Intel® StrongARM® SA-1111 Microprocessor Development Module

User's Guide

ADVANCE INFORMATION

January 2000

Phase 4

Notice: This document contains information on products in the sampling and initial production phases of development. Revised information will be published when this product is available.

Order No: 278281-003

Contents

Figures

Tables

Introduction 1

The Intel® StrongARM® SA-1111 Microprocessor Companion Chip (SA-1111) is a companion chip to the SA-1110, providing a variety of functions suitable for use in a high-performance handheld computer system. The SA-1111 brings a new level of integration to small systems, providing a variety of I/O functions that enable complete systems to be built with very little "glue" logic. In addition to a complete USB Host Controller, the SA-1111 includes extensive support for PCMCIA and Compact Flash (CF), two PS/2 ports, two industry-standard serial ports, and other I/O capabilities. It can acquire the system memory bus and do DMA transfers to system memory (SDRAM) with its high-performance memory controller. For more information about the SA-1111 device, see the *Intel® StrongARM® SA-1111 Companion Chip Developer's Manual*, order number 278242.

The Intel® StrongARM® SA-1110 Microprocessor (SA-1110) is a highly integrated communications microcontroller that incorporates a 32-bit StrongARM® RISC processor core, system support logic, multiple communication channels, an LCD controller, a memory and PCMCIA controller, and general-purpose I/O ports. For more information about the SA-1110 device, see the *Intel® StrongARM® SA-1110 Microprocessor Advanced Developer's Manual,* order number 278240.

The Intel® StrongARM® SA-1110 Development Platform (SA-1110 Development Platform) is based upon these devices and is composed of a three board set:

- Intel® StrongARM® SA-1110 Development Board[1] (SA-1110 Development Board) order number SA1110DEVBD.
- Intel® StrongARM® SA-1111 Companion Chip Development module (SA-1111 development module) order number SA1111DEVMOD.
- Graphics accelerator board (available from third party vendors)

The SA-1110 Development Platform has many purposes:

- Hand held applications development and reference design
- Windows CE applications development platform
- General OS applications development platform
- RF communications development platform

Note: This document and module are for Phase 4 of this product. For the latest information and updates, see the hardware release notes that are provided in hardcopy format, and the software readme.txt files that are provided in the software kits.

Figure 1-1 shows the preliminary front view of the SA-1110 Development Platform and Figure 1-2 shows a preliminary cross-sectional view.

1. For more information about the SA-1110 Development Board, see the *Intel® StrongARM® SA-1110 Microprocessor Development Board User's Guide.*

Figure 1-1. Preliminary Intel® StrongARM® SA-1110 Developer Platform

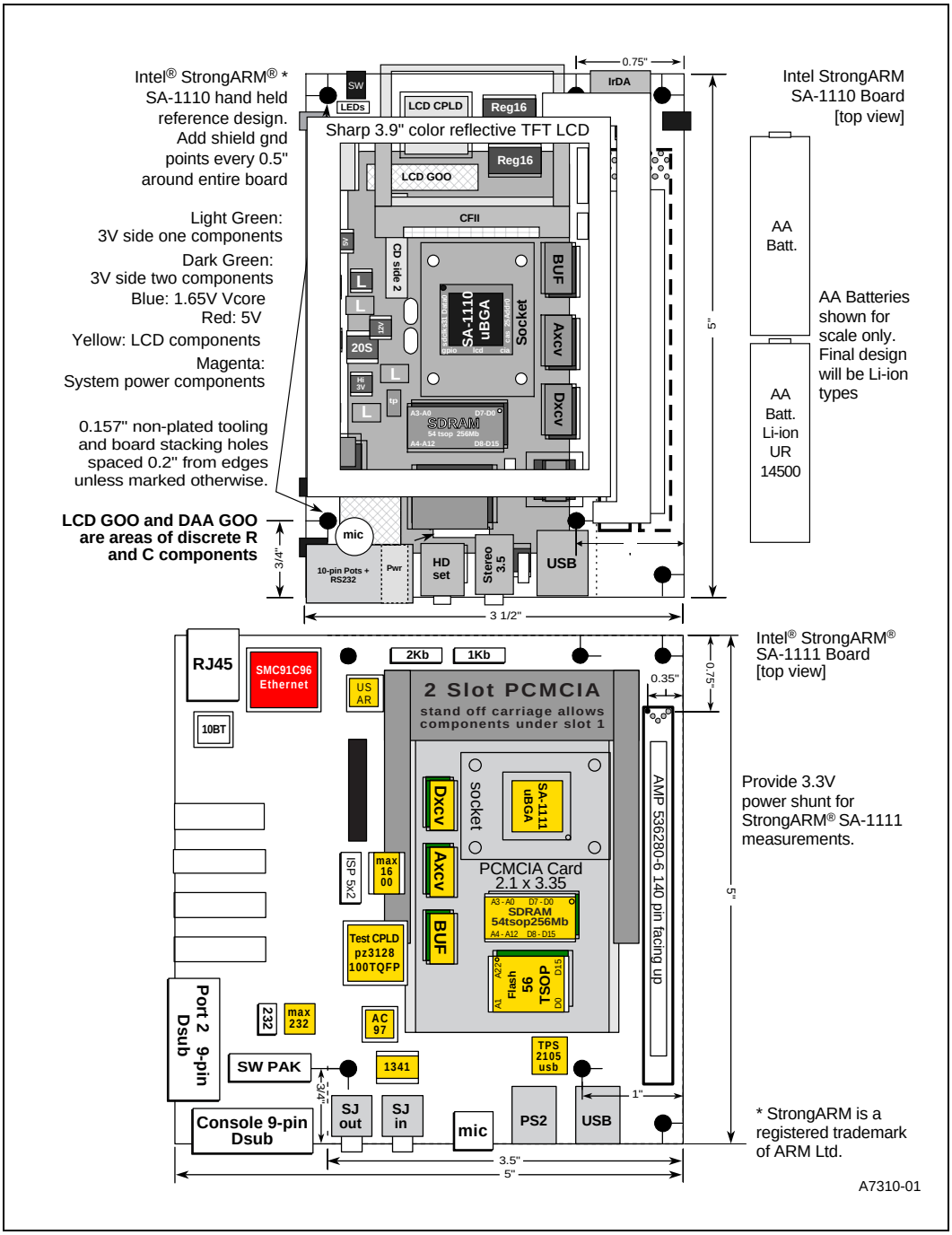
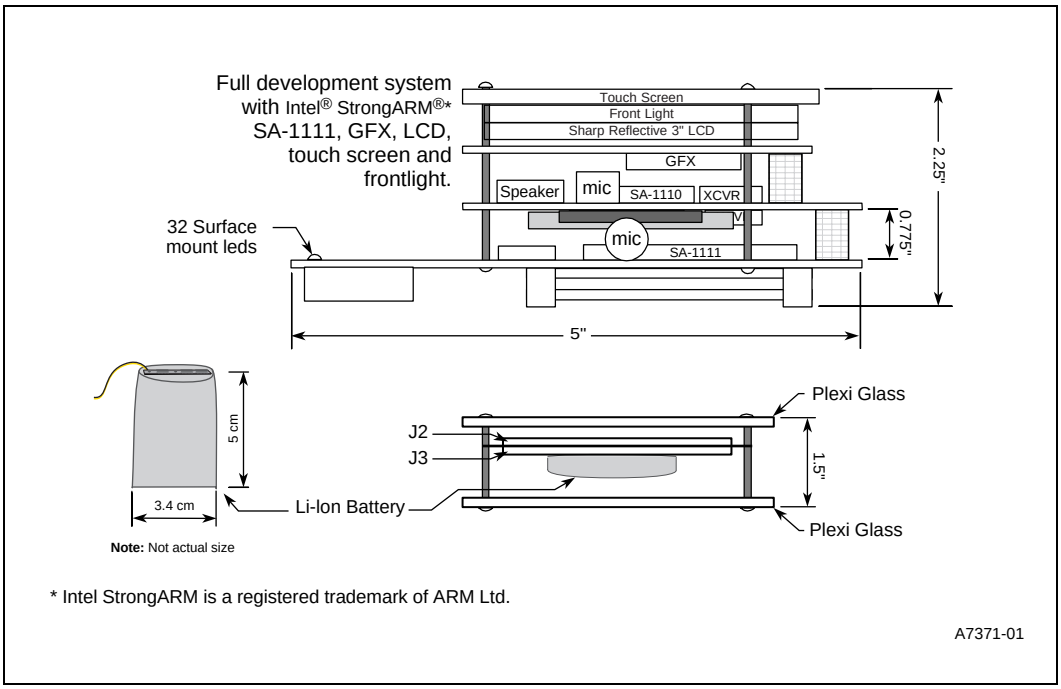


Figure 1-2 shows a preliminary sideview of the SA-1110 development board:

Figure 1-2. Preliminary Cross-Sectional view of SA-1110 Development Platform



The modular design of the SA-1110 Development Platform makes it a flexible, extendable and reusable design that supports[1] the following requirements:

- Targeted for hand-held applications
- Multiple OS development platform
- Third-party vendor graphics accelerator development platform
- Development platform for SA-1111 Companion Chip
- SA-1110 Development Board may be used as a palm PC reference design and development platform.
- SA-1110 Development Board may be used as host board for PCI bridge ASIC/FPGA and a Windows CE reference platform
- Battery powered design example
- Power management and battery management including SMBus and SBS development platform
- RF communications development board with interfaces for CDMA and GSM and Bluetooth[2] RF modules

1. The SA-1110 Development Platform has not been certified as being compliant with FCC, CE, UC, or PTT telephone standards or regulations.
2. For the phase 1 release of this document, formal Bluetooth interface specifications were not available. For more information, see section Section 4.13.3.

1.1 Related Documentation

Other documentation that may be helpful while reading this document are described in the following table:

Title	Location
Intel® StrongARM® SA-1110 Microprocessor Advanced Developer's Manual, order number: 278240	Intel's website for developers is at: http://developer.intel.com
Intel® StrongARM® SA-1110 Microprocessor Specification Update, order number: 278259	
Intel® StrongARM® SA-1110 Development Board Parts List, order number: 278280	
Intel® StrongARM® SA-1110 Development Board User's Guide, order number: 278278	
Intel® StrongARM® SA-1110 Development Board Schematics, order number: 278279	
Intel® StrongARM® SA-1111 Microprocessor Companion Chip Developer's Manual, order number: 278242	
Intel® StrongARM® SA-1111 Microprocessor Companion Chip Specification Update, order number: 278260	
Intel® StrongARM® SA-1111 Companion Chip Development Board Schematics, order number: 289282	
Intel® StrongARM® SA-1111 Companion Chip Development Board Parts List, order number: 278283	
The ARM Debug Monitor: Angel	ARM's website is at: http://www.arm.com
ARM Architecture Reference Manual	

1.2 SA-1110 Development Platform Overview

The SA-1110 Development Board, the SA-1111 Development Module, and the graphics accelerator boards are designed to be a flexible software and hardware development environment and example design. When the SA-1110 Development Board is used as a development system, the SA-1111 Development Module is also required to provide Ethernet, serial port, LEDs and logic analyzer support.

