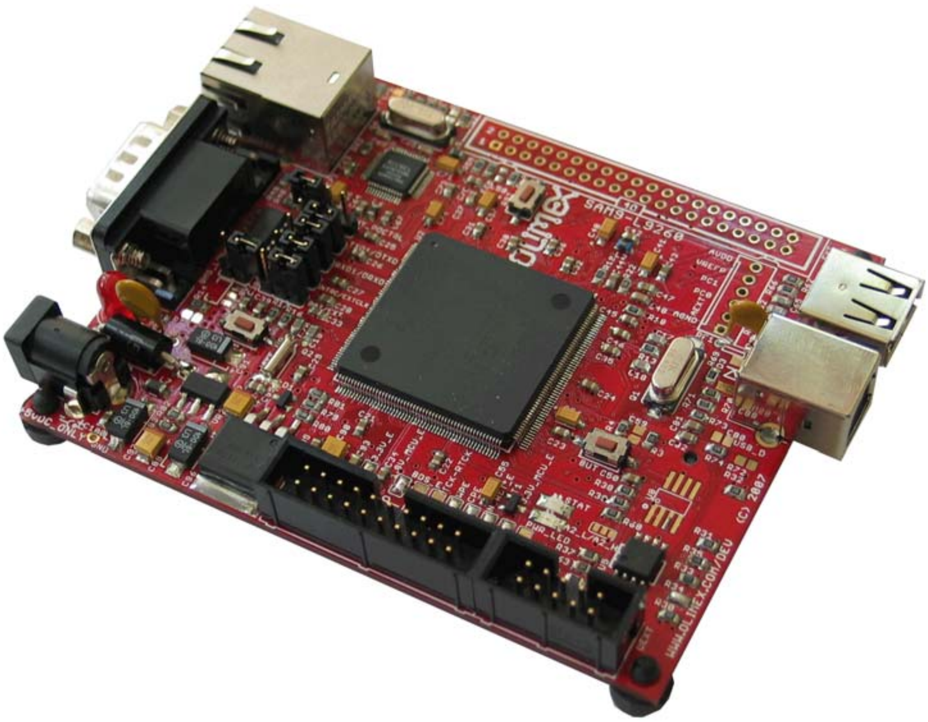


SAM9-L9260 development board

Users Manual

All boards produced by Olimex are ROHS compliant

Rev.A, April 2008

INTRODUCTION:

SAM9-L9260 is a low cost development platform with ARM9 microcontroller, 64MB SDRAM and 512MB NAND Flash. The board has Ethernet 100Mbit controller, USB host, USB device, RS232 and a 40 pin extension port with all unused SAM9260 ports available for add-on boards. SAM9-L9260 has vast amounts of Flash and RAM and runs Linux, WindowsCE and other RTOS natively. There is an on-board RTC clock with 3V Li battery backup.

BOARD FEATURES:

- MCU: AT91SAM9260 16/32 bit ARM9™ 180MHz operation
- standard JTAG connector with ARM 2x10 pin layout for programming/debugging with ARM-JTAG
- 64 MB SDRAM
- 512MB NAND Flash (seen in Linux as silicon drive)
- Ethernet 100Mbit connector
- USB host and USB device connectors
- RS232 interface and drivers
- SD/MMC card connector
- one user button and one reset button
- one power and two status LEDs
- on board voltage regulator 3.3V with up to 800mA current
- single power supply: 5V DC required
- power supply filtering capacitor
- 18.432 Mhz crystal
- extension port
- PCB: FR-4, 1.5 mm (0,062"), soldermask, silkscreen component print
- Dimensions: 100 x 80 mm (3.94 x 3.15")

ELECTROSTATIC WARNING:

The SAM9-L9260 board is shipped in protective anti-static packaging. The board must not be subject to high electrostatic potentials. General practices for working with static sensitive devices should be applied when working with this board.

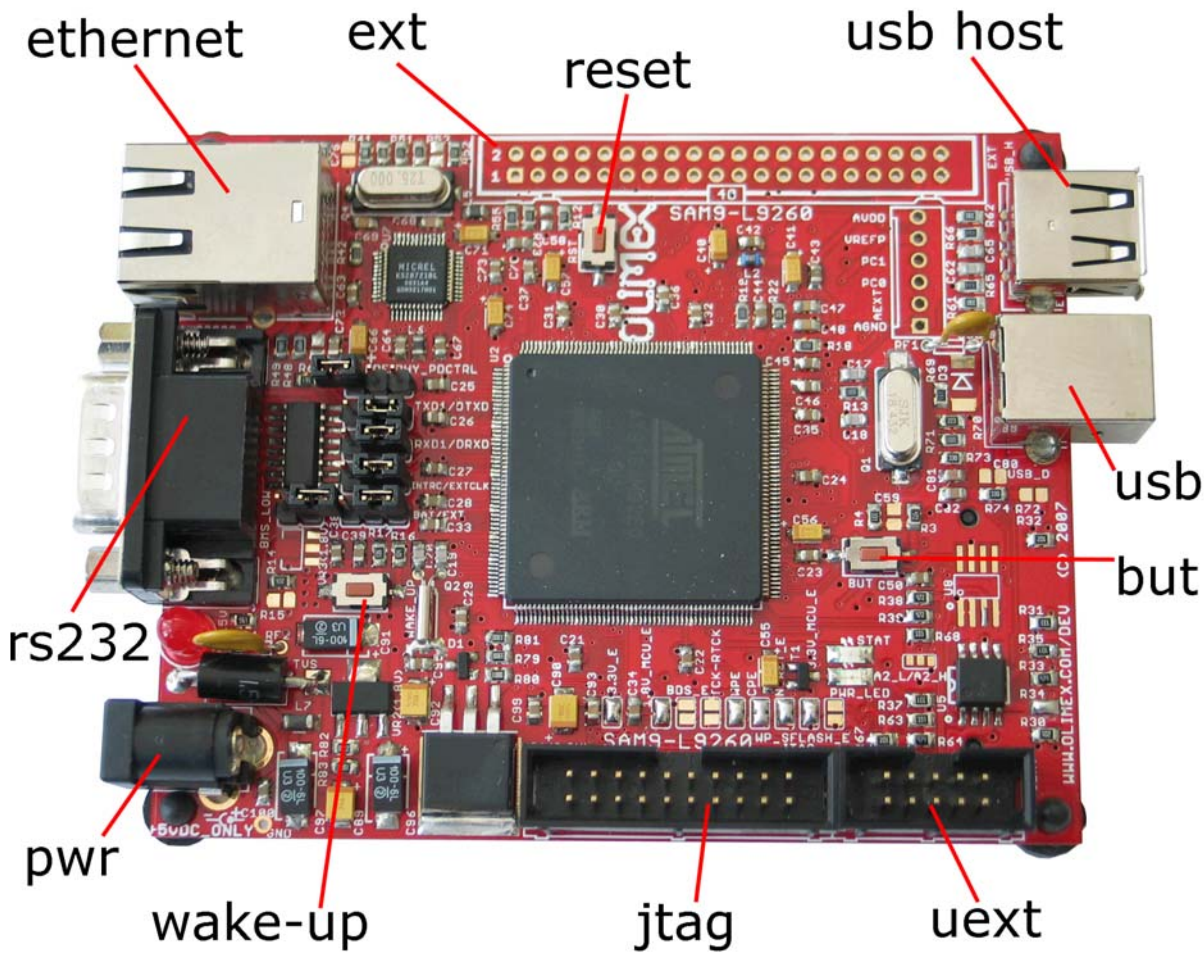
BOARD USE REQUIREMENTS:

