SBX
FC[0-7]X = SBX MCS0
FC[8-F]X = SBX MCS1
FBXX = USER CHIP SELECT (L)
FAXX = USER CHIP SELECT (L)
U5 & U10:
10nS for 33MHz
15ns FOR 25MHz
U5
ATF16V8BQL-10JC
RST_H
FXXX_L
RUN_L
BOOTCS_L
VCC
LED2
GRN PG1111C
RUN
LED1
RED BR1111C
LOAD
U3:A
DS87C5550
22.1184 MHz
X1
C1 18pF
C2 18pF
C7 10uF
C8 100nF
ROM SELECT
INT
JP2
EXT
C5 18pF
C51 100nF
C54 100nF
U4
74AC573 SOL20
RP1
RP2
100
100
SYSTRONIX
555 SOUTH 300 EAST
SALT LAKE CITY, UT, USA 84111
TEL: +1-801-534-1017 FAX: -1019
WWW.SYSTRONIX.COM INFO@SYSTRONIX.COM
Title HSM550 DEVELOPMENT/EVAL BOARD MICROCONTROLLER
Size B
Number
Rev A
Date Fri Apr 30, 1999
Drawn by B Boyes
Filename hsm550a.sch
Sheet 3 of 11
Systronix Inc Proprietary Data
Use and/or disclosure is governed by the statement on the title page of this document.