The SA-1110 Development Board has the following resources and features:

- Up to 32MB socketed fast page mode Intel® StrataFlash™ memory
- Up to 64MB 100 MHz SDRAM
- Un-buffered main memory interface allows highest possible SDRAM memory bandwidth
- Small form factor 3.5" x 5" x 0.5" (same size as the Nino*)
- UCB1300 codec supports microphone, speaker, POTS line soft modem DAA connections and touch screen
- UDA1341 stereo codec supports high quality 16 bit stereo audio record and playback

- Infra-red interface for IrDA data links up to 4Mb
- Battery powered, using high efficiency DC-DC converters and a single Lithium ion (Li-ion) cell
- Smart battery technology development platform and reference design when used with smart battery packs
- Integrated 3.9” reflective color TFT LCD and touch screen
- Optional 8” color passive LCD for use with GFX companion chip
- Two expansion headers for the SA-1111 Development Module, the graphics accelerator board, or other boards
- Built in mini speaker and two microphones
- Built in quiet alert vibration motor
- One type II Compact Flash socket (available only when the SA-1111 Development Module and graphics accelerator boards are not present)
- USB slave port for PC synchronization and battery trickle charge
- RF module interface connector for CDMA, GSM and Bluetooth modules up to 4 Watts
- Base station connector for JTAG programming, RS232, power input, and telephone

The SA-1111 Development Module has the following features[1]:

- Expansion Flash bank. Up to 32MB socketed fast page mode StrataFlash memory
- Expansion SDRAM bank (up to 64MB 50MHz SDRAM)
- Supports UDA1341 stereo codec for high quality 16 bit stereo audio record and playback
- Supports AC97 stereo codec for high quality 16 bit stereo audio record and playback
- Form factor is six inches by five inches by 1.38 inches
- Built in microphone
- One type II PCMCIA socket (inner slot)
- One type II Compact Flash socket (outer slot)
- USB host port
- USB host plug and power control device
- PS\2 port for mouse and trackpad
- Logic analyzer pods
- 32 LEDs to display program debug data
- 10BASE-T Ethernet controller
- Two debug serial ports (RS-232)

1. For more information about the SA-1111 Development Module, see *the Intel® StrongARM® SA-1111 Companion Chip Development Module User’s Guide*.

The graphics accelerator board has the following features[1]:

- Third-party high performance graphics accelerator
- Two head display support; Analog XGA and LCD XGA
- Direct connection for Sharp LM8V31 dual scan STN VGA color panel with backlight and touch screen

1. For more information about the graphics accelerator board, see the third-party documentation.

Getting Started 2

This SA-1111 Development Module is supplied as a daughter card, which is an optional development module for the SA-1110 Development Board. This chapter provides a physical description of the SA-1111 Development Module and describes how to:

- Unpack the cards and give them a visual inspection
- Install required hardware
- Install required software
- Select the various card modes
- Configure the card to suit your application
- Power-up the cards for the first time

2.1 Physical Description

Figure 2-1 shows the physical layout of the SA-1111 Development Module. The SA-1111 Development Module is a 10-layer double-sided board with surface mount assembly technology. The SA-1111 Development Module contains the SA-1111 companion chip, expansion flash, expansion SDRAM, audio codecs, various serial I/O connections, PCMCIA and Compact Flash support, an in-circuit programmable CPLD pre-programmed for control and additional debug capability, and an interface to the SA-1110 Development Board. The default setting for switches and jumpers are defined in Table 4-2.

The following are debug specific features which are isolated from the rest of the board:

- Nine-pin DB9 RS-232 communications port connector with full modem support for system debug and software development.
- Ten-pin header RS-232 second communications port with full modem support that can be used for hardware flow control or for passing debug messages from the development environment.
- Ethernet Controller SMC LAN91C96 for development environment communications.
- 32 LEDs for user-defined status and system debug display.
- Four buffered 38-pin Mictor logic analyzer connectors.
- 32 LEDs for status display

Figure 2-1. SA-1111 Development Module (Side 1)

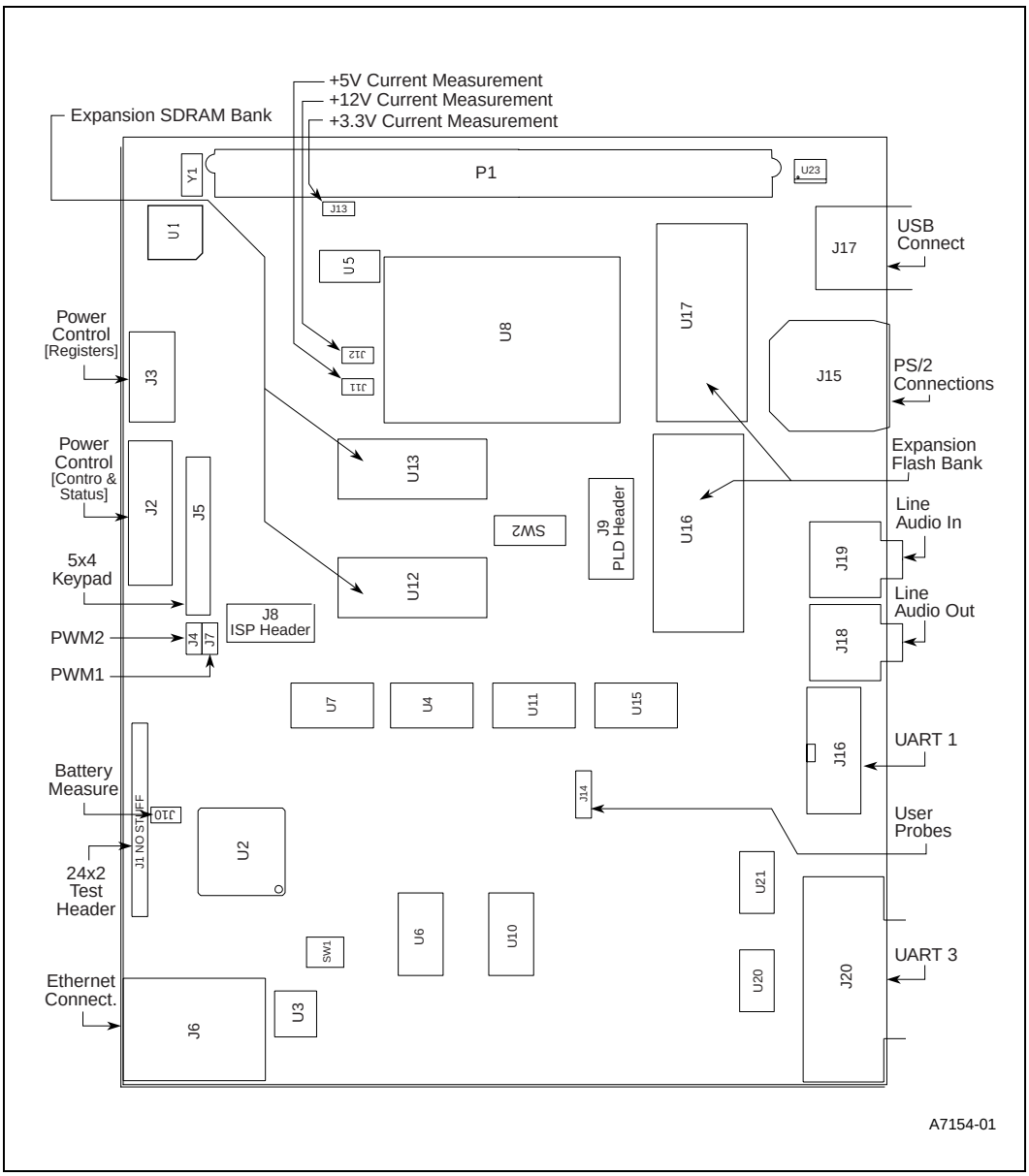
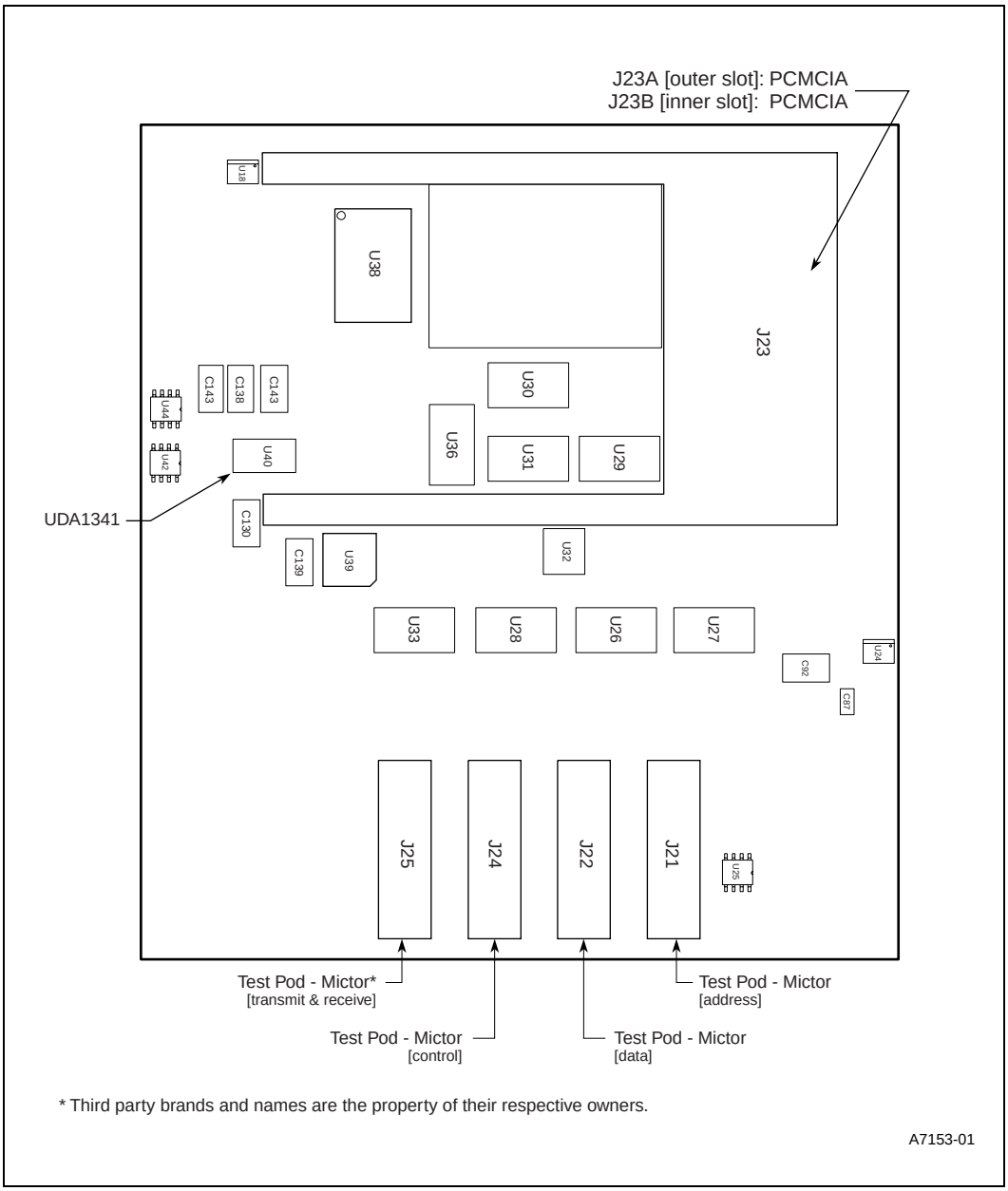


Figure 2-2. SA-1111 Development Module (Side 2)



2.1 Unpacking the Intel® StrongARM® SA-1111 Development Module

Caution: This board contains electronic components that are susceptible to permanent damage from electrostatic discharge (static electricity). To prevent electrostatic discharge, it is supplied in an

antistatic bag. When handling the board, risk of damage can be alleviated by following a few simple precautions:

- Remove the board from the bag only when you are working on an antistatic, earthed surface and wearing an earthed antistatic wrist strap.
- Keep the antistatic bag that the card was supplied in; if you remove the board from a system, store it back in the bag.
- Do not touch the gold contacts.