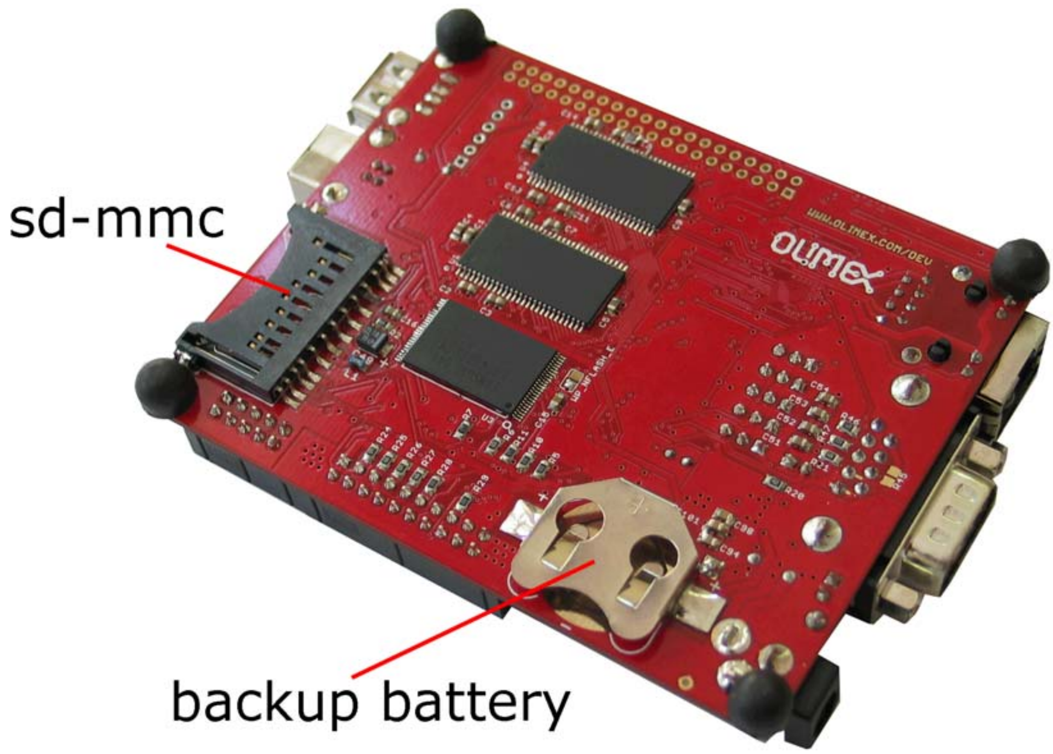
Cables: 1.8 meter USB A-B cable to connect with USB host.
Null modem RS232 female – female to connect with PC COM port.

Hardware: **ARM-JTAG, ARM-USB-OCD** or other compatible tool if you want to program this board with JTAG, usually with Linux installed you can develop without the need for JTAG

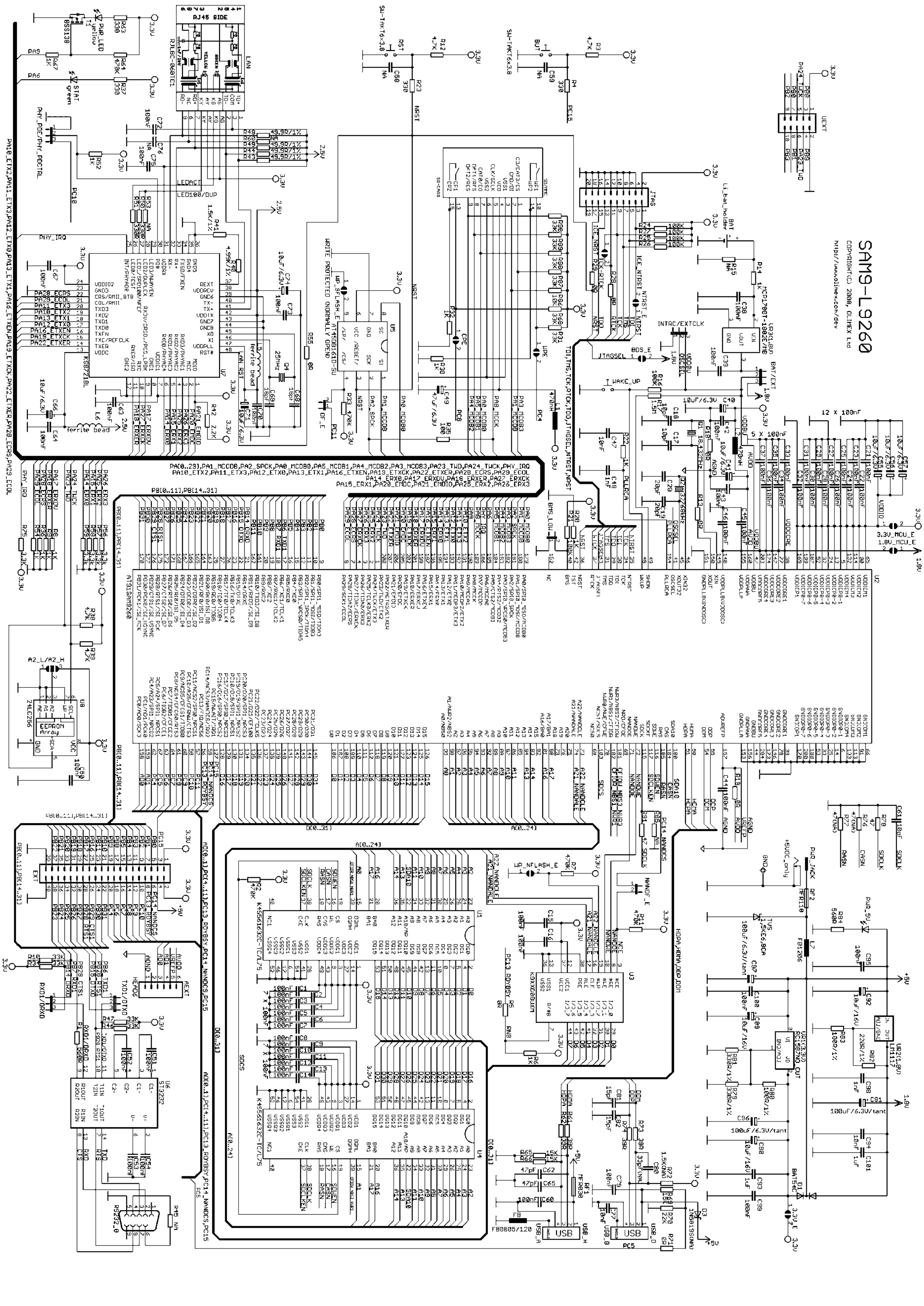
Software: The CD contains Linux 2.6 complete with source and binary on CD