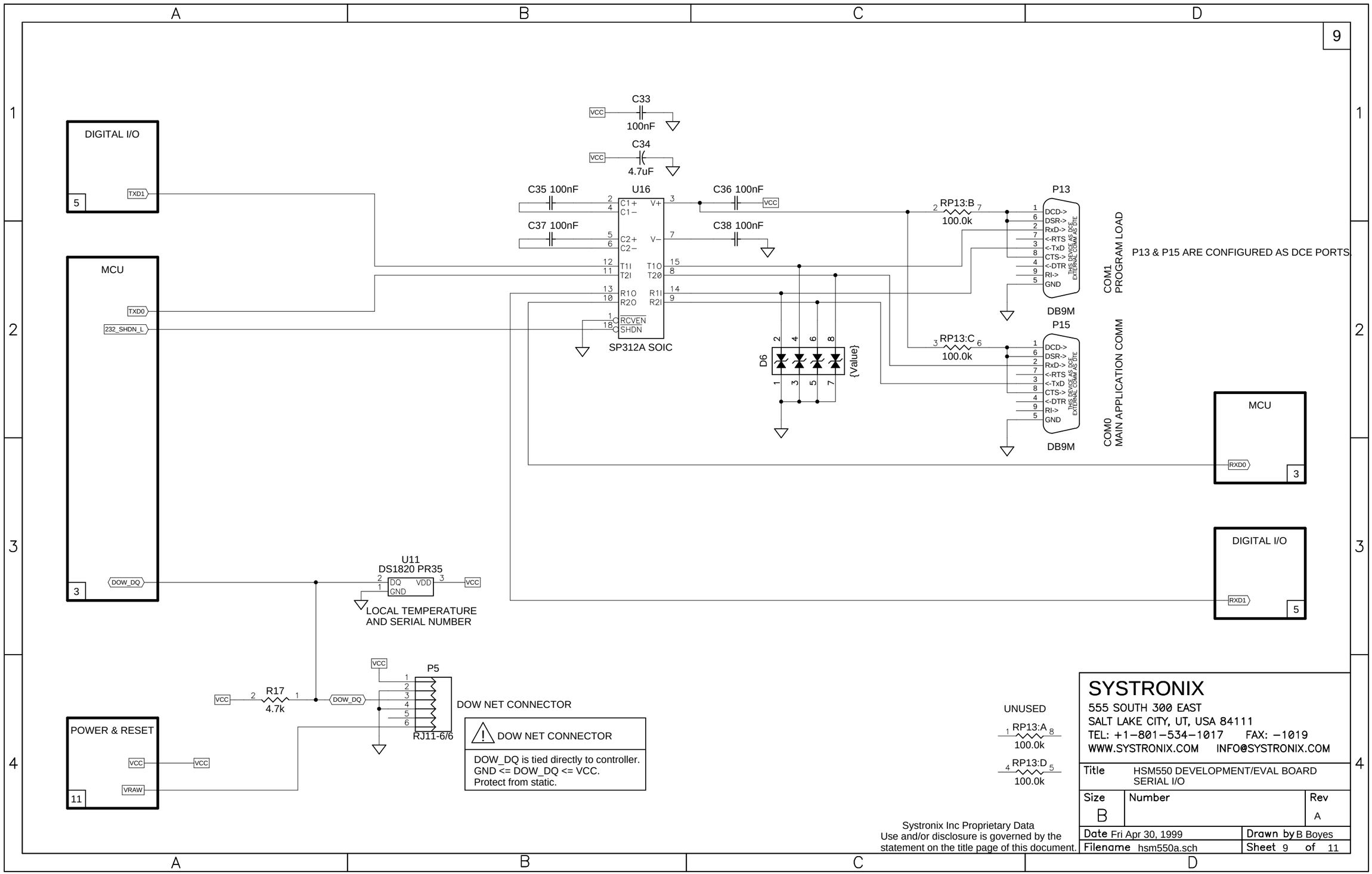


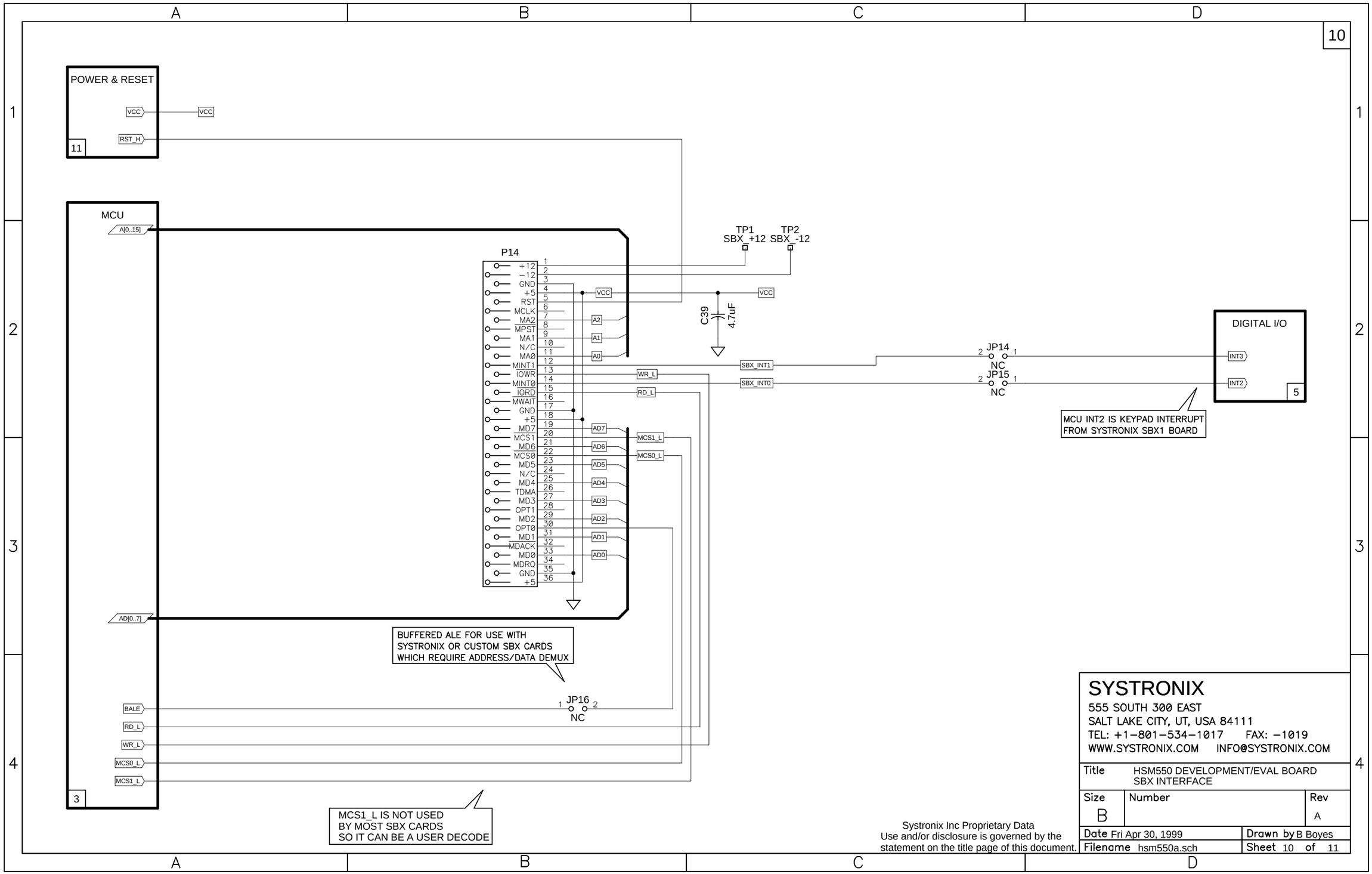

A
B
C
D
1
2
3
4
5
PUSHBUTTON HIGH
VCC
S3
P2
R13
1k
JUMP 4X2
SBX
INT2
INT3
10
SERIAL I/O
TXD1
9
U3:B
PORT 1
P1.0/CT0/INT2
P1.1/CT1/INT3
P1.2/CT2/INT4
P1.3/CT3/INT5
P1.4/T2
P1.5/T2EX
P1.6/RxD1
P1.7/TxD1
DS87C550
P4
PORT1
HEADER8X1
J4
PB_HIGH
PB_LOW
LED3_L
LED4_L
HEADER4X1
R14
1k
LED3
YEL FY1111C
LED4
YEL FY1111C
R16
1k
PUSHBUTTON LOW
S4
P3
JUMP 2X2
R15
10k
MCU
INT0
INT1
DAC_CS
SDINP
SCLK
3
JP3
NC
JP4
NC
JP5
NC
SERIAL I/O
RXD1
9
U3:C
PORT 4
P4.0/CMSR0
P4.1/CMSR1
P4.2/CMSR2
P4.3/CMSR3
P4.4/CMSR4
P4.5/CMSR5
P4.6/CMT0
P4.7/CMT1
T2_SR0
T2_SR1
T2_SR2
T2_SR3
T2_SR4
T2_SR5
T2_TGL0
T2_TGL1
DS87C550
P1
PORT 4
HEADER8X1
MCU
P3_BUS
3
P19