2.1.1 Intel® StrongARM® SA-1111 Development Module Software

To purchase an ARM Software Development Kit (SDT), see your Intel sales representative.

The following source and executable files are available from the StrongARM section in the developer's area on the Intel website.

- Sample source code including I/O drivers such as the software video-processing engine drivers and the SCB library available through Intel's developer's web site.
- Angel boot loader—Software component of ARM that loads an application from a remote host computer or from the application flash
- Set of microHAL libraries (to be used with Angel)—Set of drivers for communicating with the SA-1100 multimedia development board
- Diagnostics---Test program that analyze the functions of the SA-1110 Development Board.

Note: All software is available from the StrongARM section in the developer's area on the Intel website.

2.2 Hardware Installation

After unpacking and inspecting the contents of the SA-1111 Development Module and the SA-1110 Development Board packages, the SA-1111 Development Module must be mounted to the SA-1110 Development Board.

Caution: The 140-pin connectors are very fragile on the SA-1110 Development Board and the SA-1111 Development Module. Care must be taken when assembling these modules. If the 140-pin connector is damaged due to an improper assembly, the module(s) must be replaced.

2.2.1 Connecting and Disconnecting the Modules

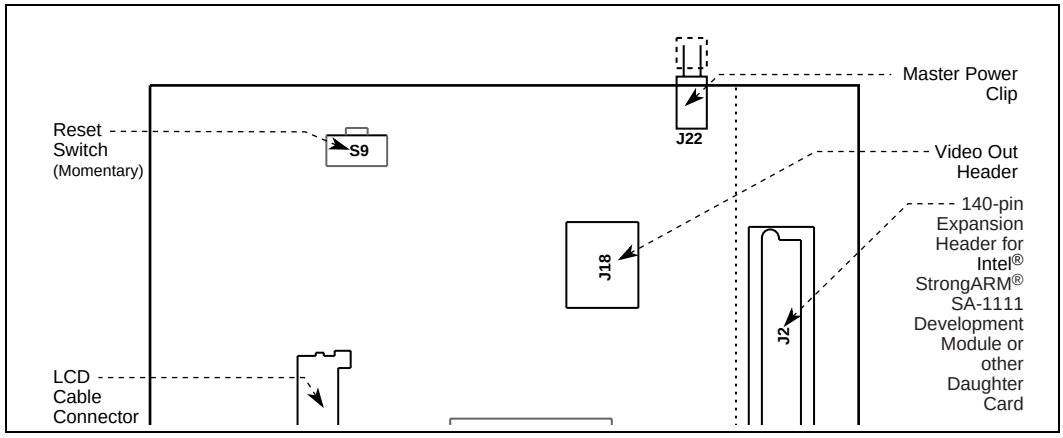
This section describes how to connect and disconnect the modules.

Note: The following procedure assembles the SA-1110 Development Platform providing easy access to components and test points. The SA-1110 Development Platform can also be assembled in a

compact form factor as shown in Figure 1-2. The same method is used for either installation except that the SA-1110 Development Board is inverted for compactness.

1. On the SA-1110 Development Board, verify that the jumper (shunt) for J22 is **not** on the Master Power Clip (J22 should appear as two bare pins).

Figure 2-3. Master Power Clip Location on the SA-1110 Development Board

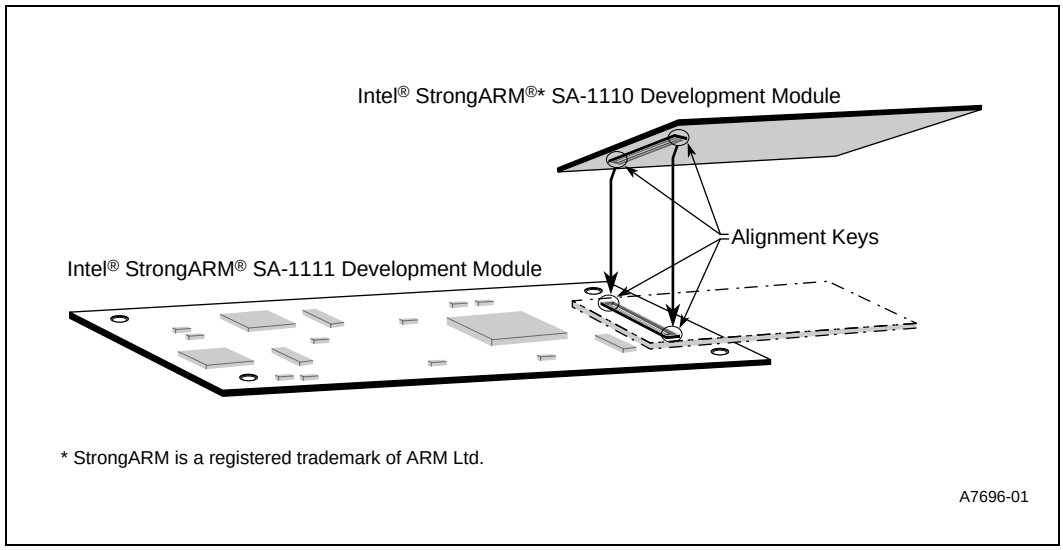


2. Verify that the Compact Flash adapter card has been installed in the inner slot of the PCMCIA socket J23B.

Note: Because of the extreme low power requirements, special precautions for airflow are not required.

3. Align the keys of the 140-pin AMP expansion connector P1 on the SA-1111 Development Module to the mating expansion connector on the SA-1110 Development Board at the board edges as shown in Figure 2-4.

Figure 2-4. Aligning Connectors



4. Nearest the board edges, gently press together and install the mounting hardware as shown in Figure 2-5 and Figure 2-6. Slide the connector in until it is firmly seated.

Note: The opposite facing second expansion connector on the SA-1110 Development Board can be used by an optional third-party graphics module.

Figure 2-5. Connecting the Modules–Top View

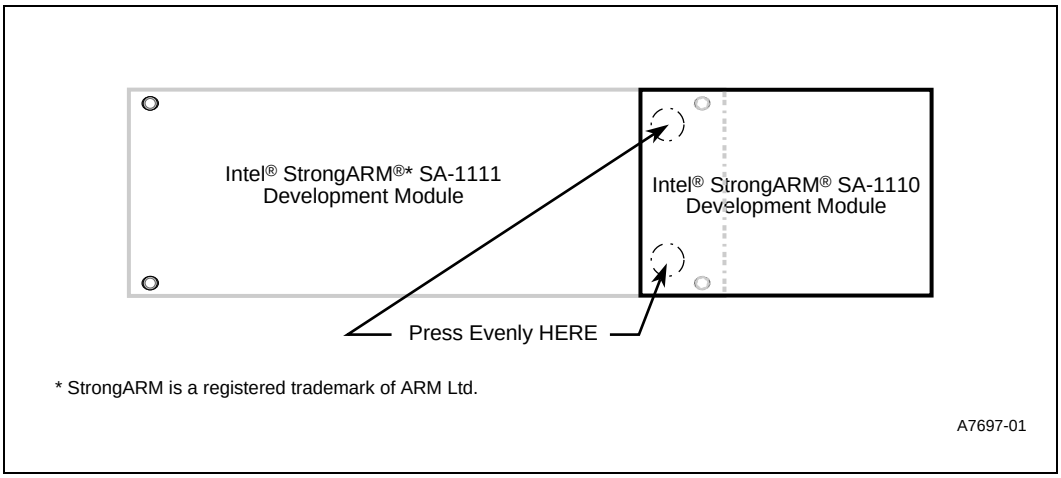
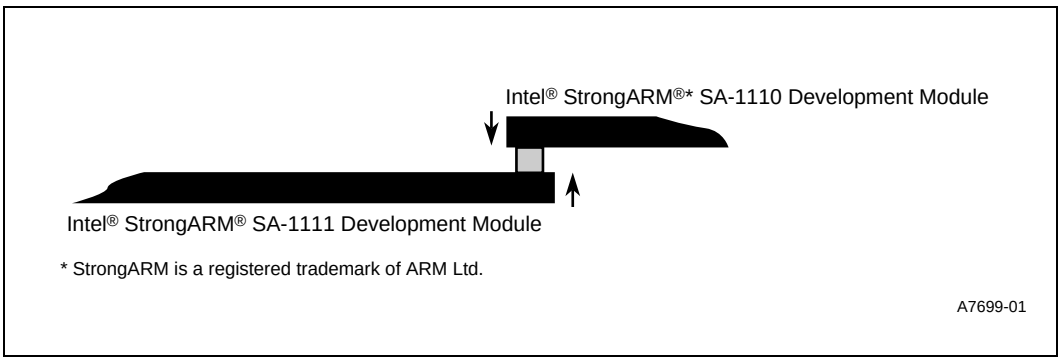


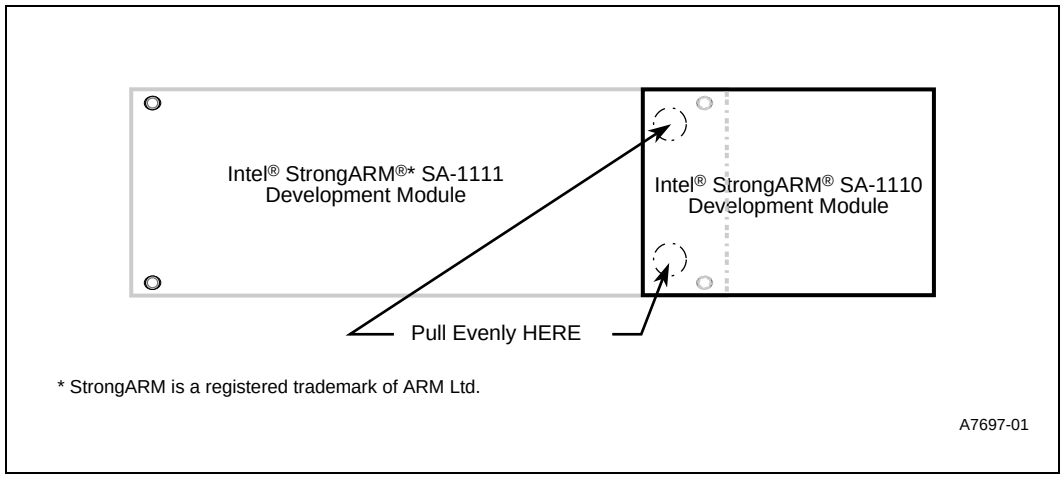
Figure 2-6. Connecting the Modules–Side View



5. To remove the SA-1111 Development Module, evenly pull the modules apart as shown in Figure 2-7.

Note: Twisting the modules apart or applying uneven pressure can damage the connectors.