BOARD LAYOUT:





SCHEMATIC:


SAM9-L9260
COPYRIGHT(C) 2008, OLIMEX Ltd
http://www.olimex.com/dev

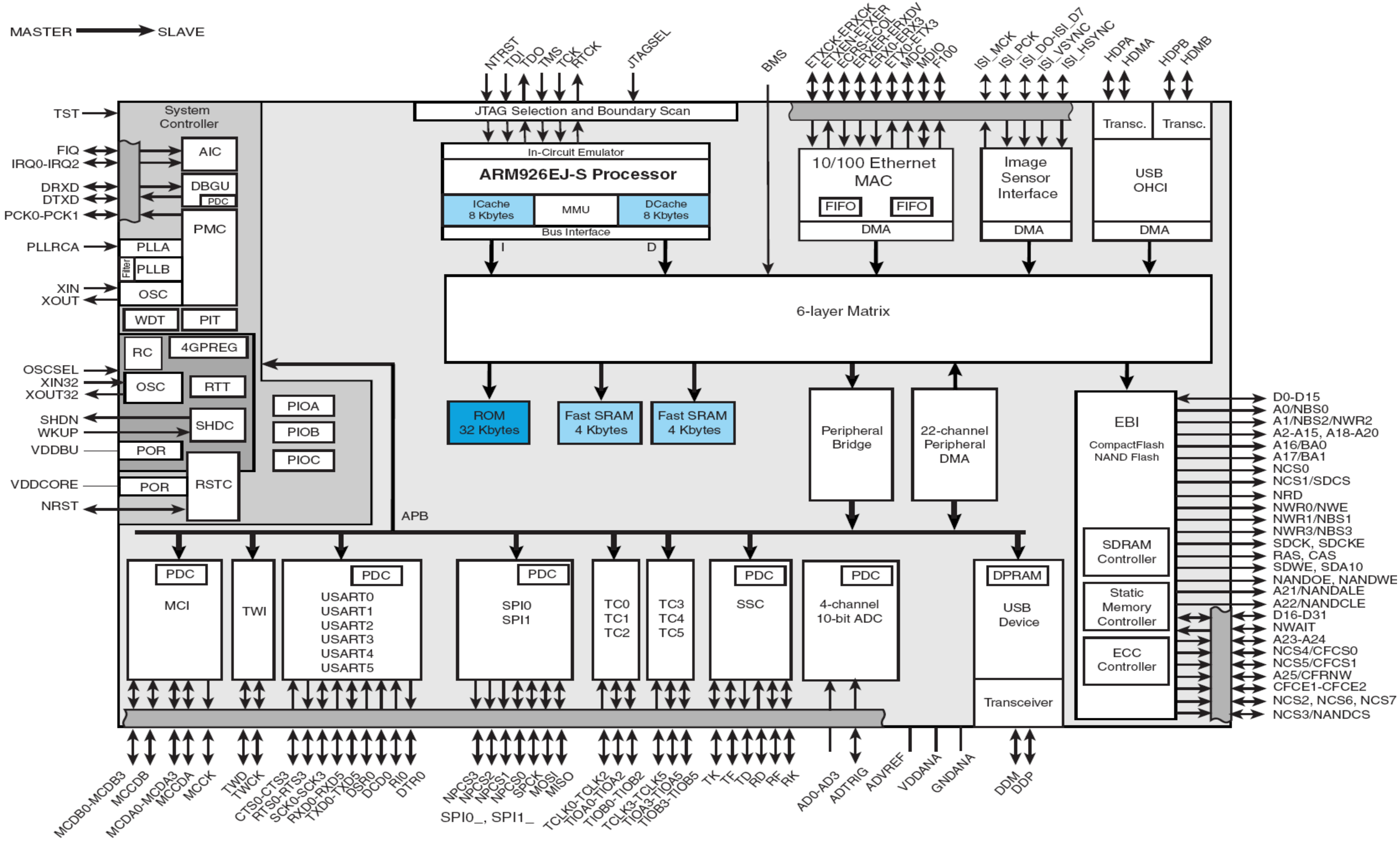
PROCESSOR FEATURES:

SAM9-L9260 board use CPU **AT91SAM9260** from Atmel® with these features:

- Incorporates the ARM926EJ-S™ ARM® Thumb® Processor
  - DSP Instruction Extensions, ARM Jazelle® Technology for Java® Acceleration
- External Bus Interface (EBI)
  - Supports SDRAM, Static Memory, ECC-enabled NAND Flash and CompactFlash®
- USB 2.0 Full Speed (12 Mbits per second) Device Port
  - On-chip Transceiver, 2,432-byte Configurable Integrated DPRAM
- USB 2.0 Full Speed (12 Mbits per second) Host
- Ethernet MAC 10/100 Base T
  - Media Independent Interface or Reduced Media Independent Interface
  - 28-byte FIFOs and Dedicated DMA Channels for Receive and Transmit
- Bus Matrix
  - Six 32-bit-layer Matrix
  - Boot Mode Select Option, Remap Command
- Fully-featured System Controller, including
  - Reset Controller, Shutdown Controller
  - Four 32-bit Battery Backup Registers for a Total of 16 Bytes
  - Clock Generator and Power Management Controller
  - Advanced Interrupt Controller and Debug Unit
  - Periodic Interval Timer, Watchdog Timer and Real-time Timer
- Reset Controller (RSTC)
  - Based on a Power-on Reset Cell, Reset Source Identification and Reset Output Control
- Clock Generator (CKGR)
  - Selectable 32,768 Hz Low-power Oscillator or Internal Low Power RC Oscillator on Battery Backup Power Supply, Providing a Permanent Slow Clock
  - 3 to 20 MHz On-chip Oscillator, One up to 240 MHz PLL and One up to 130 MHz PLL
- Power Management Controller (PMC)
  - Very Slow Clock Operating Mode, Software Programmable Power Optimization Capabilities
  - Two Programmable External Clock Signals
- Advanced Interrupt Controller (AIC)
  - Individually Maskable, Eight-level Priority, Vectored Interrupt Sources
  - Three External Interrupt Sources and One Fast Interrupt Source, Spurious Interrupt Protected
- Debug Unit (DBGU)
  - 2-wire UART and Support for Debug Communication Channel, Programmable ICE Access Prevention
- Periodic Interval Timer (PIT)
  - 20-bit Interval Timer plus 12-bit Interval Counter
- Watchdog Timer (WDT)
  - Key-protected, Programmable Only Once, Windowed 16-bit Counter Running at Slow Clock
- Real-time Timer (RTT)
  - 32-bit Free-running Backup Counter Running at Slow Clock with 16-bit Prescaler

