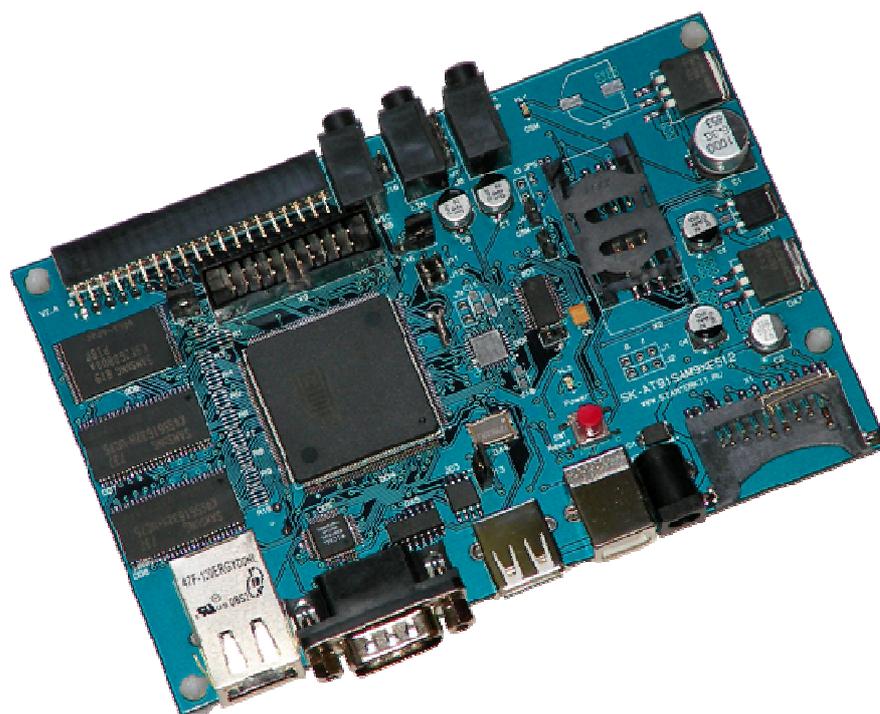


# Отладочная плата SK-AT91SAM9260-SIM300/SIM508

## Инструкция пользователя



**SK-AT91SAM9260-SIM300:**

Atmel AT91SAM9260 (ARM9 200МГц)  
SDRAM 64Мбайт (32Мx16)  
NAND Flash 256Мбайт  
SPI DataFlash 4Мбайт  
100/10М Ethernet  
SD/MMC держатель  
Аудио стерео CODEC TLV320  
USB Device  
USB Host  
RS232  
JTAG разъем  
Разъем расширения  
Посадочное место для GSM SIM300  
Держатель SIM карты (используется GSM модемом)  
Система питания

**SK-AT91SAM9260-SIM508:**

Atmel ARM9 AT91SAM9260 (ARM9 200МГц)  
SDRAM 64Мбайт (32Мx16)  
NAND Flash 256Мбайт  
SPI DataFlash 4Мбайт  
100/10М Ethernet  
SD/MMC держатель  
Аудио стерео CODEC TLV320  
USB Device  
USB Host  
RS232  
JTAG разъем  
Разъем расширения  
Посадочное место для GSM/GPS SIM508  
Держатель SIM карты (используется GSM модемом)  
Система питания

**Возможность прямого подключения:** SK-VideoADC-Plug, SK-WF43BTIBED0-Plug, SK-WF43BTIBED0TP-Plug

**Комплект поставки:** отладочная плата SK-AT91SAM9260-SIM300/SIM508, RS232 кабель, DVD диск

## 1. Общие характеристики

- Напряжение питания: 5-12В, при использовании USB-host 6В максимум, рекомендуемое напряжение 5В
- Потребляемый ток до 1А
- Габариты 136x90x20мм

## 2. Назначение джамперов

1-ый вывод перемычек и переключающих перемычек помечен квадратной контактной площадкой.