Figure 2-7. Pulling the Connectors Apart



2.2.2 Power

The SA-1111 Development Module uses a small form factor that resembles a Palm-size PC and draws power from the 140 pin connector (see sheet 3 of 11).

For information about how to power up the SA-1110 Development Platform and for information about LI-Ion (Li-Ion) batteries and battery management circuitry, see the *Intel® StrongARM® SA-1110 Development Board User's Guide*.

Theory of Operation 3

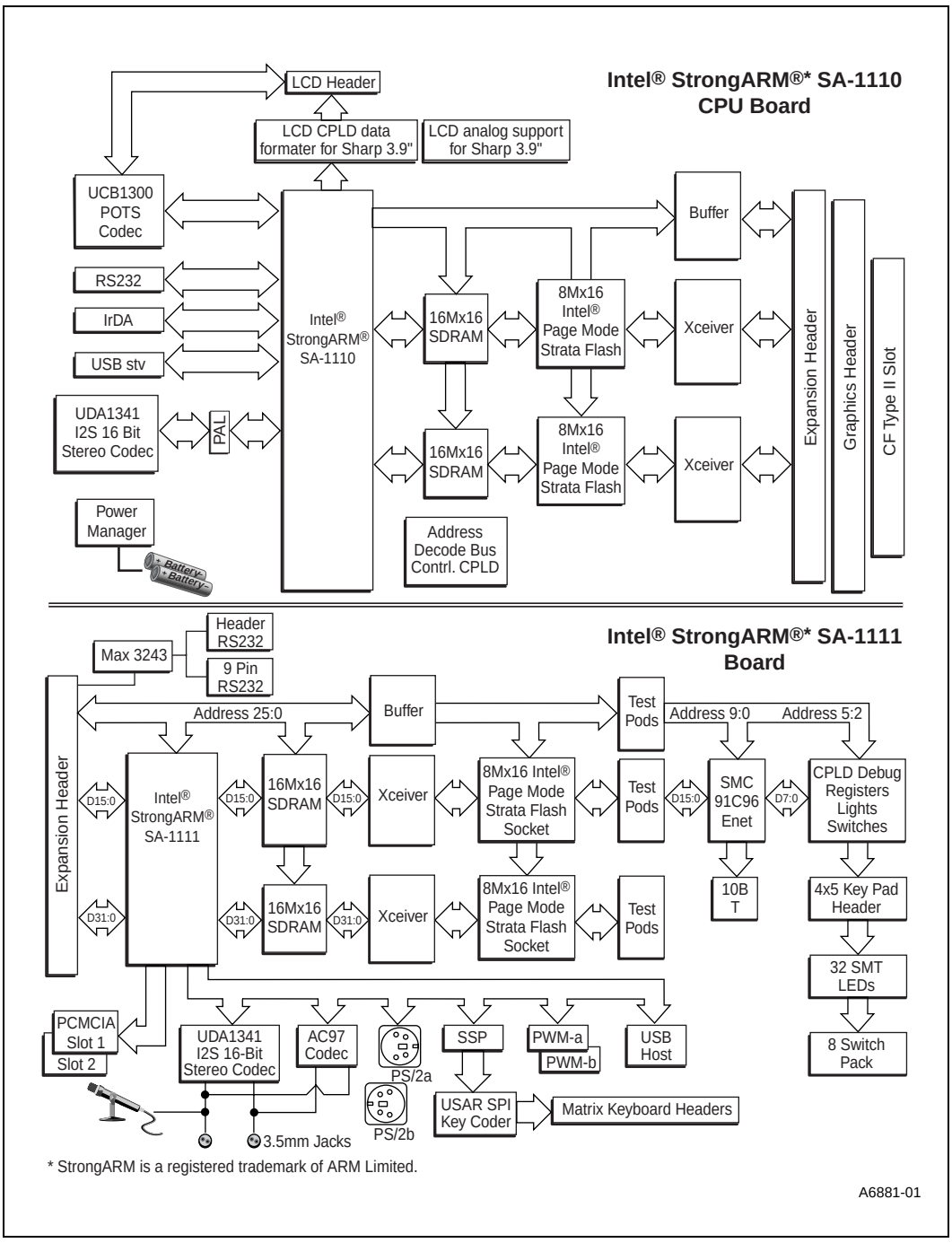
The SA-1110 Development Platform has been designed for the hardware and software development of hand held, palm top and tethered applications. The SA-1110 Development Platform provides all of the system components necessary for a Windows CE* sub-notebook system development platform with three independent video heads and high quality stereo sound.

The SA-1111 Development Module supplements the SA-1110 Development Platform by providing a vehicle to evaluate the SA-1111 Companion Chip, expanding the SA-1110 development environment, and providing general system debug features.

The SA-1111 improves the hardware and software developer's environment by supporting the following components:

- Ethernet 10BaseT port (RJ45 connector)
- Four logic analyzer connectors (Hewlett-Packard and Tektronix compatible 38-pin Mictors)
- Two debug serial ports with full modem capability
- LED display
- Keypad and matrix keypad I/O
  - — PS/2 ports with power control
  - — Expansion SDRAM memory (64 MB, 50 MHz)
  - — Socketed, buffered expansion Flash memory (32 MB Intel Strataflash™
- One each Type II/III PCMCIA socket and Compact Flash socket with power control
- Philips UDA1341 and AC97 codecs sharing the microphone, input, and output jacks
- Access to two PWM signals from the SA-1111 companion chip
- USB host connector with power control

Figure 3-1. SA-1110 Development Platform Block Diagram



The block diagram shown in Figure 3-1 illustrates the modularity, flexibility and extend ability of the design. The SA-1110 Development Board supports two 140 pin fully buffered daughter board expansion headers. One of these expansion headers is for the SA-1111 Development Module, the other is for a graphics accelerator board.

The system partitioning is intended to allow the SA-1110 Development Board to be a minimal palm PC system. All device interfacing is implemented with in-system programmable CPLDs and most system interface points are available on connectors suitable for daughter boards or cables. Although not intended as a ready to manufacture product design, the SA-1110 Development Platform provides the basis for low-cost derivative designs.

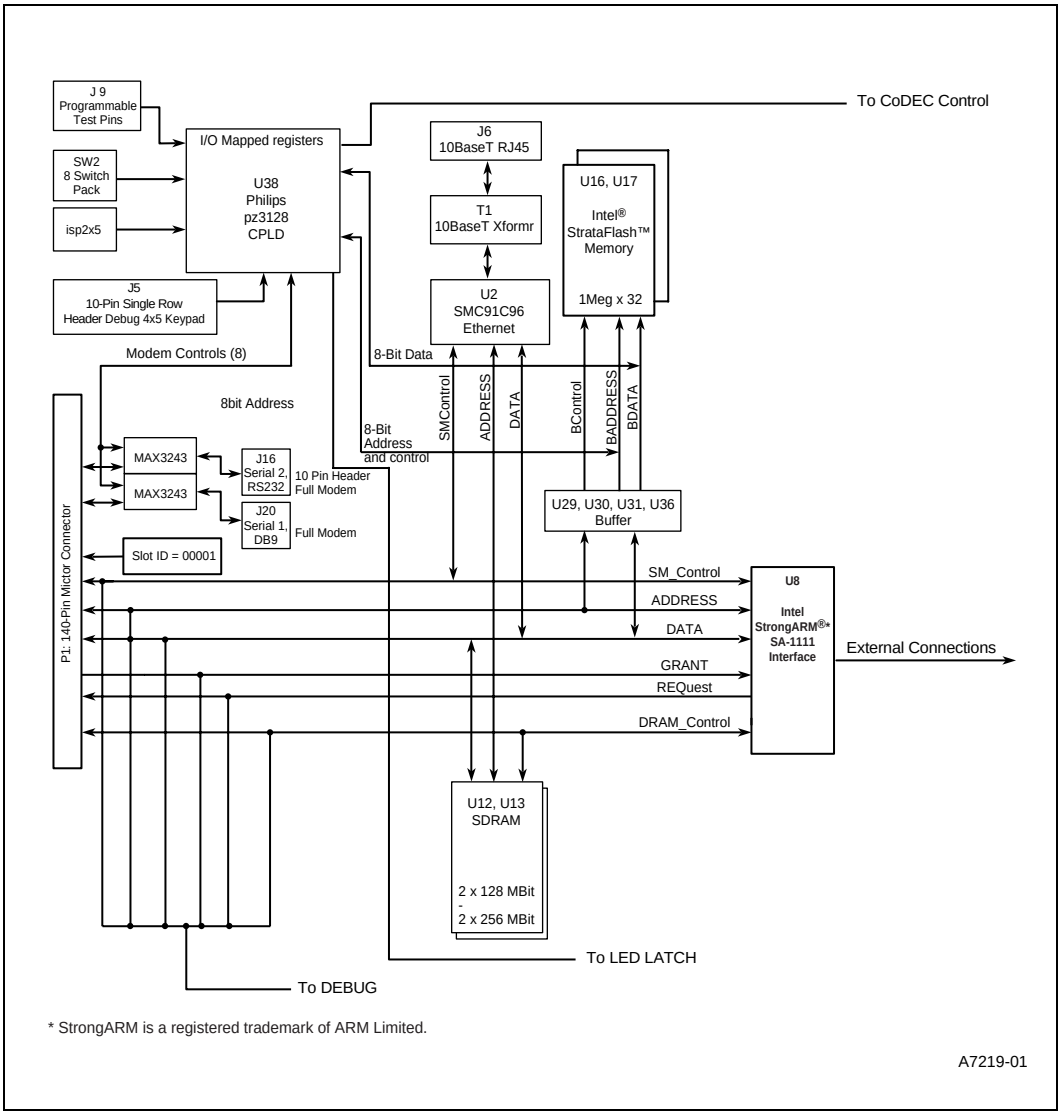
3.1 SA-1111 Block Diagram

The SA-1111 Development Module supports a variety of on board audio peripherals, expansion memory, serial I/O connectivity, support signals such as memory, Ethernet, modems, switches, CPLD control, PCMCIA and Compact Flash sockets, (see Figure 3-2 and Figure 3-3), and external stimuli such as sound, keyboard input, mouse, debugging connections and indicators (see Figure 3-4).