RXD0
TXD0
INT0
INT1
DOW_DQ
232_SHDN_L
WR_L
RD_L
PORT3
HEADER8X1
U3:D
PORT 6
P6.0/PWM00
P6.1/PWM01
P6.2/PWM02
P6.3/PWM03
P6.4/PWMC0
P6.5/PWMC1
P6.7/STADC
PWM_OUT0
PWM_OUT1
PWM_OUT2
PWM_OUT3
PWM_CLK0
PWM_CLK1
ADC_START_L
DS87C550
PWM_CLK0 IS FOR PWM_OUT0 & 1
PWM_CLK1 IS FOR PWM_OUT2 & 3
P6
PORT 6
HEADER8X1
POWER & RESET
VCC
11
Systronix Inc Proprietary Data
Use and/or disclosure is governed by the
statement on the title page of this document.
SYSTRONIX
555 SOUTH 300 EAST
SALT LAKE CITY, UT, USA 84111
TEL: +1-801-534-1017 FAX: -1019
WWW.SYSTRONIX.COM INFO@SYSTRONIX.COM
Title HSM550 DEVELOPMENT/EVAL BOARD
DIGITAL I/O
Size B
Number
Rev A
Date Fri Apr 30, 1999
Drawn by B Boyes
Filename hsm550a.sch
Sheet 5 of 11