- One 4-channel 10-bit Analog-to-Digital Converter
- Three 32-bit Parallel Input/Output Controllers (PIOA, PIOB, PIOC)
    - 96 Programmable I/O Lines Multiplexed with up to Two Peripheral I/Os
    - Input Change Interrupt Capability on Each I/O Line
    - Individually Programmable Open-drain, Pull-up Resistor and Synchronous Output
    - – High-current Drive I/O Lines, Up to 16 mA Each
- Peripheral DMA Controller Channels (PDC)
- One Two-slot MultiMedia Card Interface (MCI)
    - SDCard/SDIO and MultiMediaCard™ Compliant
    - Automatic Protocol Control and Fast Automatic Data Transfers with PDC
- One Synchronous Serial Controller (SSC)
    - Independent Clock and Frame Sync Signals for Each Receiver and Transmitter
    - I^2S Analog Interface Support, Time Division Multiplex Support
    - High-speed Continuous Data Stream Capabilities with 32-bit Data Transfer
- Four Universal Synchronous/Asynchronous Receiver Transmitters (USART)
    - Individual Baud Rate Generator, IrDA® Infrared Modulation/Demodulation, Manchester Encoding/Decoding
    - Support for ISO7816 T0/T1 Smart Card, Hardware Handshaking, RS485 Support
    - Full Modem Signal Control on USART0
- Two 2-wire UARTs
- Two Master/Slave Serial Peripheral Interfaces (SPI)
    - 8- to 16-bit Programmable Data Length, Four External Peripheral Chip Selects
    - Synchronous Communications
- Two Three-channel 16-bit Timer/Counters (TC)
    - Three External Clock Inputs, Two Multi-purpose I/O Pins per Channel
    - Double PWM Generation, Capture/Waveform Mode, Up/Down Capability
    - High-Drive Capability on Outputs TIOA0, TIOA1, TIOA2
- One Two-wire Interface (TWI)
    - Master, Multi-master and Slave Mode Operation
    - General Call Supported in Slave Mode
- IEEE® 1149.1 JTAG Boundary Scan on All Digital Pins
- Required Power Supplies:
    - 1.65V to 1.95V for VDDBU, VDDCORE and VDDPLL
    - 1.65V to 3.6V for VDDIOP1 (Peripheral I/Os)
    - 3.0V to 3.6V for VDDIOP0 and VDDANA (Analog-to-digital Converter)
    - Programmable 1.65V to 1.95V or 3.0V to 3.6V for VDDIOM (Memory I/Os)

AT91SAM9260 Block Diagram



MEMORY MAP:

Address Memory Space

0x0000 0000	Internal Memories	256M Bytes	0x0FFF FFFF
0x1000 0000	EBI Chip Select 0	256M Bytes	0x1FFF FFFF
0x2000 0000	64M on board SDRAM	64 M Bytes	0x23FFF FFFF
	EBI Chip Select 1/ SDRAMC	192M Bytes	0x2FFF FFFF
0x3000 0000	EBI Chip Select 2	256M Bytes	0x3FFF FFFF
0x4000 0000	EBI Chip Select 3/ NANDFlash Controller	512 M Bytes NANDFlash	0x4FFF FFFF
0x5000 0000	EBI Chip Select 4/ Compact Flash Slot 0	256M Bytes	0x5FFF FFFF
0x6000 0000	EBI Chip Select 5/ Compact Flash Slot 1	256M Bytes	0x6FFF FFFF
0x7000 0000	EBI Chip Select 6	256M Bytes	0x7FFF FFFF
0x8000 0000	EBI Chip Select 7	256M Bytes	0x8FFF FFFF
0x9000 0000	Undefined (Abort)	1,518M Bytes	0xEFFF FFFF
0xF000 0000	Internal Peripherals	256M Bytes	0xFFFF FFFF

Internal Memory Mapping

0x0000 0000	Boot Memory (1)	
0x10 0000	ROM	32K Bytes
0x10 8000	Reserved	
0x20 0000	SRAM0	4K Bytes
0x20 1000	Reserved	
0x30 0000	SRAM1	4K Bytes
0x30 1000	Reserved	
0x50 0000	UHP	16K Bytes
0x50 4000	Reserved	
0x0FFF FFFF		

Notes :

(1) Can be ROM, EBI_NCS0 or SRAM depending on BMS and REMAP

Peripheral Mapping

0xF000 0000	Reserved	
0xFFFA 0000	TCO, TC1, TC2	16K Bytes
0xFFFA 4000	UDP	16K Bytes
0xFFFA 8000	MCI	16K Bytes
0xFFFA C000	TWI	16K Bytes
0xFFFB 0000	USART0	16K Bytes
0xFFFB 4000	USART1	16K Bytes
0xFFFB 8000	USART2	16K Bytes
0xFFFB C000	SSC	16K Bytes
0xFFFC 0000	ISI	16K Bytes
0xFFFC 4000	EMAC	16K Bytes
0xFFFC 8000	SPI0	16K Bytes
0xFFFC C000	SPI1	16K Bytes
0xFFFD 0000	USART3	16K Bytes
0xFFFD 4000	USART4	16K Bytes
0xFFFD 8000	USART5	16K Bytes
0xFFFD C000	TC3, TC4, TC5	16K Bytes
0xFFFE 0000	ADC	16K Bytes
0xFFFE 4000	Reserved	
0xFFFF C000	SYSC	16K Bytes
0xFFFF FFFF		

System Controller Mapping

0xFFFF C000	Reserved	
0xFFFF E800	ECC	512 Bytes
0xFFFF EA00	SDRAMC	512 Bytes
0xFFFF EC00	SMC	512 Bytes
0xFFFF EE00	MATRIX	512 Bytes
0xFFFF EF10	CCFG	
0xFFFF F000	AIC	512 Bytes
0xFFFF F200	DBGU	512 Bytes
0xFFFF F400	PIOA	512 Bytes
0xFFFF F600	PIOB	512 bytes
0xFFFF F800	PIOC	512 bytes
0xFFFF FA00	Reserved	
0xFFFF FC00	PMC	256 Bytes
0xFFFF FD00	RSTC	16 Bytes
0xFFFF FD10	SHDC	16 Bytes
0xFFFF FD20	RTTC	16 Bytes
0xFFFF FD30	PITC	16 Bytes
0xFFFF FD40	WDTC	16 Bytes
0xFFFF FD50	GPBR	16 Bytes
0xFFFF FD60	Reserved	
0xFFFF FFFF		

POWER SUPPLY CIRCUIT:

The power supply for SAM9-L9260 must be regulated +5VDC, please apply exactly 5V as the same power line goes to USB hosts and if you apply over 5V you will damage the USB devices attached to the host.
The current consumption is typical 250mA with 180 MHz clock of SAM9260 and 90MHz clock of external bus.
For the RTC there is a battery backup power supply from a small 3V Li battery type CR2032.