3.1.1 Expansion Connector Interface

Figure 3-2 shows the expansion connector interface from the SA-1110 Development Board to the SA-1111 Development Module. The SA-1110 control signals from serial channels 1 and 3 are also made available on the expansion connector to implement two RS-232 serial ports on the SA-1111 Development Module. These serial ports are application dependent, but are primarily intended for in passing messages in the debug environment. Serial port 2 at J16 can be used to support a PC-type keyboard, since the +12 V keyboard power can be optionally supplied to the connector. In addition to supporting system I/O mapped registers, an in system programmable CPLD is used to interpret an 8-position switch pack, providing modem control signals to the RS-232 debug ports, control switching between 1 of 2 on-board audio codecs, and interface to a 4x5 keypad used for debug purposes. The unused CPLD pins are brought out to a header for user access. Expansion SDRAM and socketed Flash memory can be accessed by either the SA-1110 or SA-1111 device. An Ethernet controller and 10BaseT interface are provided for communications in software development environments.

Figure 3-2. On-board Support Signals



3.1.2 Serial Connectivity

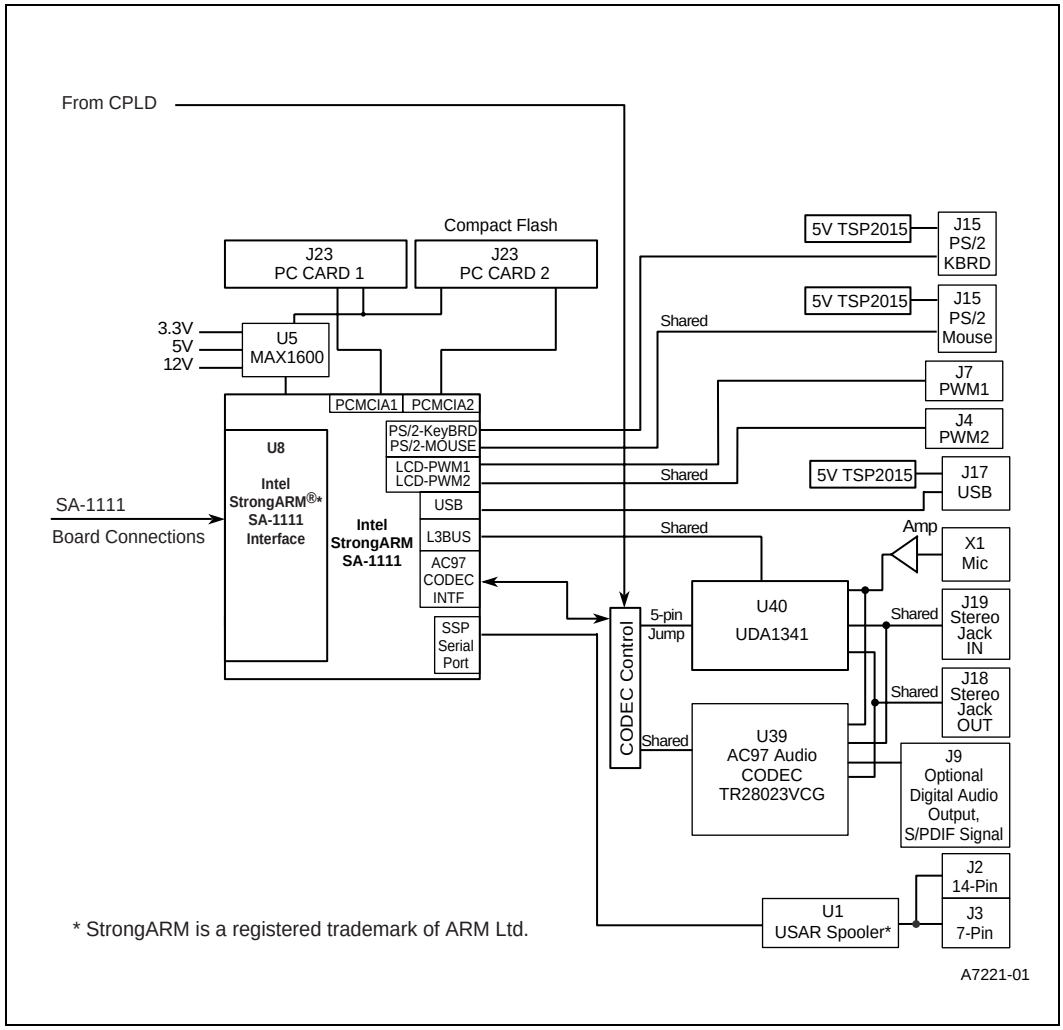
Figure 3-3 shows the SA-1111 Development Module's serial connectivity. A PS/2 serial port and a USB host port are supported by the SA-1111 device. The SA-1111 Development Module provides power control for each of these ports. Two PWM signals from the SA-1111 device are available for possible applications in LCD brightness and contrast control.

The SA-1111 Development Module contains two audio codec devices. The Philips UDA1341 is I^2S compatible and interfaces to the SA-1111 L3 bus. An AC 97 codec is also available that connects to the AC-Link interface and shares the SA-1111 Development Module's microphone and audio jacks, as well as the serial data paths between the SA-1111 and the codecs. Since only one

audio codec can be active at a time, some external logic controlled by the SA-1111 Development Module's CPLD determines device selection. Optionally, installing an AC97 device at U39 with digital audio output signal S/PDIF at pin 48, allows for that signal to be present at header J9.

The SSP serial port block on the SA-1111 device is connected to a USAR SPIcoder* device that interfaces to an 8x14 matrix keyboard. Also shown in Figure 3-3 are the SA-1111 PCMCIA and Compact Flash interfaces that incorporate a complete set of internal buffers for slot address, data, and control signals. Power control is provided on the SA-1111 Development Module for each socket of the Type II/III dual carriage. The Compact Flash dedicated inner socket requires a PCMCIA-to-Compact Flash adapter board.

Figure 3-3. Audio and External Inputs



3.1.3 Logic Analyzer Connections

Figure 3-4 shows some of the debug features available on the SA-1111 development module. There are four Hewlett-Packard and Tektronix -compatible 38-pin Mictor connectors populated with buffered versions of nearly every signal present on the SA-1110 development board expansion connector for logic analyzer connectivity. See Table 3-1 for a cross reference of these signals in Hewlett-Packard and Tektronix pin assignment schemes. Clocks are available at Mictor connector J21 and J22, supplied as a GPIO output from the SA-1110 Development Board. Header J14 provides two spare buffer inputs available to the user to select additional signals to make present at the logic analyzer through J24. Adapters may be used to connect it to some other brands of logic analyzer connections.

Also shown in Figure 3-4 are the LED latches, U6 and U10, for the bank of 32 SMT green LEDs. These LEDs can be used for development and debug status.

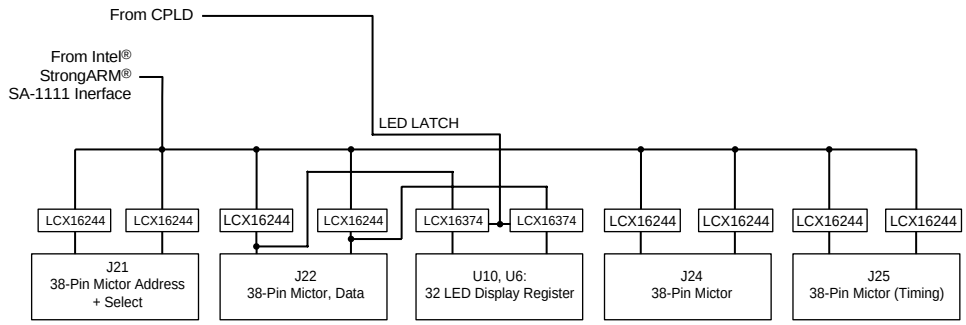
Table 3-1. Logic Analyzer Pin Assignments

Mictor Pin	Tektronix	HP	J24	J25	J21	J22
1	—	—	NC	NC	NC	NC
2	—	—	NC	NC	NC	NC
3	CLK	CLK	NC	NC	BSDCLK1	PROBECLK0
4	3:7	D15	P_SDA	P_NWE	BnSDCS_2	BXD31
5	3:6	D14	P_SCL	P_NOE	BXNRAS0	BXD30
6	3:5	D13	P_GPIO0	P_RDY	BXNCAS3	BXD29
7	3:4	D12	P_GPIO1	P_NRESET_OUT	BXNCAS2	BXD28
8	3:3	D11	P_PROBECLK0	P_RD_NWR	BXNCAS1	BXD27
9	3:2	D10	P_TXD2	P_SDCKE1	BXNCAS0	BXD26
10	3:1	D9	P_RXD2	P_SDCKE0	BXA25	BXD25
11	3:0	D8	P_TXD	P_SDCLK2	BXA24	BXD24
12	2:7	D7	P_RXD	P_SDCLK1	BXA23	BXD23
13	2:6	D6	P_BATT_FAULT	P_SDCLK0	BXA22	BXD22
14	2:5	D5	P_GFX_IRQ	P_NCS5	BXA21	BXD21
15	2:4	D4	P_SA1111_PWM1	P_NCS4	BXA20	BXD20
16	2:3	D3	P_SA1111_PWM0	P_NCS3	BXA19	BXD19
17	2:2	D2	P_MSDATA	P_NCS2	BXA18	BXD18
18	2:1	D1	P_MSCLK	P_NCS1	BXA17	BXD17
19	2:0	D0	P_SDATA_OUT	P_NCS0	BXA16	BXD16
20	0:0	D0	P_SDATA_IN	P_NPREG	BXA0	BXD0
21	0:1	D1	P_USB_PLUS	P_FLASHWAIT	BXA1	BXD1
22	0:2	D2	P_USB_MINUS	P_NIOIS16	BXA2	BXD2
23	0:3	D3	P_RXD_SSP	P_NPWAIT	BXA3	BXD3
24	0:4	D4	P_TXD_SSP	P_SA1111_IRQ	BXA4	BXD4
25	0:5	D5	P_SFRM	P_NPCE1	BXA5	BXD5
26	0:6	D6	P_SCLK	P_NPCE2	BXA6	BXD6

Table 3-1. Logic Analyzer Pin Assignments

Mictor Pin	Tektronix	HP	J24	J25	J21	J22
27	0:7	D7	P_USER_PROBE1	P_NPIOR	BXA7	BXD7
28	1:0	D8	P_USER_PROBE0	P_NPIOW	BXA8	BXD8
29	1:1	D9	P_SMROM_EN	P_NPWE	BXA9	BXD9
30	1:2	D10	P_XCVR2OE	P_NPOE	BXA10	BXD10
31	1:3	D11	P_NRESET_IN	P_BGPIO_27	BXA11	BXD11
32	1:4	D12	P_INGFXRESET	P_MBGNT	BXA12	BXD12
33	1:5	D13	P_AUDIO_CLK_OUT	P_MBREQ	BXA13	BXD13
34	1:6	D14	P_BIT_CLK	P_XNSDRAS	BXA14	BXD14
35	1:7	D15	P_FRAME_SYNC	P_XNSDCAS	BXA15	BXD15
36	CLK	CLK	NC	NC	BXNSDRAS	BXNSDCAS
37	—	—	NC	NC	NC	NC
38	—	—	NC	NC	NC	NC
39	GND	GND	GND	GND	GND	GND
40	GND	GND	GND	GND	GND	GND
41	GND	GND	GND	GND	GND	GND
42	GND	GND	GND	GND	GND	GND
43	GND	GND	GND	GND	GND	GND

Figure 3-4. Debug and LED Connections



* StrongARM is a registered trademark of ARM Limited.