- J1,J2 переключают UART порты, А или В, GPS приемника ко входу процессора.
- J3 позволяет подключить внешнюю батарею для GPS приемника (для того чтобы не терять альманах во время краткосрочной потери питания)
- J4 позволяет подключить внешнюю батарею для GSM модема (RTC)
- J5 аудио выход GSM модема
- J6 аудио выход кодека TLV320
- J7 дублирует X4 - линейный выход с аудио кодеков
- J8 дублирует X6 - микрофонный вход аудио кодеков
- J9 выход усилителя мощности кодека AT73C213
- J10 дублирует X8 - линейный вход аудио кодеков
- J11 подключается к входу WKUP процессора
- J13 позволяет разорвать цепь загрузки из SPI DataFlash
- J14 позволяет разорвать цепь загрузки из NAND Flash
- J15 позволяет подключить внешний источник опорного напряжения для внутреннего АЦП процессора

По умолчанию замкнуты перемычки: J1-2 (положение 1-2), J13, J14, J16 (положение 1-2)

## 3. Начало работы

Подключите RS232 кабель, идущий в комплекте, к COM порту PC (или USB-COM преобразователю), настройте терминальную программу на используемый COM порт с параметрами 115200 без управления потоком.

Подключите сетевой (Ethernet) кабель, настройте IP адрес сетевой карты PC в диапазоне 192.168.0.XXX.

При необходимости, подключите SK-WF43BTIBED0TP-Plug или SK-WF43BTIBED0-Plug к разъему X11.

Подключите питание (питающее напряжение – центральная жила разъема), в терминальной программе появятся следующие сообщения:

```
U-Boot 2010.03 (Apr 21 2010 - 18:56:06)

DRAM: 64 MB
NAND: 256 MiB
DataFlash:AT45DB321
Nb pages: 8192
Page Size: 528
Size= 4325376 bytes
Logical address: 0xC0000000
Area 0: C0000000 to C00041FF (RO) Bootstrap
Area 1: C0004200 to C00083FF Environment
Area 2: C0008400 to C0041FFF (RO) U-Boot
```

```

Area 3: C0042000 to C0251FFF      Kernel
Area 4: C0252000 to C041FFFF      FS
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   macb0
macb0: PHY present at 1
macb0: Starting autonegotiation...
macb0: Autonegotiation complete
macb0: link up, 100Mbps full-duplex (lpa: 0xcdel)
Hit any key to stop autoboot:  0

NAND read: device 0 offset 0x60000, size 0x780000
7864320 bytes read: OK
## Booting kernel from Legacy Image at 20400000 ...
   Image Name:     Linux Kernel Image
   Image Type:     ARM Linux Kernel Image (gzip compressed)
   Data Size:      4438174 Bytes = 4.2 MB
   Load Address:  20008000
   Entry Point:    20008000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK

Starting kernel ...

Linux version 2.6.28 (user@debian) (gcc version 4.2.0 20070413 (prerelease) (CodeSourcery
Sourcery G++ Lite 2007q1-10)) #46 Sat Mar 27 17:36:08 EDT 2010
CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177
CPU: VIVT data cache, VIVT instruction cache
Machine: Starterkit.ru SK-AT91SAM9XXXXX-SIMXXX
Memory policy: ECC disabled, Data cache writeback
Clocks: CPU 200 MHz, master 100 MHz, main 12.000 MHz
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 16256
Kernel command line: root=/dev/ram0 rw initrd=0x23100000,0x800000 console=ttyS0,115200
mem=64M
AT91: 96 gpio irqs in 3 banks
PID hash table entries: 256 (order: 8, 1024 bytes)
Console: colour dummy device 80x30
console [ttyS0] enabled
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 64MB = 64MB total
Memory: 50564KB available (2988K code, 238K data, 2880K init)
Calibrating delay loop... 99.73 BogoMIPS (lpj=498688)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
net_namespace: 288 bytes
NET: Registered protocol family 16
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 2048 (order: 2, 16384 bytes)
TCP bind hash table entries: 2048 (order: 1, 8192 bytes)
TCP: Hash tables configured (established 2048 bind 2048)
TCP reno registered
NET: Registered protocol family 1
checking if image is initramfs...it isn't (bad gzip magic numbers); looks like an initrd
Freeing initrd memory: 8192K
NetWinder Floating Point Emulator V0.97 (double precision)
JFFS2 version 2.2. (NAND) (SUMMARY) ,† 2001-2006 Red Hat, Inc.
yaffs Mar 27 2010 12:35:28 Installing.
msgmni has been set to 114
io scheduler noop registered (default)
Console: switching to colour frame buffer device 60x34
fb0: Hecuba frame buffer device, using 510K of video memory
atmel_usart.0: ttyS0 at MMIO 0xfefff200 (irq = 1) is a ATMEL_SERIAL
atmel_usart.1: ttyS1 at MMIO 0xffffb0000 (irq = 6) is a ATMEL_SERIAL
atmel_usart.2: ttyS2 at MMIO 0xffffb8000 (irq = 8) is a ATMEL_SERIAL
brd: module loaded
loop: module loaded
ssc ssc.0: Atmel SSC device at 0xc48f8000 (irq 14)
MACB_mii_bus: probed
eth0: Atmel MACB at 0xffffc4000 irq 21 (00:1f:f2:00:00:00)
eth0: attached PHY driver [Generic PHY] (mii_bus:phy_addr=ffffff:01, irq=-1)
Driver 'sd' needs updating - please use bus_type methods
NAND device: Manufacturer ID: 0xec, Chip ID: 0xda (Samsung NAND 256MiB 3,3V 8-bit)
Scanning device for bad blocks
Bad eraseblock 99 at 0x00c60000