RESET CIRCUIT:

SAM9-L9260 reset circuit contains a 4.7k pull-up resistor and RST button connected to GND.

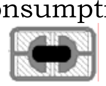
CLOCK CIRCUIT:

Quartz crystal Q1-18.432Mhz is connected to SAM9-L9260 Xin and Xout pins.
Quartz crystal Q2-32768Hz is connected to SAM9-L9260 Xin32 and Xout32 pins.

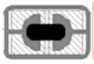
JUMPER DESCRIPTION:

SMD jumper description

3.3V_E This jumper connects the output of main 3.3V regulator VR1(3.3V)-RC1587 to the rest part of schematic. It is useful to measure the current consumption.

<u>Default state - closed</u> 

3.3V_MCU_E This jumper connects 3.3V to the SAM9260 microcontroller. It is useful to measure the current consumption.

<u>Default state - closed</u> 

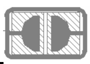
1.8V_MCU_E This jumper connects 1.8V to the SAM9260 microcontroller. It is useful to measure the current consumption.

<u>Default state - closed</u> 

BDS_E BounDary Scan Enable. The BDS_E jumper is used to select the JTAG boundary scan when JTAGSEL pin asserted at a high level (tied to VDDBU). This pin integrates a permanent pull-down resistor of about 15KΩ to GNDBU. When BDS_E is open JTAG function is selected.

<u>Default state – open</u> 

TCK-RTCK Connects RTCK and TCK pins of SAM9260.

<u>Default state open</u> 

WPE Connects PC4(pin62) to Write Protection pin of SD/MMC socket. If WP function is not used, WPE jumper has to be open and PC4 is available of EXT connector pin 20.

Default state - closed 

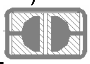
CPE Connects PC8(pin61) to Card Present pin of SD/MMC socket. If CP function is not used, CPE jumper has to be open and PC8 is available of EXT connector pin 14.

Default state - closed 

NTRST_E When the NTRST_E jumper is closed – connects NTRST(pin 35) to JTAG connector (pin3).

Default state - closed 

WP_SFLASH_E When the WriteProtect_SerialFLASH_Enable jumper is closed it allows to protect the boot code written to U5(AT45DB161D-SU) flash memory.

Default state open 

WP_NFLASH_E When the WriteProtect_NandFLASH_Enable jumper is closed user can't write in the NAND flash.

Default state open

A2_L/A2_H Connects Address2(A2)pin of U8-24LC256 memory (default not mounted) to logical 0 or logical 1, i.e. A2_L/A2_H define the memory address of I2C bus.
Default state - open

PTH jumper description:

BMS_LOW Boot Mode Sellect _ LOW jumper select the boot memory External memory or embedded ROM. When BMS_LOW is closed – BMS pin is logical 0, otherwise – logical 1.

Address	REMAP = 0		REMAP = 1
	BMS = 1	BMS = 0	
0x0000 0000	ROM	EBI_NCS0	SRAM0 4K

Default state - open

BMS_LOW

BAT/EXT The BATerry/EXTernal jumper defines the power source which supplied backup logic from VDDBU – pin 47.

BAT position – 3V Li battery type CR2032 plugged in BAT holder supplied VDDBU through backup VR3(1.8V) MCP1700T-1802E/MB voltage regulator.

EXT position – The VDDBU is powered from main 1.8V voltage regulator VR2(1.8V) – LM1117.

Default state BAT/EXT

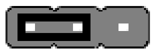
INTRC/EXTCLK

The INTRC/EXTCLK jumper defines SAM9260 slow clock source.

INTRC position – internal RC slow clock oscilator is selected

EXTCLK position – external 32768 crystal is used for SAM9260 slow clock.

Default state INTRC/EXTCLK



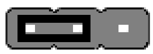
RXD1/DRXD

The RXD1/DRXD jumper defines which pin - RXD1 or DRXD is connected to RS232 driver (ST3232), i.e. the board allows comunication with PC COM port through RXD1 or DRXD.

RXD1 position – RXD1 function of SAM9260 pin 18 is tied to pin12(R1OUT) of U6(ST3232).

DRXD position – DRXD function of SAM9260 pin 21 is tied to pin12(R1OUT) of U6(ST3232).

Default state RXD1/DRXD



TXD1/DTXD

The TXD1/DTXD jumper defines which pin TXD1 or DTXD is connected to RS232 driver (ST3232), i.e. the board allows comunication with PC COM port through TXD1 or DTXD.

T**XD1** position – TXD1 function of SAM9260 pin 17 is tied to pin11(T1IN) of U6(ST3232).

DTXD position – DTXD function of SAM9260 pin 22 is tied to pin11(T1IN) of U6(ST3232).

Default state TXD1/DTXD

PHY_PDE/PHY_PDCTRL

PHY_PDE position – The PHY chip U7(KS8721BL) enter to power down mode.

PHY_PDCTRL position – The PHY chip power down mode is controled from SAM9260 PC1(pin58).

OPEN position – The PHY chip is always enabled.

Default state- open   PHY_PDE/PHY_PDCTRL

NANDF_E The NANDFlash_Enable allows PC14/NAND_CS pin of SAM9260 to controls CE pin of NAND FLASH memory U3(K9F4G08UXM). If the board has to boot from NAND flash the NANDF_E jumper must be closed.

Default state- close   NANDF_E

DF_E The DataFlash_Enable allows PC11/SPI0_NPCS1 pin of SAM9260 to controls CS pin of serial Data Flash memory U5(AT45DB161D-SU). If the board has to boot from Data Flash the DF_E jumper must be closed.

Default state- open   DF_E

INPUT/OUTPUT:

RS232_0 is used as the Linux console, so you can connect with a terminal program and work at command prompt.

The cable between SAM9-L9260 and PC must be female – female, null modem type. Terminal settings are 115200 , 8bits, 1stop, no parity, no flow control.

User button named **BUT** – connected to SAM9260 pin127 PC15(IRQ1);

Status green LED with name **STAT** (SAM9260 pin185 PA6) is connected to the system timer and blinks every second.

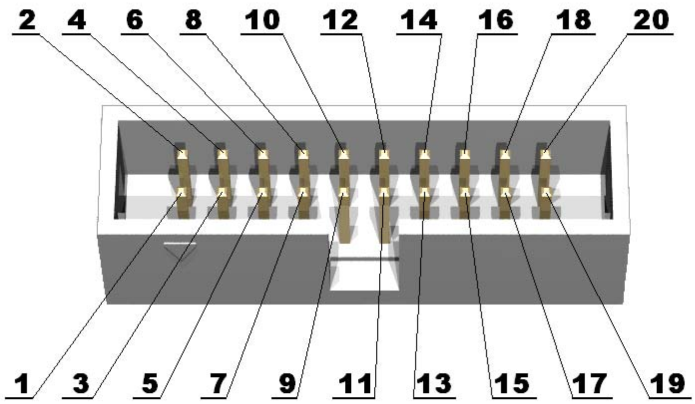
Power supply yellow LED with name **PWR_LED** indicates the state of SAM9260. It is linked to the CPU load and is on when the CPU is idle.