A7220-01

Hardware Considerations 4

4.1 General-Purpose Expansion Bus Headers

The general-purpose expansion bus headers provided on the SA-1110 Development Board are referred to as the Xbus headers (not to be confused with ISA Xbus). All SA-1110 address, data and memory interface signals are buffered before driving the Xbus headers.

The Xbus headers allow interfacing to the SA-1111 Development Module as well as a third-party graphics chips.

For information on the system memory map, see the *Intel® StrongARM® SA-1110 Development Board User's Guide.*

4.2 Expansion Flash Memory

Two socketed Intel page mode 32Mbyte StrataFlash™ devices provide expansion flash and a selectable boot flash. The SA-1111 Development Module's flash bank is populated with 32Mbyte socketed Intel fast page mode 3V StrataFlash devices. These components provide 32 Mbyte flash banks. The design supports 56-lead TSOP StrataFlash FlashFile™ devices with densities of 32 Mbit, 64 Mbit, and 128 Mbit.

The SA-1110 microprocessor always boots from Bank 0. Bank 0, which can be on the SA-1110 Development Board or the SA-1111 Development Module, is controlled by component SW2, switch 8 when the SA-1111 Development Module is assembled to the SA-1110 motherboard. When Bank 0 is on the SA-1110 Development Board, Bank 1 is on the SA-1111 Development Module. When Bank 0 is on the SA-1111 Development Module, Bank 1 is on the SA-1110 Development Board. When the SA-1110 Development Board does not have a daughterboard attached, Bank 0 is on the SA-1110 Development Board.

The control of the location of Bank 0 is described in Table 4-1.

Table 4-1. Bank 0 Location

When Component SW2 Switch 8 is…	Then Bank 0 is located on the:
ON position	SA-1110 Development Board
OFF position	SA-1111 Development Module

The SA-1111 Development Module circuitry that controls the location of Bank 0 is shown in the SA-1111 schematics on sheet 7 of 11, signal name SWPK7. When this signal enters the SA-1110 Development Board, it becomes SWAP_FLASH.

4.3 General-purpose Status Input Switch and Jumper Descriptions

Component SW2 is used for determining where bank 0 is located, and jumps to one of four boot images. Unused switches are user defined.

The switch setting may be read from the Switch Status Register in the CPLD SWPK register.

Table 4-2. Switch and Jumper Descriptions

Switch	Condition	Function	Default
SW2-1	LSB of 2 bit address	GPIO 0 –part of a 2 bit address to jump to one of four boot images.	Off
SW2-2	MSB of 2 bit address	GPIO 1–part of a 2 bit address to jump to one of four boot images.	Off
SW2-3	—	User defined	Off
SW2-4	—	User defined	Off
SW2-5	—	User defined	Off
SW2-6	—	User defined	Off
SW2-7	—	User defined	Off
SW2-8	—	Off — Bank 0 is located on the SA-1111 Development Module On — Bank 0 is located on the SA-1110 Development Board	On
SW1-1	—	Ethernet Controller – ECEEP	Off
SW1-2	—	Ethernet Controller – IOS2	Off
SW1-3	—	Ethernet Controller – IOS1	Off
SW1-4	—	Ethernet Controller – IOS0	Off
J26	Clock selection	SA-1111 SDRAM Bank Select	Connect pins 2 and 3

4.4 CPLD Register Map

The SA-1111 Development Module implements several memory mapped IO registers. The SA-1110 CS2 address space, which starts at 0x1000,0000, is dedicated to system level registers. The SA-1111 Development Module registers are implemented in a CPLD with an 8-bit data path to the 32-bit system bus. The 8-bit CPLD is addressed on 32-bit word boundaries. The CPLD needs lower address bits A[7:2] to decode the registers within the CS2 address space.

The SA-1111 device should be used for controlling the interface to different features. For example, use the SA-1111 to control the audio controls, do **not** use the CPLD to control the audio controls.

The General-Purpose I/O (GPIO) interface is a functional block that provides up to 19 bits of programmable input/output. As inputs, each pin can be configured as an interrupt source, wake-up source, or both. The GPIO ports provide general-purpose digital I/O capabilities for the SA-1111 (see sheet 9 of 11).

In addition, several register based GPIO pins are assigned to sense and control the CF socket. The SA-1110 GPIO22 and GPIO21 pins serve dual functions in the SA-1110 CF design. When the SA-1111 Development Module is attached, GPIO22 and GPIO21 are configured to be the GPIO alternate function bus DMA control signals nMBREQ and nMBGNT. When the SA-1111 Development Module is not attached and the SA-1110 CF slot is active, GPIO22 and GPIO21 are used as CF card detect interrupt and CF RDY interrupt signals.

For more information about GPIOs, see the *Intel® StrongARM® SA-1111 CPU Companion Chip Developer's Manual.*

Table 4-3 describes the SA-1111 CPLD register map, which controls the CPU transceivers and some of the CF support signals.

For more information about CPLD functions, see Appendix 4.11.

The CPLD source code for the SA-1111 Development Module is provided in Appendix A.

Table 4-3. Register Descriptions

Address	Name	Description
1000,00C0	AUD_CTL	Audio controls TBD CPLD [TBD] (RW)
1000,00B0	MDM_CTL	Modem control signals CPLD [TBD] (RW)
1000,00A0	CR	Control Register CPLD [7:0] (RW)
1000,0090	KP_X_OUT	Keypad row write CPLD [4:0] (RW)
1000,0080	KP_Y_IN	Keypad column read CPLD [3:0] (RO)
1000,0020	SWPK	Switch pack, 8 switches CPLD [7:0] (RO)
1000,0000	WHOAMI	System ID register CPLD (RO) reads constant 0x11 0x11 = SA-1111 ID + SA-1110 ID

4.5 Interrupt Reason Register

The Interrupt Reason Register is a read-only register that reads the Ethernet device, the USAR device, and the SA-1111 device.

Table 4-4 provides bit descriptions of the Interrupt Reason Register:

Table 4-4. Interrupt Reason Register 10000024

	System Configuration																						SA-1110 Development Board									
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved																													SA1111_IRQ	nATN	nEnetIRQ
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bits	Name	Description
:0	nEnetIRQ	Ethernet IRQ status 0 – Ethernet device IRQ true 1 – Ethernet device IRQ false
1	nATN	USAR IRQ status 0 – USAR device IRQ true 1 – USAR device IRQ false
2	SA1111_IRQ	SA-1111 Device IRQ 0 – SA-1111 device IRQ true 1 – SA-1111 device IRQ false
31:3	Reserved	—

4.6 SDRAM

Two 128 Mbit SDRAM chips provide 32 Mbyte of expansion SDRAM and can fit in this generic 54-pin TSOP footprint. Up to 64 Mbytes of expansion SDRAM can be supported when these locations are populated with 256 Mbit devices. Using SDRAM on the SA-1111 Development Module demonstrates the buffering scheme and the flexibility of the SA-1110 memory timing by supporting two banks of SDRAM with different timings. The SA-1111 SDRAM could be configured to be the target of SA-1111 DMA.

Logic jumpers (zero ohm resistors) on the SA-1111 Development Module allow this SDRAM (bank 2) to be the target of SA-1111 DMA instead of the SA-1110 Development Board SDRAM (bank 0).

Note: The standard jumper configuration will not allow the SA-1111 companion chip to DMA to the SA-1111 Development Module's SDRAM.

Table 4-5. Bank Selection Jumpers

R52	R51	Condition
Out	Out	Illegal
Out	In	Allows SA-1111 DMA access to Bank 0 of SA-1110 Development Board (default configuration)
In	Out	Allows SA-1111 DMA access to Bank 2 of SA-1111 Development Module
In	In	Illegal

4.6.1 Expansion SDRAM Interface

The SA-1111 Development Module provides support for 64 Mbit, 128 Mbit or 256 Mbit SDRAMs. The 54-pin TSOP footprint supports a wide variety of SDRAM types and sizes. The SA-1111 Development Module SDRAMs are buffered from the SA-1110 CPU and must run on a SA-1110 CoreClk/4 permitting a maximum SDRAM clock speed of 51.5 MHz for this SDRAM bank.

Using two 256 Mb 16 Mx16 SDRAMs provides a expansion memory load of 64 MBytes. Due to cost and availability issues with 256 Mbit SDRAMs, the SA-1111 Development Module will initially be configured with 128 Mb SDRAMs for a 32 MByte memory load.

4.7 PCMCIA and Compact Flash

The SA-1111 chip supports one PCMCIA and one Compact Flash (CF) socket. The SA-1111 Development Module provides a dual socket type II/III PCMCIA carriage. Socket A provides a complete PCMCIA interface. Socket B (inner slot) provides a Compact Flash subset of the PCMCIA interface. Use of the dedicated CF socket B requires a PCMCIA to CF adapter board.

Caution: The SA-1110 Development Board supports a single type II CF socket. The SA-1110 Development Board's CF slot uses the same interface signals required by the SA-1111 PCMCIA interface logic, therefore the SA-1110 Development Board CF socket only functions when there are no daughter boards present.
A CF card plugged into the SA-1110 Development Board's CF socket when the SA-1111 Development Module or the graphics boards are present will crash the system and may damage the transceivers.
The buffer and transceiver components on the SA-1110 Development Board that normally drive the CF socket are reconfigured to become the system buffers and transceivers to isolate the SA-1111 Development Module and the graphics board from the SA-1110 Development Board when the daughter boards are attached.

4.8 USB Host

The SA-1111 Development Module provides a standard USB host connector, J17, which is also routed to the CPLD. In addition, a USB power control (Texas Instruments TPS2015) is provided to control the 5V USB power feed to enable this port. The fault sense signals from the power control

chip are attached to the spare SA-1111 GPIO pins to allow application software to report USB power faults. In order to protect the SA-1111 USB pins from electrostatic discharge, damage protection diodes (Texas Instruments SN75240) are also provided.