```

```

Bad eraseblock 607 at 0x04be0000
Bad eraseblock 1433 at 0x0b320000
Bad eraseblock 1512 at 0x0bd00000
Bad eraseblock 1784 at 0x0df00000
Creating 2 MTD partitions on "atmel_nand":
0x00000000-0x00800000 : "Partition 1"
0x00800000-0x10000000 : "Partition 2"
atmel_spi atmel_spi.1: Atmel SPI Controller at 0xffffcc000 (irq 13)
usbmon: debugfs is not available
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
at91_ohci at91_ohci: AT91 OHCI
at91_ohci at91_ohci: new USB bus registered, assigned bus number 1
at91_ohci at91_ohci: irq 20, io mem 0x00500000
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
udc: at91_udc version 3 May 2006
mice: PS/2 mouse device common for all mice
input: HP Jornada touchscreen as /class/input/input0
i2c /dev entries driver
usbcore: registered new interface driver usbhid
usbhid: v2.6:USB HID core driver
Advanced Linux Sound Architecture Driver Version 1.0.18rc3.
tlv320aic23b spi1.0: tlv320aic23b: supported bitrate is 47348 (66 divider)
ALSA device list:
  #0: AT91SAM9260 external CODEC on irq 14
TCP cubic registered
NET: Registered protocol family 17
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
Freeing init memory: 2880K
Initializing random number generator... done.
Starting network...
route: SIOCADDRT: File exists
Starting vsftpd: start-stop-daemon: applet not found
yaffs: dev is 32505857 name is "mtdblock1"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.1, "mtdblock1"
yaffs: auto selecting yaffs2
yaffs_read_super: isCheckpointed 0
tlv320aic23b: playback bitrate is 47348 (66 divider)
eth0: link up (100/Full)

Welcome to SK-AT91SAM9XXXXX-SIMXXX development board!
SK-AT91SAM9 login:

```

Что означает, что система успешно загрузилась и готова к работе, во конце загрузки, при подключении наушников к X4, будет слышен музыкальный проигрывш.

Для входа в консоль введите имя пользователя root, пароль не требуется (других пользователей в системе нет), после чего имеете полный консольный доступ к системе. Так же можно подключиться с помощью Telnet, FTP, HTTP, сетевой адрес платы 192.168.0.136. При подключении-отключении USB, SD/MMC карт памяти, они будут автоматически монтироваться-размонтироваться в системе.

Если был подключен SK-WF43BTIBEDO-Plug, на экране появится графическое изображение и сообщение о старте системы, при подключении SK-WF43BTIBED0TP-Plug, при нажатии на сенсорный экран панели, будут индексироваться координаты нажатия.

#### 4. Состав ОС Linux

Ядро 2.6.28, включая драйвера:

- Ethernet
- NAND flash
- DataFlash
- USB-host
- USB-gadget

- Audio
- I2C
- SPI
- UART
- RTC
- WatchDog
- FB WF43BTIBED0
- TP ADS7843
- ...

Корневая файловая система (ФС), в поставляемом варианте платы, хранится в NAND flash и после загрузки располагается в оперативной памяти (SDRAM), поэтому, любые ее изменения не повлияют на ее состояние при перезагрузке. Хранить данные можно в NAND flash, которая монтируется при старте системы /mnt/nand – тип файловой системы YAFFS2.

Корневая ФС содержит набор базовых приложений (большинство из которых являются реализацией мультифункционального приложения BusyBox), содержит:

- HTTPD – сервер HTTP
- FTPD – сервер FTP
- Telnetd – сервер Telnet
- TFTP – утилита приема-передачи файлов по TFTP протоколу
- Z-modem утилиты (для обмена файлами через COM порт)
- Tip – терминальная программа
- TS-lib – набор утилит для операций с сенсорной панелью
- Memtester – тест памяти
- Madplay – проигрыватель MP3 файлов
- ...