The LED **PWR_5V** (red) indicates +5V presence to the board.

EXTERNAL CONNECTOR DESCRIPTION:

JTAG:

The JTAG connector allows software debugger to talk via a JTAG (Joint Test Action Group) port directly to the core. Instructions may be inserted and executed by the core thus allowing SAM9260 memory to be programmed with code and executed step by step by the host software.
For more details refer to IEEE Standard 1149.1 - 1990 Standard Test Access Port and Boundary Scan Architecture and SAM9260 datasheets and users manual.

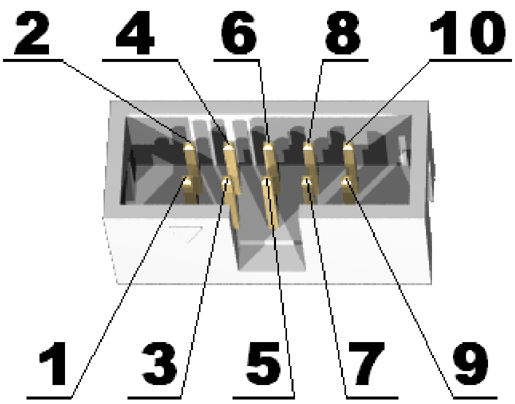


Pin #	Signal Name	Pin #	Signal Name
1	VCC	2	VCC
3	ICE_NTRST	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	RTCK	12	GND
13	TDO	14	GND
15	ICE_NRST	16	GND
17	NC	18	GND
19	NC	20	GND

Standard 20 pin ARM JTAG connector for programming and debugging.

UEXT

Pin #	Signal Name
1	VCC
2	GND
3	PB8(TXD2)
4	PB9(RXD2)
5	PA24_TWCK
6	PA23_TWD
7	PB0(SPI1_MISO)
8	PB1(SPI1_MOSI)
9	PB2(SPI1_SPCK)
10	PB3(SPI1_NPCS0)



UEXT is a universal OLIMEX connector with 3.3V power supply and UART, I2C and SPI interface. Other device or modules with these interfaces can connected with UEXT. For example: MOD-NRF24L, MOD-RFID125, MOD-MP3 and many other are on the way

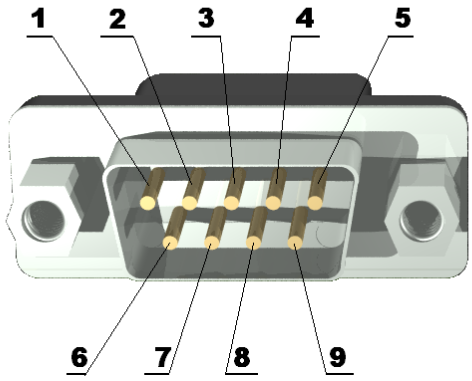
SPI: The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Master or Slave Mode. It also enables communication between processors if an external processor is connected to the system.

I2C: The Atmel Two-wire Interface (TWI) interconnects components on a unique two-wire bus, made up of one clock line and one data line with speeds of up to 400 Kbits per second, based on a byte-oriented transfer format. It can be used with any Atmel two-wire bus Serial
EEPROM. The TWI is programmable as a master or a slave with sequential or single-byte access. Multiple master capability is supported. Arbitration of the bus is performed internally and puts the TWI in slave mode automatically if the bus arbitration is lost. A configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.

UART: The Universal Synchronous Asynchronous Receiver Transmitter (USART) provides one full duplex universal synchronous asynchronous serial link. Data frame format is widely programmable (data length, parity, number of stop bits) to support a maximum of standards. The receiver implements parity error, framing error and overrun error detection. The receiver timeout enables handling variable-length frames and the transmitter timeguard facilitates communications with slow remote devices. Multidrop communications are also supported through address bit handling in reception and transmission.

RS232 connector:

Pin #	Signal Name
1	NC
2	RXD
3	TXD
4	6
5	GND
6	4
7	RTS
8	CTS
9	NC



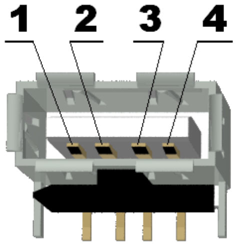
USB_D

The USB Device Port (UDP) is compliant with the Universal Serial Bus (USB) V2.0 full-speed device specification. Each endpoint can be configured in one of several USB transfer types. It can be associated with one or two banks of a dual-port RAM used to store the current data payload. If two banks are used, one DPR bank is read or written by the processor, while the other is read or written by the USB device peripheral. This feature is mandatory for isochronous endpoints. Thus the device maintains the maximum bandwidth (1M bytes/s) by working with endpoints with two banks of DPR.

Pin #	Signal Name
1	+5V(input)
2	USBDM
3	USBDP
4	GND

USB_A:

Pin #	Signal Name
1	+5V(Out)
2	HDMA
3	HDPA
4	GND

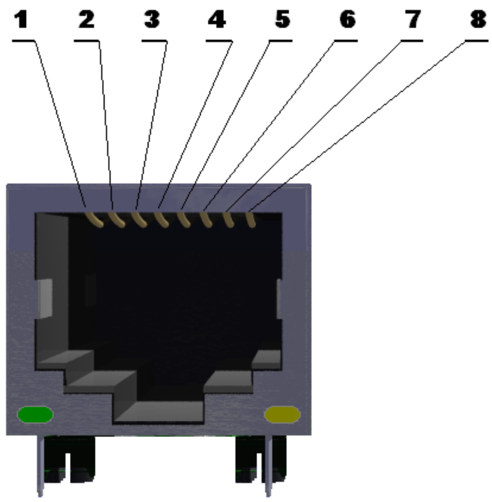


USB host with +5V power line direct connected through 300mA resetable fuse to power jack.
The USB Host Port (UHP) interfaces the USB with the host application. It handles Open HCI protocol (Open Host Controller Interface) as well as USB v2.0 Full-speed and Low-speed protocols. The USB Host Port integrates a root hub and transceivers on downstream ports. It provides several high-speed half-duplex serial communication ports at a baud rate of 12 Mbit/s. Up to 127 USB devices (printer, camera, mouse, keyboard, disk, etc.) and the USB hub can be connected to the USB host in the USB "tiered star" topology. The USB Host Port controller is fully compliant with the Open HCI specification. The standard OHCI USB stack driver can be easily ported to Atmel's architecture in the same way all existing class drivers run without hardware specialization.

LAN:

The EMAC module implements a 10/100 Ethernet MAC compatible with the IEEE 802.3 standard using an address checker, statistics and control registers, receive and transmit blocks, and a DMA interface. The address checker recognizes four specific 48-bit addresses and contains a 64-bit hash register for matching multicast and unicast addresses. It can recognize the broadcast address of all ones, copy all frames, and act on an external address match signal.

