SAM-BA 3 Training

MPU Team
June 26

Objectives

- **Training Objectives**
  - Provide a basic understanding of the SAM-BA 3 tool
  - Become familiar with the SAM-BA command line
  - Use command-line to program a SAMA5D4-Xplained board
  - Use a QML script to program a SAMA5D4-Xplained board
  - Rebuild an existing SAM-BA applet from SoftPack
  - Modify and use SAM-BA QML scripts

Overview

- **SAM-BA : SAM Boot Assistant**

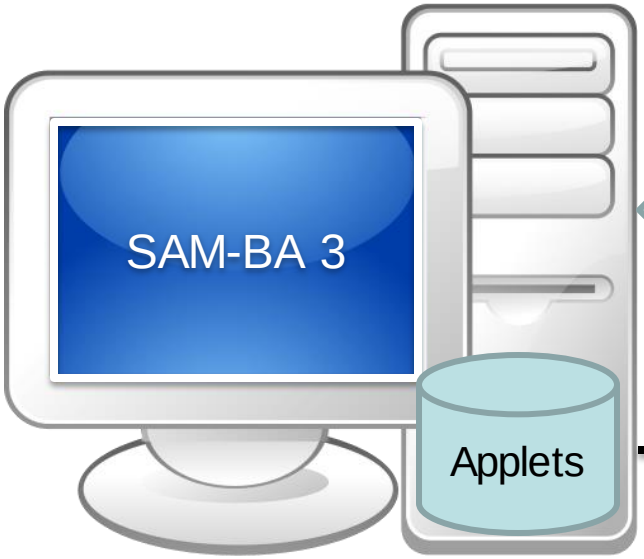
  - In-System Programming Tool for external memories (SPI Flash, QSPI Flash, NAND Flash, eMMC, SD Card, etc.) connected to the chip on Atmel Xplained board or customer board.
  - Communication with the target over USB CDC, UART and JTAG
  - For USB and UART communications, SAM-BA relies on the SAM-BA Monitor program running in the ROM code of all ARM products designed in Rousset
  - Uses small programs called ‘applets’ uploaded into the SRAM of the device and running from there
  - Each applet is dedicated to one type of external memory, and contains the memory read/write operations algorithms

SAM-BA

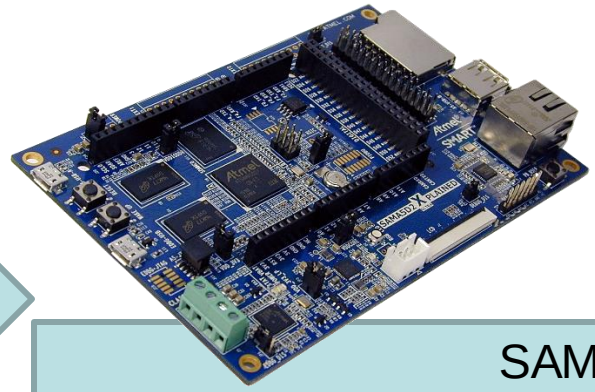
Host

Target

Linux, Windows, MAC
Any ARM-based system



USB, serial, JTAG



SAMA5

SRAM

Applet

The applet is downloaded in the product's SRAM
and executed on the target

Running SAM-BA

- **Command Line interface**
  - Easy, self documented
  - Can be used to script SAM-BA using an external scripting tool (Bash scripts, Windows Batch files, etc.)
  - Supports most simple SAM-BA purposes: erase / write / verify / read

- **QML Scripting**
  - Gives access to all SAM-BA features
  - More complicated than command-line but allows complex scripts to be executed

- **GUI**
  - Not yet implemented

SAM-BA Command Line (1)

- **SAM-BA Tool is self-documented**

sam-ba --help

```
SAM-BA Command Line Tool v3.1.1
Copyright 2015-2016 ATMEL Corporation

Usage: sam-ba [options]

Options:
  -v, --version                          Displays version information.
  -h, --help                             Displays this help.
  -x, --execute <script.qml>             Execute script <script-file>.
  -p, --port <port[:options:...]>        Communicate with device using <port>.
  -d, --device <device>                  Connected device is <device>.
  -b, --board <board>                    Connected board is <board>.
  -m, --monitor <command[:options:...]>  Run monitor command <command>.
  -a, --applet <applet[:options:...]>    Load and initialize applet <applet>.
  -c, --command <command[:args:...]>     Run command <command>.
```