## 5. Виртуальная машина VMware

Для сборки ядра и корневой ФС используется виртуальная машина VMware с установленной ОС Debian, в состав которой входят все исходные тексты, компилятор и утилиты для сборки (toolchain), скрипты. Так же в виртуальной машине установлены и настроены сервисы для удобства взаимодействия с «материнской» ОС и отладочной платой: SSH, FTP, TFTP.

Разархивируйте файл “SK-AT91SAM9XXXXX-SIMXXX linux build machine.rar”, установите VMware-player или VMware, откройте и проект виртуальной машины.

Для работы необходимо настроить сетевые интерфейсы (появляющиеся после установки VMware), присвоив им описываемые ниже IP адреса:

Eth0 (Bridget) с адресом 192.168.0.2, задуман для взаимодействие с платой, для загрузки образов по TFTP ... Т.е. для нормальной работы, потребуется присвоить IP адрес PC сетевой карты (к которой подключается отладочная плата) 192.168.0.1

Eth1 (Host-only) с адресом 192.168.2.2, задуман для взаимодействия с PC (т.к. Bridget интерфейс отключается при физически выключенном кабеле), в частности, для возможности копирования файлов из виртуальной системы по FTP. В свойствах сетевых устройств, этому виртуальному адаптеру нужно присвоить IP 192.168.2.1

После правильной настройки (и с подключенной платой) должны успешно проходить PING с PC по адресам 192.168.2.2, 192.168.0.2, 192.168.0.136.

После того как сетевые интерфейсы настроены, можно запускать виртуальную машину, после загрузки ее не обязательно выключать, достаточно будет нажать кнопку паузы и во время следующего сеанса работы не придется ждать загрузки виртуальной ОС, но при этом, в некоторых случаях, нужно следить за системными часами, особенно при копировании новых файлов (имеющих более позднюю дату создания относительно системы) для сборки.

По умолчанию, в системе присутствует два пользователя:

- root, пароль 123456
- user, пароль 123456 (настоятельно рекомендую работать под этим пользователем, или создать нового, но не вести всю работу под root)

После входа переключаемся на консоль (Ctrl+Alt+F(1-6)) (потребуется в опциях VMware освободить сочетание клавиш Ctrl+Alt - по умолчанию это выход из окна виртуальной машины), запускаем MidnightComander (mc).

Основная рабочая папка /home/user/src, ее содержимое:

- Bootstrap-v1.11 - исходники и бинарные файлы первого загрузчика (загружается SAM-BA скриптами "Send Boot file").
- buildroot-2010.02 - пакет сборки корневой файловой системы
- linux-2.6.28 - ядро, скрипты сборки внутри
- Ядро 2.6.33.1 собрано для "оценки", драйвера не интегрированы
- rootfs - образец старой корневой ФС
- u-boot 1.1.5 - второй загрузчик (загружается SAM-BA в DataFlash по адресу 0x8000 или в NAND flash по адресу 0x20000)
- utils - набор утилит для "старой" ФС, в контексте buildroot не актуально, но makefile некоторых может быть полезен

В корневом каталоге ядра присутствует два скрипта:

**make\_kernel** – собирает ядро и копирует файл в папку TFTP сервера

**make\_menuconfig** – запускает конфигурационное меню ядра

В корневом каталоге buildroot-2010.02 присутствует скрипт:

**build\_system** – собирает файловую систему и запускает скрипт сборки ядра

Например, необходимо добавить некоторый файл в корневую ФС, для этого:

- копируем файл в buldroot-2010.02/target/(в требуемую директорию)
- запускаем скрипт buldroot-2010.02/build\_system
- включаем/перезагружаем плату с подключенным Ethernet и RS232 кабелями
- прерываем в u-boot процесс загрузки нажатием любой клавиши
- выполняем "run nand\_update" либо "run tftp\_boot"

Например, необходимо обновить ядро Linux, для этого:

- запускаем скрипт linux-2.6.XX/make\_kernel
- включаем/перезагружаем плату с подключенным Ethernet и RS232 кабелями
- прерываем в u-boot процесс загрузки нажатием любой клавиши
- выполняем "run nand\_update" либо "run tftp\_boot"

## 6. Общий принцип работы системы

После подачи питания (перезагрузки), процессор запускает первичный загрузчик (находится во внутренней не перепрограммируемой ROM) и по определенному алгоритму определяет наличие исполняемого кода во внешних носителях. Если приложение не найдено, процессор остается в режиме, который подразумевает взаимодействие с ним по средством утилиты SAM-BA, которая позволяет программировать внешние носители, подключенные к процессору.