For more information about the USB Host functions, see the *Intel® StrongARM® SA-1111 CPU Companion Chip Developer's Manual*.

4.9 PS/2 Port

The SA-1111 Development Module provides a standard PS/2 connector, J15. In addition a power control chip (Texas Instruments TPS2015) is provided to control the 5V PS/2 power feed pins to enable these ports. The fault sense signals from the power control chip are attached to spare SA-1111 GPIO pins to allow application software to report PS/2 power faults. In order to protect the PS/2 pins from electrostatic discharge, damage protection diodes (Texas Instruments SN75240) are provided.

For more information about the PS/2 port functions, see the *Intel® StrongARM® SA-1111 CPU Companion Chip Developer's Manual*.

4.10 RS-232 Ports

The SA-1111 Development Module provides one 9-pin RS-232 port connector, J20 (UART3). This port is used for system debug and firmware development.

A second port on the SA-1111 Development Module is provided via a standard 5x2 0.1" header, J16 (UART1). The J16 port may be used for displaying debug messages from the development environment while entering inputs on J20. Both connectors, which are in the schematics on Sheet 9 of 11 and shown in Figure 2-1, are routed to the CPLD and have full modem support with control signals.

4.11 Debug CPLD

The SA-1111 Development Module includes a Philips pz3128 low-power CPLD with in-system programmability. This 100 pin TQFP is accessible as a memory mapped IO device via an eight-bit wide program interface to the SA-1110. All registers are implemented in the CPLD that manage the 4 x 4 keypad, LEDs, switch pack and audio codec interface switching, modem control, and power control for the PS/2 ports.

A 10-pin JTAG header at J8 is provided that supports the pz3128 in system programming feature. The large CPLD allows design flexibility as well as a silicon scratch pad for hardware modifications or future function additions. Spare CPLD pins, which can be user-defined, are available on J9.

The following registers are implemented in the CPLD:

Table 4-6. Control Register0 NCR_0 0x1000,00A0

	System Configuration																					SA-1110 Development Board										
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved																									A1VPP	A0VPP	SPI_KB_WKUP	ENET_OSC_EN	MS_PWR_EN	P_PWR_EN	GP01_OFF
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bits	Name	Description
0	GP01_OFF	SA-1110 GPIO 0 and GPIO 1 drivers tri-state control (RW) Reset to 0 0 – SW0 SW1 drive GPIO0 GPIO1 1 – GPIO0 GPIO1 drivers tri-state
1	TP_PWR_EN	TRACKPAD power ON 0 – TrackPad power on 1 – TrackPad power off
2	MS_PWR_EN	PS2 Mouse power ON 0 – Mouse power on 1 – Mouse power off
3	ENET_OSC_EN	Ethernet Oscillator ON 0 – Ethernet OSC off 1 – Ethernet OSC on
4	SPI_KB_WKUP	SPI Keyboard Wakeup control 0 – WKUP signal tristated 1- WKUP signal forced low
5	A0VPP	PCCARD VPP A0 power control (RW) Reset to 0
6	A1VPP	PCCARD VPP A1 power control (RW) Reset to 0

4.11.1 Modem Control

This register, which is CPLD controlled, determines the modem functions:

Table 4-7. Modem Control Register MDM_CTL_0 0x1000,00B0

	System Configuration																					SA-1110 Development Board									
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved																												DTR2	RTS2	DTR1	RTS1
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bits	Name	Description
0	RTS1	Modem1 TX Control (R/W) Reset 0
1	DTR1	Modem1 TX Control (R/W) Reset 0
2	RTS2	Modem2 TX Control (R/W) Reset 0
3	DTR2	Ethernet Oscillator ON 0 – Ethernet OSC off 1 – Ethernet OSC on

Table 4-8. Modem Control Register MDM_CTL_0 0x1000,00B0

	System Configuration																					SA-1110 Development Board									
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved																										DCD2	DSR2	CTS2	DCD1	DSR1	CTS1
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bits	Name	Description
0	CTS1	Modem1 RX Control (RO)
1	DSR1	Modem1 RX Control (RO)
2	DCD1	Modem1 RX Control (RO)
3	CTS2	Modem2 RX Control (RO)
4	DSR2	Modem2 RX Control (RO)
5	DCD2	Modem2 RX Control (RO)

4.11.2 4x4 Debug Keypad

The primary use of the 4 x 4 keypad is as a simple input device for system debugging and diagnostic interaction, which is connected to J5. The debug CPLD provides the logic interface. A ten-pin in-line 0.1” header is provided to connect to the keypad ribbon cable. Pin 10 must be ground while pins 1 to 9 are connected to a pz3128 CPLD. The 4 x 4keypad interface provides four write bits for the X drive and five read bits for the Y read.

The 4 x 4 matrix keypad is also suitable for use with telephone applications is supported in the SA-1111 Development Module. The primary use of the keypad is as a simple input device for system debugging and diagnostic interaction. Applications may also use this resource.

Table 4-9. Debug Keypad Description

	Description
1	Column 0
2	Column 1
3	Column 2
4	Column 3
5	Row 0
6	Row 1
7	Row 2
8	Row 3
9	GND

4.11.3 LEDs

The SA-1111 Development Module provides a 32 low-power SMT LEDs connected to a 32-bit I/O mapped register, which is CPLD controlled. These LEDs are mapped as follows:

Table 4-10. LED Mapping

Address	Name	Description
1000,001C	LEDS_B3	LEDs register byte 3 (WO) LEDs [31:24]
1000,0018	LEDS_B2	LEDs register byte 2 (WO) LEDs [23:16]
1000,0014	LEDS_B1	LEDs register byte 1 (WO) LEDs [15:8]
1000,0010	LEDS_B0	LEDs register byte 0 (WO) LEDs [7:0]

4.11.4 Audio

The SA-1111 Development Module supports the AC 97 and the I^2S interfaces, which is shown on sheet 4 of 11 in the SA-1111 evaluation board schematics. The SA-1111 Development Module can support either interface, but **not both** at the same time. Either interface can receive inputs from the on-board microphone and input/output jacks.

The AC 97 codec, which can be a variety of devices, is shown as a TR28023VCG. When the AC97 codec is a Cirrus Logic CS4299 component, a speaker output is also available. This signal is part of a pair of speaker connections on pin 16 (S_PDIF_OUT) and pin 15 (ground) on J9 (see schematic sheet 7 of 11). The I^2S interface is available from the UDA1341TS.

A pair of 3.5mm stereo line in and line out jacks (J18, J19) are provided on the SA-1111 Development Module as I/O to the AC97 codec or the UDA1341 through analog multiplexers under the audio control register bit 0, as shown in Table 4-11.

For more information about the audio port, see the Intel® StrongARM® SA-1111 Companion Chip Developer's Manual.

Table 4-11. Audio Control Register AUD_CTL 0x1000,00C0

	System Configuration																				SA-1110 Development Board									
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved																														MUTE_1341	SEL_1341
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bits	Name	Description
0	SEL_1341	Select I2S Audio (RW) Reset value = 0 0 – Select AC 97 Codec 1 – Select UDA 1341 codec
1	MUTE_1341	Mute 1341 Codec Reset value = 0 0 – Normal mode 1 – Mute mode

4.12 SPI Port

The SA-1111 Development Module supports an SPI port when enabled by the SA-1111 device. The SA-1110 development system interfaces to a USAR SPI coder keyboard chip. Two JST connectors (J2 and J3) directly interface to an 8 x 14 matrix keyboard. For more information, see information describing the SSP function in the *Intel®StrongARM® SA-1111 Companion Chip Developer's Manual.*

4.13 Ethernet Interface

This Ethernet interface, which is J6, is an RJ45 connector, and is used for debugging purposes. Table 4-12 describes the Ethernet LED indicators:

Table 4-12. Ethernet LED Indicators

LED	Indicator
D1 (yellow)	Base selection
D2 (red)	Transmitting
D3 (yellow)	Receiving
D4 (green)	Link

Table 4-13 describes the Etherenet interface signals:

Table 4-13. Ethernet Interface Signals

Signal	Description
ENET_nREG	Register select as in nREG in PCMIA
NCS3	Chip select
ENET_nWAIT	Wait signal
NENET_WE	Write enable signal
NENET_OE	Output enable signal
NENET_IRQ	Interrupt signal
NENET_IRQ	Interrupt signal
NIORD	IO Read signal
NIOWR	IO Write signal
LAN_RESET	Chip reset active high
ENET_OSC_EN	Clock enable

The SMSC9196 Ethernet controller is used on the SA-1111 Development Module for LAN interface. The SMSC9196 interfaces directly to the SA-1110, which maps into the CS3 address space. The interface is a PCMCIA type with control signals generated by the CPLD. For more information on CPLD code, see Appendix A.

Access to the configuration register of the SMSC9196 is similar to the access method of attribute memory space for the PCMCIA protocol.

The oscillator can be disabled to save power when the SMSC91916 is not used. This can be done through bit 3 of Control Register 0. For more information, seeTable 4-6.

The Serial EEPROM is provided to store LAN identification and configuration information. The EEPROM is driven directly by the SMSC9196 controller. Data can be read or written to the EEPROM by issuing a command to the SMSC9196.

4.14 Power Measurement

Test hooks are provided for the accurate measurement of the SA-1111 current Idd. Current is calculated from the voltage drop across a low-Ohm resistor (0.02 Ohms). The shunt resistors also function for SA-1111 voltage margining.

The following current test points are provided and are shown in the schematics on sheet 3 of 11:

Table 4-14. Test Hooks

Jumper	Measurement Point
J10	Battery Current Sense node
J11	5V Current Sense node
J12	12V Current Sense node
J13	3.3V Current Sense node

4.15 Analog Outputs

The SA-1111 Development Module's companion chip provides two Pulse Width Modulated (PWM) general-purpose outputs, J7 and J4 (see schematic sheet 9 of 11). These signals are available as test points on the SA-1111 Development Module and are wired to the 140-pin connectors. The PWM signals are intended for use in controlling the brightness and contrast of a display system and are not used in the SA-1110 Development Board but may be used by the graphics boards.

The AC 97 Codec provides for direct drive of small low power speaker. This signal is available as part of a pair of speaker connections on pin 16 (S_PDIF_OUT) and pin 15 (ground) on J9 (see schematic sheet 7 of 11).

4.15.1 Board Identification Bits

The Xbus header provides four board identification pins that allow the various daughter boards to identify themselves. The SA-1111 Development Module uses a soft ground (using a low-Ohm value resistor) on the DC_ID_0 pin and provides 50 K Ohm pull-ups for DC_ID_[3:1] (see schematic sheet 3 of 11).

The WHOAMI register also identifies the system (see Table 4-3 for a description of the WHOAMI register).