Pin #	Signal Name
1	TD+
2	TD-
3	RD+
4	GND_LAN
5	GND_LAN
6	RD-
7	GND_LAN
8	GND_LAN



LED	Color	Usage

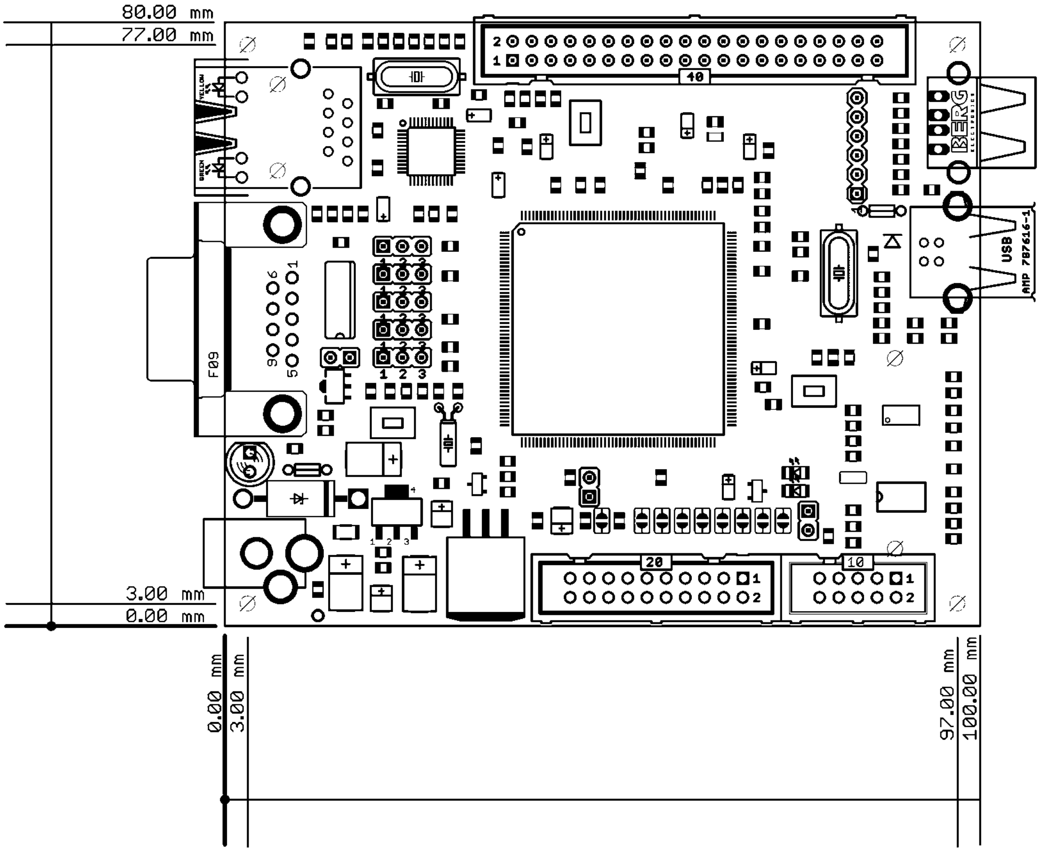
Right	Yellow	Activity
Left	Green	100MBits/s (Half/Full duplex)

EXT:

Pin #	Signal Name	Pin #	Signal Name
1	3.3V(OUT)	2	3.3V(OUT)
3	PC15/NWAIT/IRQ1	4	+5V(OUT)
5	PB0/SPI1_MISO/TIOA3	6	PC14/NCS3/NANDCS/IRQ2 NAND Flash Chip select MCU out
7	PB1/SPI1_MOSI/TIOB3	8	PC13/FIQ/NCS6 NAND Flash RDY/BSY MCU in
9	PB2/SPI1_SPCK/TIOA4	10	PC10/A25/CFRNW/CTS3
11	PB3/SPI1_NPCS0/TIOA5	12	PC9/NCS5/CFCS1/TIOB0
13	PB4/TXD0	14	PC8/NCS4/CFCS0/RTS3
15	PB5/RXD0	16	PC7/TIOB1/CFCE2
17	PB8/TXD2	18	PC6/TIOB2/CFCE1
19	PB9/RXD2	20	PC4/A23/SPI1_NPCS2
21	PB10/TXD3/ISI_D8	22	PB31/PCK1/ISI_MCK
23	PB11/RXD3/ISI_D9	24	PB30/PCK0/ISI_HSYNC
25	PB16/TK0/TCLK3	26	PB29/CTS1/ISI_VSYNC RS232_0 CTS – MCU input
27	PB17/TF0/TCLK4	28	PB28/RTS1/ISI_PCK RS232_0 RTS – MCU output
29	PB18/TD0/TIOB4	30	PB27/CTS0/ISI_D7
31	PB19/RD0/TIOB5	32	PB26/RTS0/ISI_D6
33	PB20/RK0/ISI_D0	34	PB25/RI0/ISI_D5
35	PB21/RF0/ISI_D1	36	PB24/DTR0/ISI_D4
37	PB22/DSR0/ISI_D2	38	PB23/DCD0/ISI_D3
39	GND	40	GND

This is provision for 40 pin external connector (not mounted) with all unused SAM9260 ports available for user applications and add-on boards.

MECHANICAL DIMENSIONS:



SOFTWARE development with SAM9-L9260:

Overview

The board comes with Linux preloaded in the NAND flash. It's based on a custom-built kernel and a Debian 4.0 userland. To use it, connect a null-modem cable to the board and to a serial port on your computer, start a terminal program (e.g. HyperTerminal on Windows, minicom on Unix systems) and configure it to use a 115200 baud rate, 8 data bist, 1 stop bit and no parity and no flow control. Then apply power to the board (use a 5VDC regulated power supply with at least 500mA output current) and you should see the board start-up messages.

The default root password is 'olimex'

Restoring the default bootloader and kernel

If for some reason you need to restore the default factory configuration of the board, the procedure is as follows:

First install the ATMEL AT91-ISP package which comes on the disk. Reboot the computer if needed and copy the *NANDFLASH.tcl* file from the ATMEL directory on the CD to "*C:\Program Files\ATMEL Corporation\AT91-ISP v1.9\SAM-BA v2.5\lib\AT91SAM9260-EK*".

Remove the NANDF_E and DF_E jumpers on the SAM9-L9260 board and power it up. Connect an USB cable to the USB_D connector on the board and wait for the board to be detected (if necessary install the USB driver, available in the "*atmel/samba driver/*" directory).

Close the NANDF_E jumper and run the *AT91SAM9260_demo_linux_NandFlash.bat* file from the sam9-l9260-samba directory. After a while the log file will be displayed and the system should be restored to the default state.

WARNING! This procedure erases the whole NAND flash and the root filesystem will also be destroyed in the process.

Restoring the on-board root filesystem

Boot-up the board with an alternate root filesystem (e.g. a USB flash drive, NFS exported filesystem...) and use the following command (assuming that the rootjffs2.img file is available in.)

sam9-l9260:~# *flash_eraseall -j /dev/mtd1*
sam9-l9260:~# *nandwrite -a /dev/mtd1 /rootjffs2.img*

You may get some errors about bad blocks not being erased - this is normal and is related to the priciple of operation of NAND flashes. After the process is completed, reboot the board.