Поскольку внешняя SDRAM (или любая другая память не инициализирована), первое запускаемое приложение должно быть загрузчиком т.к. его максимальный размер не может превышать 4К. Это приложение (загрузчик) в первую очередь должен проинициализировать внешнюю память (например, правильно настроить параметры SDRAM), скопировать исполняемое приложение из внешней Flash памяти во внешнюю SDRAM/SRAM память и передать ему управление.

В нашем контексте, первым приложением (размером 4К) является так называемый bootstrap загрузчик (предоставляемый фирмой Atmel и адаптированный под конкретную плату), который инициализирует SDRAM, копирует из NAND приложение и запускает его.

Вторым приложением так же является загрузчик (u-boot) но с уже более обширными возможностями, например, он умеет копировать файлы по TFTP, USB, SD/MMC ... поддерживает целый набор команд и режимов. В переменных окружения u-boot есть команда запуска, в которой указано, по какому адресу NAND flash следует прочитать образ ядра с интегрированной корневой файловой системой, куда этот образ памяти записать и по какому адресу запустить. Следующие сообщения консоли иллюстрируют этот процесс:

```
U-Boot 2010.03 (Apr 21 2010 - 18:56:06)
DRAM: 64 MB
NAND: 256 MiB
DataFlash:AT45DB321
Nb pages: 8192
Page Size: 528
Size= 4325376 bytes
Logical address: 0xC0000000
Area 0: C0000000 to C00041FF (RO) Bootstrap
Area 1: C0004200 to C00083FF Environment
Area 2: C0008400 to C0041FFF (RO) U-Boot
Area 3: C0042000 to C0251FFF Kernel
Area 4: C0252000 to C041FFFF FS
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Net: macb0
macb0: PHY present at 1
macb0: Starting autonegotiation...
macb0: Autonegotiation complete
macb0: link up, 100Mbps full-duplex (lpa: 0xcde1)
Hit any key to stop autoboot: 0

NAND read: device 0 offset 0x60000, size 0x780000
7864320 bytes read: OK
***Переменные окружения u-boot подразумевают возможность хранить команды и некоторые аргументы (например, IP адрес сервера) в специально отведенном блоке flash памяти,сообщение вида "*** Warning - bad CRC, using default environment" говорит о том, что u-boot не находит валидные переменные окружения и использует те что запрограммированы по умолчанию.
```

Перед запуском ядра Linux, оно первым делом проверяет контрольную сумму собственного архива и распаковывает «себя» и корневую ФС, иллюстрация:

```
## Booting kernel from Legacy Image at 20400000 ...
Image Name: Linux Kernel Image
Image Type: ARM Linux Kernel Image (gzip compressed)
```

```
Data Size:      4438174 Bytes =  4.2 MB
Load Address:   20008000
Entry Point:    20008000
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK
```

Starting kernel ...

Далее идет инициализация всей системы, драйверов, файловых систем, после чего управление передается скрипту начального запуска.

## 7. Программирование внешних носителей с помощью SAM-BA

В большинстве случаев, обновить-восстановить систему можно из загрузчика u-boot, но если он или bootstrap поврежден, тогда восстановить систему можно только с помощью утилиты (предоставляемой фирмой Atmel) SAM-BA. К сожалению, с ростом версии этой утилиты, стабильность ее работы не улучшается, скорее наоборот, скрипты работы с платами (в текущих версиях) уже не предоставляются в открытом виде, исключен JTAG интерфейс (в версии 2,9) ...