CPLD Code A

This appendix lists the CPLD code used on the Intel® StrongARM® SA-1111 Development Module.

Note: These code listing are for reference only. See Intel's web site for developers to obtain the latest source code.

A.1 SA11MOD1.PHD File Contents

Here are the contents of the sa11mod1.phd file:

```
Module sa1111cpld
// Filename: sa11mod1.phd
// This file is coded in Phillips HDL format

Title'SA1111 Board control CPLD'

declarations

"INPUTS

SDCLK0  pin 114;
BATT_FAULTpin40;
ADDR5..ADDR0pin 45,47,44,43,41,42;
nCS1    pin 48;
nCS2    pin 49;
nCS3    pin 50;

XA25    pin 115;
nENETIRQpin116;
nOE     pin 117;
nWE     pin 53;
KY_Y_IN4..KY_Y_IN0 pin59,58,56,55,54;
nATN    pin 60;
SWPK7..SWPK0pin72,71,67,65,64,63,62,61;
nRESET  pin 73;
CTS1    pin 74;
DSR1    pin 75;
DCD1    pin 76;
CTS2    pin 77;
DSR2    pin 79;
DCD2    pin 80;
SA1111_INT_OUTpin81;

"INPUT/OUTPUT
DATA7..DATA0pin3,2,121,122,110,1,120,119; "BIDIR"
```

```
WKUP    pin 100; "BIDIR"
GPIO_0  pin 22; "BIDIR"
GPIO_1  pin 109; "BIDIR"

"OUTPUTS
nIORD   pin 12  istype  'com';
nIOWR   pin 15  istype  'com';
nENETWEpin  13  istype  'com';
nENETOEpin  14  istype  'com';
POWEROK pin 30  istype  'com';
KY_X_DR3..KY_X_DR0pin10,18,28,36istype'com';
ENET_OSC_ENpin38istype 'com';
TP_PWR_ENpin128istype  'com';
MS_PWR_ENpin19 istype  'com';
SA1111_BOARD_IRQpin39istype'com';
RTS1    pin 20  istype  'com';
DTR1    pin 23  istype  'com';
RTS2    pin 29  istype  'com';
DTR2    pin 9   istype  'com';
A0VPP   pin 24  istype  'com';
A1VPP   pin 31  istype  'com';
FLASH_PLD_ENpin11  istype  'com';
FL_PLD_WEpin17 istype  'com';

SELECT_1341pin26istype 'com';
MUTE_1341pin32 istype  'com';

LED_B0  pin 127 istype  'com';
LED_B1  pin 124 istype  'com';
LED_B2  pin 126 istype  'com';
LED_B3  pin 125 istype  'com';

// Internal nodes

NCR0_reg6..NCR0_reg0nodeistype'reg_d';
ADCTL_reg1..ADCTL_reg0nodeistype'reg_d';
KY_X_DR_reg3..KY_X_DR_reg0nodeistype'reg_d';
MDCTL_reg3..MDCTL_reg0nodeistype'reg_d';
KY_X_DR_reg_ld nodeistype 'com';
NCR0_reg_ldnodeistype 'com';
ADCTL_reg_ld   nodeistype 'com';
MDCTL_reg_ld   nodeistype 'com';

"BUS defination
ADDR_5_0 = [ADDR5,ADDR4,ADDR3,ADDR2,ADDR1,ADDR0];
KYIN_7_0 = [0,0,0,KY_Y_IN4,KY_Y_IN3,KY_Y_IN2,KY_Y_IN1,KY_Y_IN0];
SWPK_7_0 = [SWPK7,SWPK6,SWPK5,SWPK4,SWPK3,SWPK2,SWPK1,SWPK0];
DATA_7_0 = [DATA7,DATA6,DATA5,DATA4,DATA3,DATA2,DATA1,DATA0];
KY_X_DR_3_0 = [KY_X_DR3,KY_X_DR2,KY_X_DR1,KY_X_DR0];

NCR0_reg6_0 =
[NCR0_reg6,NCR0_reg5,NCR0_reg4,NCR0_reg3,NCR0_reg2,NCR0_reg1,NCR0_reg0];
ADCTL_reg1_0 = [ADCTL_reg1,ADCTL_reg0];
KY_X_DR_reg3_0 = [KY_X_DR_reg3,KY_X_DR_reg2,KY_X_DR_reg1,KY_X_DR_reg0];
MDCTL_reg3_0 = [MDCTL_reg3,MDCTL_reg2,MDCTL_reg1,MDCTL_reg0];
```

```
// All registers are clocked with SDCLK0.

equations

nIORD = nCS3 # XA25 # nOE;
nIOWR = nCS3 # XA25 # nWE;

nENETWE = nCS3 # (!XA25) # nOE;
nENETOE = nCS3 # (!XA25) # nWE;

FLASH_PLD_EN = nCS1 & nCS2;
FL_PLD_WE = nWE;

POWEROK = !BATT_FAULT;

//Convert active low interrupts to all active high
SA1111_BOARD_IRQ = SA1111_INT_OUT # (!nATN) # (!nENETIRQ);

/* A low active signal is generated and the rising edge will latch the
    LED data from CPU */
LED_B0 = !(!ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & !nWE);
LED_B1 = !(!ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & ADDR0 & !nWE);
LED_B2 = !(!ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & !ADDR0 & !nWE);
LED_B3 = !(!ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & ADDR1 & ADDR0 & !nWE);

// Register decode and read data

//Drive DATA bus only when reading from nCS2 (reserved for ASSABET Control
registers)
DATA_7_0.oe = !nCS2 & !nOE;

/*DATA muxing from registers to DATA bus
  Some registers are readonly registers */
when (ADDR_5_0 == ^b000000)then
    { DATA_7_0 = ^b00010001; } else "System ID register read h00"

when (ADDR_5_0 == ^b001000)then
    { DATA_7_0 = SWPK_7_0; } else "Switch Pack read 10000020h"

when (ADDR_5_0 == ^b100000)then
    { DATA_7_0 = KYIN_7_0; } else "Keypad colomn read 10000080h"

when (ADDR_5_0 == ^b100100)then "Keypad row read 10000090h"
    { DATA_7_0 =
[0,0,0,0,KY_X_DR_reg3.q,KY_X_DR_reg2.q,KY_X_DR_reg1.q,KY_X_DR_reg0.q]; } else

when (ADDR_5_0 == ^b101000)then
    { DATA_7_0 = [0,NCR0_reg6_0.q]; } else "Control register 0 read 100000A0h"

when (ADDR_5_0 == ^b101100)then  "Modem register 0 read/write 100000B0h"
    { DATA_7_0 = [0,0,0,0,MDCTL_reg3.q,MDCTL_reg2.q,MDCTL_reg1.q,MDCTL_reg0.q]; }
else

when (ADDR_5_0 == ^b101101)then  "Modem register 1 read only 100000B4h"
```

```
    { DATA_7_0 = [0,0,DCD2,DSR2,CTS2,DCD1,DSR1,CTS1]; } else

when (ADDR_5_0 == ^b110000)then  "Audio control read 100000C0"
    { DATA_7_0 = [0,0,0,0,0,0,ADCTL_reg1.q,ADCTL_reg0.q]; } else

    { DATA_7_0 = ^b00000000; }

// All registers are implemented with synchronous clock (SDCLK0).
// The register load controls the data being loaded every clock.

// Define register inputs, clocks
//Key board drive register
when (KY_X_DR_reg_ld) then
    {KY_X_DR_reg3_0.d = [DATA3..DATA0];}
else
    {KY_X_DR_reg3_0.d = KY_X_DR_reg3_0.q;}
KY_X_DR_reg_ld = ( ADDR5 & !ADDR4 & !ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & !nWE & !nCS2
); //ADDR 10000090h
KY_X_DR_reg3_0.clk = SDCLK0;
KY_X_DR_reg3_0.ar = !nRESET;

//SA1111 board control register 0
when (NCR0_reg_ld) then
    {NCR0_reg6_0.d = [DATA6..DATA0];}
else
    {NCR0_reg6_0.d = NCR0_reg6_0.q;}
NCR0_reg_ld = ( ADDR5 & !ADDR4 & ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & !nWE & !nCS2 );
//ADDR 100000A0h
NCR0_reg6_0.clk = SDCLK0;
NCR0_reg6_0.ar = !nRESET;

//Audio control registers
when (ADCTL_reg_ld) then
    {ADCTL_reg1_0.d = [DATA1,DATA0];}
else
    {ADCTL_reg1_0.d = ADCTL_reg1_0.q;};
ADCTL_reg_ld = ( ADDR5 & ADDR4 & !ADDR3 & !ADDR2 & !ADDR1 & !ADDR0 & !nWE & !nCS2
); //ADDR 100000C0h
ADCTL_reg1_0.clk = SDCLK0;
ADCTL_reg1_0.ar = !nRESET;

//Modem control register
when (MDCTL_reg_ld) then
    {MDCTL_reg3_0.d = [DATA3..DATA0];}
else
    {MDCTL_reg3_0.d = MDCTL_reg3_0.q;}
MDCTL_reg_ld = ( ADDR5 & !ADDR4 & ADDR3 & ADDR2 & !ADDR1 & !ADDR0 & !nWE & !nCS2 );
//ADDR 100000B0h
MDCTL_reg3_0.clk = SDCLK0;
MDCTL_reg3_0.ar = !nRESET;

//Logic assignments to control GPIO 0 and 1 pins
GPIO_0 = SWPK0;
GPIO_0.oe = !NCR0_reg0;  //NCR0 register bit 0
```

```
GPIO_1 = SWPK1;
GPIO_1.oe = !NCR0_reg0;  //NCR0 register bit 0

// WKUP is a tri PAD and it’s a active low signal
WKUP= ^b0;
WKUP.oe= NCR0_reg4.q;

// Keypad Control bits output
KY_X_DR_3_0 = KY_X_DR_reg3_0.q;

// ETERNET OSCILLATOR Enable signal
ENET_OSC_EN = NCR0_reg3.q;

// Mouse PAD power control (PS2 port)
MS_PWR_EN = NCR0_reg2.q;

// Track PAD power control (PS2 port)
TP_PWR_EN = NCR0_reg1.q;

//Modem control signals
RTS1 = MDCTL_reg0.q;
DTR1 = MDCTL_reg1.q;

RTS2 = MDCTL_reg2.q;
DTR2 = MDCTL_reg3.q;

//PCCARD VPP contro signals
A0VPP = NCR0_reg5.q;
A1VPP = NCR0_reg6.q;

//I2S or ACLINK control signal
SELECT_1341 = ADCTL_reg0.q;
MUTE_1341 = ADCTL_reg1.q;

end "End of module"
```

Support, Products, and Documentation

If you need general information or support, call **1-800-628-8686** or visit Intel's website at:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, a product catalog, or other Intel literature may be obtained by calling **1-800-548-4725** or by visiting Intel's website for developers at:

http://developer.intel.com