Running with another root filesystem

You may choose to use another media for the root filesystem for various reasons - more capacity, faster access, etc. A complete root tree is archived in the root_sam9_fs.tgz file which can be extracted to an empty ext3 partition on an USB drive or to some exported via NFS directory. Then you need to tell the kernel where to find the root - this is accomplished by interrupting the u-boot process at the "Hit any key to stop autoboot:..." prompt and setting the bootargs variable. For example, to boot from a USB flash drive, the command is:

U-Boot> *setenv bootargs mem=64M console=ttyS0,115200 root=/dev/sda1 rootdelay=10*

and for booting from an NFS server at adress 192.168.0.75:

U-Boot> *setenv bootargs mem=64M console=ttyS0,115200 root=/dev/nfs nfsroot=192.168.0.75:nfsroot ip=192.168.0.222:192.168.0.75*

Building a custom kernel

The recommended build method is to use a cross-compiler. Building natively should also work but will be very time-consuming. At the moment of writing, the current kernel version was 2.6.23, of which a tarball is provided. You also need to apply the two patches in the linux directory - the 2.6.23-rc3-at91.patch needs to be applied first. It adds general support for at91-based boards to the kernel. The sam9_l9260.diff adds support for the Olimex SAM9-L9260 board and should be applied second. After that, you can build the default kernel by typing

$ *make ARCH=arm CROSS_COMPILE=arm-linux- sam9_l9260_defconfig*
$ *make ARCH=arm CROSS_COMPILE=arm-linux- uImage*

After the compilation, the kernel should be available at arch/arm/boot/uImage
If the build process fails to detect the mkimage program, it is available in the u-boot archive
The new kernel can be transferred to the board by various means - e.g. use the board restoration process and change the kernel in there, tftpboot-ing the board, etc.

Building the bootstrap binary

Extract the sources from boot/at91bootstrap-2.3-olimex_18.04.08.tgz to your working directory and issue the following commands (substituting <your-compiler-prefix-> with e.g. arm-linux-)
$ *make CROSS_COMPILE=<your-compiler-prefix-> MEMORY=nandflash at91sam9260ek_defconfig*
$ *make CROSS_COMPILE=<your-compiler-prefix-> MEMORY=nandflash*

If everything is correct, the resulting binary file will be located in the /binaries directory.

Building U-Boot

Extract the sources from boot/u-boot-1.2.0-atmel-olimex_18_04_08.tgz, and issue:

$ *make CROSS_COMPILE=<your-compiler-prefix-> sam9l9260_config*
$ *make CROSS_COMPILE=<your-compiler-prefix->*

WARNING! Some compiler revisions seem to produce incorrect code and although the compilation succeedes, the resultant u-boot binary is not working - it will either hang completely without any output or will restart when detecting the NAND flash. The only revision that is known to work reliably is gcc-3.4.3, while gcc-4.0.1 and 4.1.1 are known to be broken.

Cross-compiling a simple "hello world" example

Extract one of the provided cross-compilers on your host system and add it to the PATH variable. Use the cross-compiler to build the example, then transfer it to the board by e.g. USB flash drive, http download etc. Example commands:

```
----- On the host system -----
$ PATH=$PATH:/usr/local/arm/4.1.1-920t/bin
$ cat > hello.c
#include <stdio.h>
int main(void)
{
	unsigned int i;
	printf("\r\nProba proba ");
	for (i=0; i<10; i++)
		printf("\r\n%d", i);
	return 0;
}
^D
$ arm-linux-gcc -o hello hello.c
$ cp hello ~/htdocs/
----- On the board -----
~ # wget http://192.168.0.xx/hello
```

~ # chmod 777 hello
~ # ./hello
Proba proba
0
1
....

Using JTAG to program the board

A sample project is provided in the "TEST_BUTT" directory that demonstrates how to write a project that runs directly on the core, without the need of an operating system. It was developed using IAR Embedded Workbench for ARM ver. 4.42A with a Segger J-Link JTAG adapter

Common Questions

Q: When booting from the internal NAND flash the board seems to hang at "INIT: version 2.86 booting" and/or "Activating swap...done" lines
A: When mounting the JFFS2 root filesystem, the system performs a consistency check (similar to fsck). This almost blocks all access to the nand flash and the system appears to hang. Please wait - on a first boot of a new filesystem this could take up to 5 minutes and is normal.

Q: There are messages "Buffer I/O error on device mtdblock0, logical block 0;end_request: I/O error, dev mtdblock0, sector 0" during boot-up. Is there a problem on the board?
A: These messages indicate incorrect OOB records in the part of the flash where the bootloader is stored and are due to the version of SAM-BA which is used to write the various parts of the bootloader. We consider the above messages as harmless.

Q: Why use an older version of SAM-BA with the OOB problem described above?
A: Because the used u-boot bootloader is unable to understand the (correct) new OOB layout. This should be corrected in the new revisions.

Q: The I/O operations are slow when using the on-board nand flash or USB flash drive.
A: When doing a sequential read/write (e.g. one single large file) flash memories can be fast. When reading/writing many small files the performance will be really low.

Q: How to boot from the on board DataFlash?
A: Make sure that NANDF_E jumper is not connected and DF_E jumer is connected. If the dataflash has been correctly programmed, the board should start up.

Q: Is the SD/MMC card supported?
A: The SD/MMC card is fully supported, including detection of card insertion/removal and write lock

Q: What do the two LED's indicate?
A: These two leds are driven by default by the linux LED driver. The green one is connected to the system timer and blinks every second. The second one(orange) is linked to the CPU load and is on when the CPU is idle.

Q: The system time is lost after reset, how to avoid that?
A: Unfortunately the Linux AT91SAM9 RTC driver is not yet operational. When it is completed, you would just need a standard 3V battery at the socket at the back of the board. Until then please set the date manually or use a network time synchronization utility as ntpdate.

Q: What is the default IP address of the board?
A: The board is configured to use DHCP to automatically detect it's address. If a DHCP server is not available, it will fallback to using 192.168.0.220

Acknowledgemens:

The kernel used is based on Linux-2.6.23 with patches available from http://maxim.org.za

The root filesystem is based on the NSLU2 debian filesystem, available from http://www.cyrius.com/debian/nslu2/

The bootloader is based on the u-boot-1.2.0 and at91bootstrap-2.3 packages, provided by ATMEL at http://www.at91.com

The cross-compilers are available from various sources on the Internet, namely http://arm.cirrus.com/

All of the above packages are distributed under the GPL and/or another free license (e.g. BSD license).

ORDER CODE:

SAM9-L9260 – assembled and tested (no kit, no soldering required)

How to order?
You can order to us directly or by any of our distributors.
Check our web www.olimex.com/dev for more info.

Pb-free, Green All boards produced by Olimex are RoHS compliant

Revision history:

REV.A - create April 2008