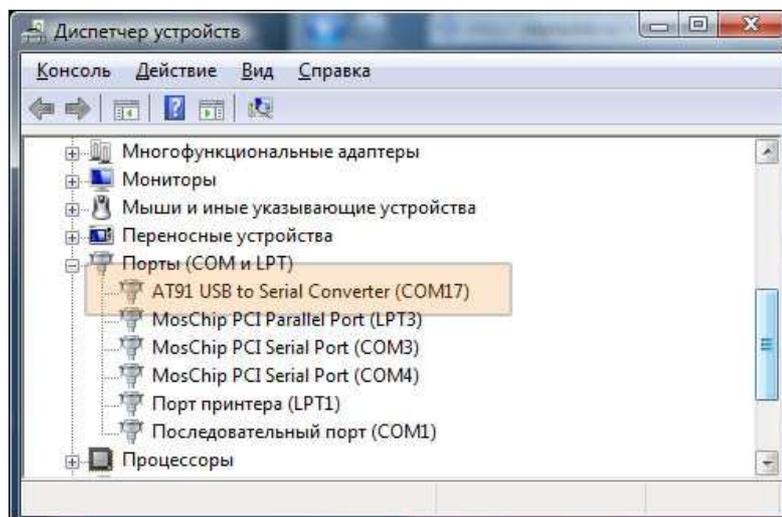
В общем виде, возможно подключение к плате 3-мя интерфейсами USB-device, через COM порт, JTAG (в текущей версии его поддержка отсутствует, присутствовала в более ранних версиях SAM-BA). Самым «надежным» себя зарекомендовал USB интерфейс, обращаю внимание, после установки драйвера (при первом подключении платы) не следует менять порт USB хаба вашего PC для взаимодействия с платой.

Для того, чтобы процессор был способен взаимодействовать с SAM-BA, необходимо, чтобы его первоначальный загрузчик не смог найти исполняемые коды во внешних носителях (в случае с JTAG интерфейсом этого условия не требуется), для этого достаточно разомкнуть перемычки J13, J14 и выключить-включить питание платы. Явным признаком того что процессор остался в первичном загрузчике, является сообщение в терминальной программе – “RomBoot” с последующим бездействием системы. После этого, J13,J14 следует замкнуть.

7.1 Распаковываем sam-ba\_2.9\_cdc\_xp\_vista.rar, инсталляции не требует, скрипты плат уже в архиве.

7.2 Подключаем USB-B кабель. При включении питания, система должна найти новое USB устройство (если ранее эта процедура не выполнялась), присваиваем ему драйвер /drv/atm6124\_cdc.inf .

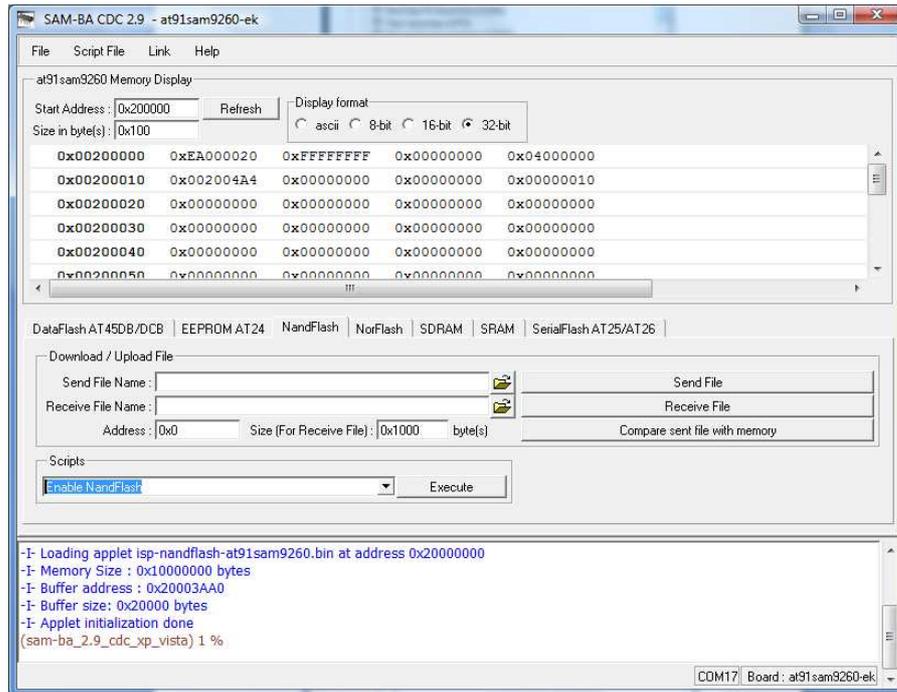
В результате манипуляций, при подключении платы, в диспетчере устройств должен появляться "AT91 USB to Serial Converter COMXX":