ANALOG INPUT RANGES:
NO JUMPER: AVref- to AVref+
JUMPER 1-2: -(AVref+ + AVref-) to AVref+
JUMPER 2-3: 0 to 2AVref+
P18
HEADER8X1
P22
HEADER8X1
GNDA
ANA_INP0
ANA_INP1
ANA_INP2
ANA_INP3
ANA_INP4
ANA_INP5
ANA_INP6
ANA_INP7
ANA_OA0
ANA_OA1
ANA_OA2
ANA_OA3
ANA_OA4
ANA_OA5
ANA_OA6
ANA_OA7
ANAJ0
ANAJ1
ANAJ2
ANAJ3
ANAJ4
ANAJ5
ANAJ6
ANAJ7
RP8:B 100.0k
RP8:C 100.0k
RP7:D 100.0k
RP7:C 100.0k
RP7:A 100.0k
RP7:B 100.0k
RP9:C 100.0k
RP9:D 100.0k
RP8:A 100.0k
RP8:D 100.0k
RP9:A 100.0k
RP9:B 100.0k
RP11:B 100.0k
RP11:C 100.0k
RP10:D 100.0k
RP10:C 100.0k
RP10:A 100.0k
RP10:B 100.0k
RP12:C 100.0k
RP12:D 100.0k
RP11:A 100.0k
RP11:D 100.0k
RP12:A 100.0k
RP12:B 100.0k
JP6
JP7
JP8
JP9
JP10
JP11
JP12
JP13
C11 1nF
C13 1nF
C20 1nF
C21 1nF
C24 1nF
C25 1nF
C29 1nF
C30 1nF
C40 100nF
C41 100nF
U12:A
U12:D
U12:B
U12:C
U15:A
U15:D
U15:B
U15:C
TLC27M4 DIP
AVREF+
VDD
ADC
ADC_INP0
ADC_INP1
ADC_INP2
ADC_INP3
ADC_INP4
ADC_INP5
ADC_INP6
ADC_INP7
NOTE: OPAMPS USE VDD AND GNDA WHICH ARE ISOLATED FROM DIGITAL VCC AND DIGITAL GROUND.
VDD == PIN 4
GND == PIN 11
SYSTRONIX
555 SOUTH 300 EAST
SALT LAKE CITY, UT, USA 84111
TEL: +1-801-534-1017 FAX: -1019
WWW.SYSTRONIX.COM INFO@SYSTRONIX.COM
Title HSM550 DEVELOPMENT/EVAL BOARD
ANALOG INPUT BUFFERS
Size B
Number
Rev A
Date Fri Apr 30, 1999
Drawn by B Boyes
Filename hsm550a.sch
Sheet 6 of 11
Systronix Inc Proprietary Data
Use and/or disclosure is governed by the statement on the title page of this document.









SYSTRONIX
555 SOUTH 300 EAST
SALT LAKE CITY, UT, USA 84111
TEL: +1−801−534−1017 FAX: −1019
WWW.SYSTRONIX.COM INFO@SYSTRONIX.COM

Title	HSM550 DEVELOPMENT/EVAL BOARD SBX INTERFACE	
Size B	Number	Rev A
Date Fri Apr 30, 1999		Drawn by B Boyes
Filename hsm550a.sch		Sheet 10 of 11

P9
HEADER 4X1

P10
HEADER 4X1

GNDD

VCC

DIGITAL SUPPLY TEST POINTS

CR2032 Lithium Battery
220 mAh, 10 year projected
life power on or off. Replace
every 5 years for best results.

Li battery current, typical
Power on –280 nA
Power off 440 nA

B1
CR2032H IN HOLDER

TP3

R25
100

TP4

4.65V RESET
ICC 5 UA MAX, TYP 50 NA

U6
MAX703CSA
VCC
VBAT
PFI
MR
VOUT
RST
PFW
GND

VBU

MEM & CLOCK
VLITH
VBU
RST_L
4

MCU
INT0
RST_L
LOADPB_L
3

R26
2 M

Rin
R9
33.2k 1%

VRAW

VPFI

PFW_L

R10
390k

JP25
NC

PFI TRIPS AT 1.25V

Rp
R11
10.0k 1%

2N4403 SOT–23
{Value}
Q1

C9
100nF

Rin = Rp((Vtrip/1.25)–1)
for Vtrip = 5.5V and Rp = 10k, Rin = 33k
use 33.2k 1%

LOAD

S1

R12
10k

C10
100nF

RESET

S2

P17
DJ-005-B

6 TO 12 V INPUT

P16
PHOENIX2 100

VINP

D3
CMSH2-40

D4
SMBJ24M 600W

VRAW

C42
22uF 35V LOW ESR

C43
22uF 35V LOW ESR

C44
1nF

U17
LTC1624
SENSE–
ITH/RUN
VFB
GND
VIN
BOOST
TG
SW

R31
0.10 1% 1/2W

F1
Si4412DY

C46
100nF

R32
10k

C45
470pF

C47
100pF

R34
32.4k 1%

R33
10.0k 1%

D5
CMSH2-40

VCC SENSE

L3
29.7uH 1A

5V 1A MAX

C48
TPS 100uF 10V

C49
TPS 100uF 10V

OTHER SHEETS
VCC
3 4
5 8
9 10

FOOT1
FOOT2
FOOT3
FOOT4

SYSTRONIX
555 SOUTH 300 EAST
SALT LAKE CITY, UT, USA 84111
TEL: +1–801–534–1017 FAX: –1019
WWW.SYSTRONIX.COM INFO@SYSTRONIX.COM

Title	HSM550 DEVELOPMENT/EVAL BOARD POWER & RESET	
Size B	Number	Rev A.01
Date Mon Jun 21, 1999		Drawn by B Boyes
Filename hsm550a02.sch		Sheet 11 of 11