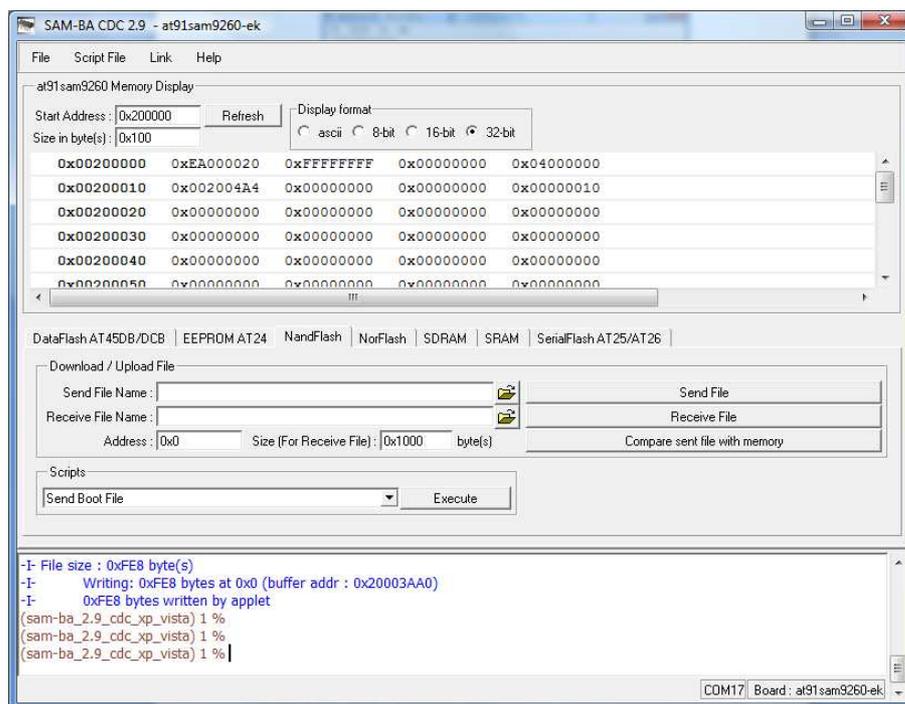
7.3 Запускаем SAM-BA, выбираем COM порт AT91 USB to Serial Converter, выбираем плату at91sam9260-ek (или at91sam9260-sk):



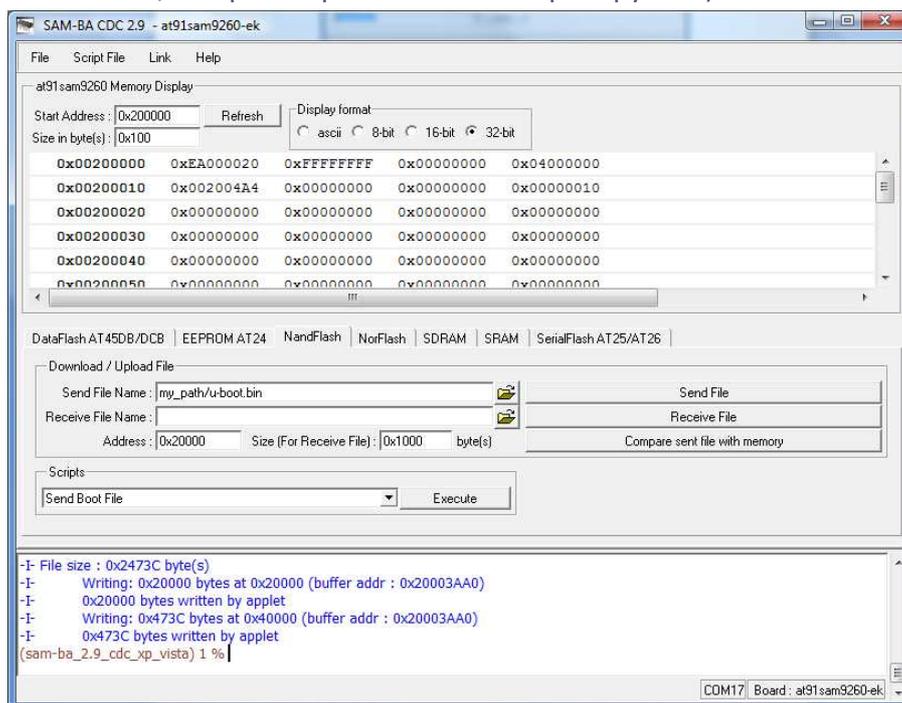
7.4 Выполняем скрипт Enable NandFlash:



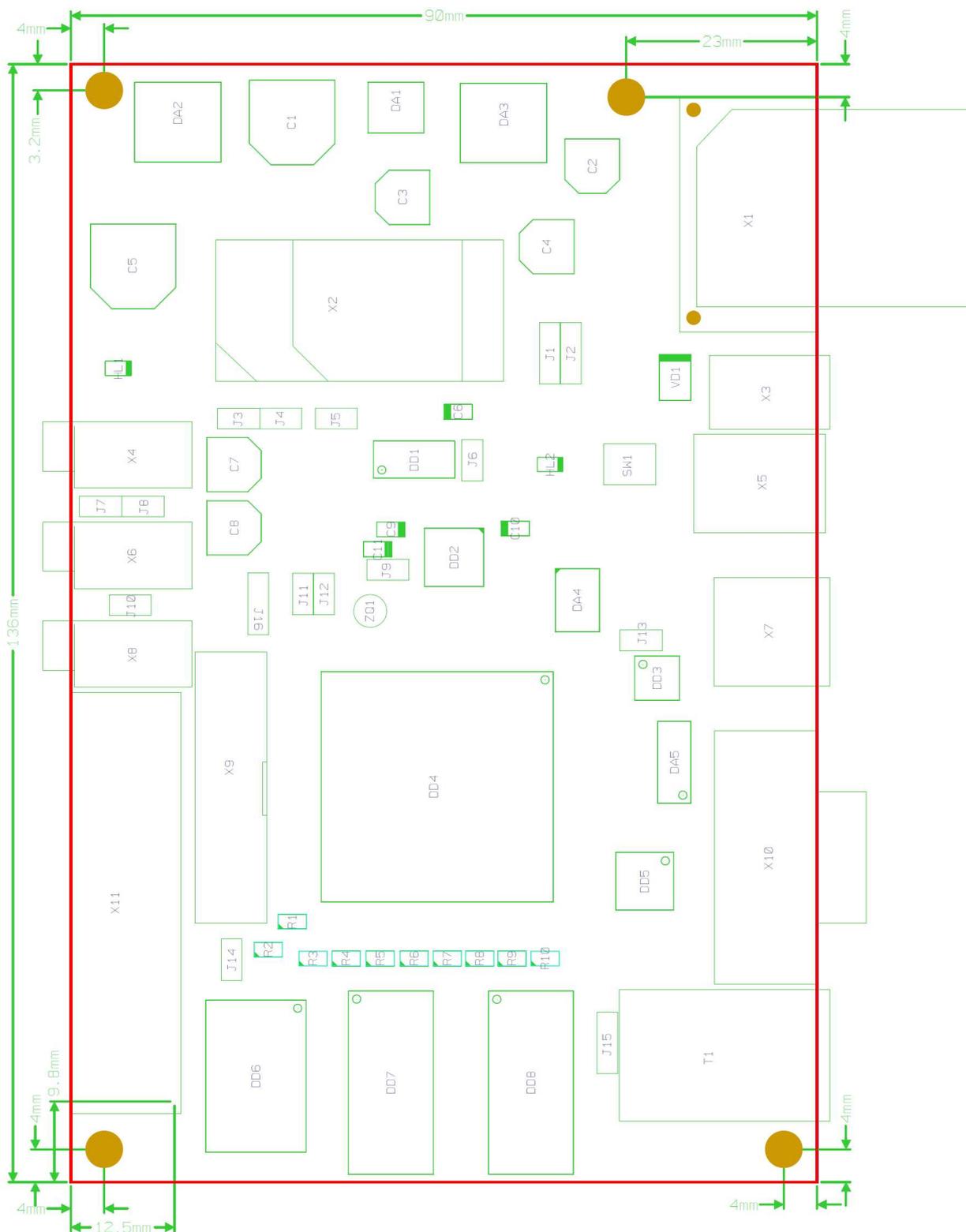
7.5 Выполняем скрипт Send boot File, выбираем файл для загрузки nandflash\_sk\_at91sam9260.bin: xhzz



7.6 Открываем файл u-boot.bin и записываем его по адресу 0x20000 (адрес начала второго блока NAND flash, в первом хранится bootstrap загрузчик):



7.7 Перезагружаемся (лучше выключить-включить питание), останавливаем процесс загрузки u-boot, запускаем виртуальную машину (убедитесь что сетевые интерфейсы настроены правильно, должен проходить PING на адрес 192.168.0.2 – именно этот адрес указан в u-boot как TFTP сервер), в u-boot выполняем “run tftp\_update”.



Габаритный чертеж