SAM-BA Command Line (2)

- **Choosing a communication port**

```
sam-ba --port help
Known ports: j-link, serial


sam-ba --port serial:help
Syntax:
    serial:[<port>]:[<baudrate>]
Examples:
    serial                serial port (will use first AT91 USB if found otherwise
first serial port)
    serial:COM80          serial port on COM80
    serial:ttyUSB0:57600  serial port on /dev/ttyUSB0, baudrate 57600
```

SAM-BA Command Line (3)

- **Board or Device?**
  - Boards include presets for external memories (controller, pinout, frequency, etc.)

- **Choosing a device/board**

```
sam-ba --device help
Known devices: sama5d2, sama5d4, samv7

sam-ba --board help
Known boards: sama5d2-xplained, sama5d4-xplained
```

SAM-BA Command Line (4)

- **Applet**
  - Each external memory type as its own applet. It contains the memory read/write algorithm
  - The applets available in SAM-BA depend on the device and board available memories

- **Choosing an applet**

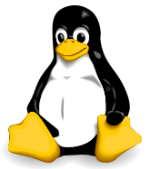
```
sam-ba -d sama5d4 --applet help
Known applets: lowlevel, serialflash, nandflash
```

- **Applet configuration**

```
sam-ba -d sama5d4 -a serialflash:help
Syntax: serialflash:[<instance>]:[<ioset>]:[<npcs>]:[<frequency>]
Parameters:
    instance   SPI controller instance
    ioset      I/O set
    npcs       SPI chip select number
    frequency  SPI clock frequency in MHz
Examples:
    serialflash           use default board settings
    serialflash:0:1:0:66  use fully custom settings (SPI0, IOSET1,
NPCS0, 66Mhz)
    serialflash::::20     use default board settings but force frequency
to 20Mhz
```

SAM-BA Command Line (hands on)

SAMA5D4-Xplained Nandflash Programing



- **Enable access to the ROM Code SAM-BA Monitor (USB communication link)**
  - Close jumper BOOT_DIS : this disables Nandflash chip select signal
  - Connect the board to the computer using A5-USB-A port : this power on the board and enumerates the USB CDC link to the computer
  - Open jumper BOOT_DIS to reconnect the Nandflash chip select signal

- **Setup clocks : runs lowlevel initialization applet**

```
sam-ba -p usb -b sama5d4-xplained -a lowlevel
```

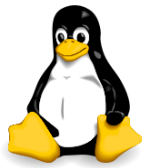
- **Erase the NAND Flash : runs nandflash applet**

```
sam-ba -p usb -b sama5d4-xplained -a nandflash -c erase
```

- **Write the demo to NAND Flash**

```
sam-ba -p usb -b sama5d4-xplained -a nandflash \
-c writeboot:at91bootstrap-sama5d4_xplained.bin \
-c write:u-boot-sama5d4-xplained.bin:0x40000 \
-c write:u-boot-env-sama5d4-xplained.bin:0xC0000 \
-c write:at91-sama5d4_xplained.dtb:0x180000 \
-c write:zImage-sama5d4-xplained.bin:0x200000 \
-c write:atmel-xplained-demo-image-sama5d4-xplained.ubi:0x800000
```

SAM-BA QML Scripts Introduction (1)



- **What is QML?**
  - Scripting language of Qt5
  - Declarative language including Javascript snippets

- **Lots of documentation on Qt website:**
  - **http://doc.qt.io/qt-5/qmlreference.html**
  - **http://doc.qt.io/qt-5/qtqml-index.html**
  - **http://doc.qt.io/qt-5/qml-tutorial.html**

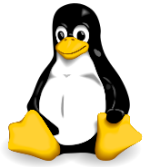
SAM-BA QML Scripts Introduction (2)

- **Minimal QML Script**

```
import QtQuick 2.3
Item {
    Component.onCompleted: {
        print("Hello world!")
    }
}
```

- **Component.onCompleted is called when QML component (script) has finished loading**

SAM-BA QML Scripts Arguments

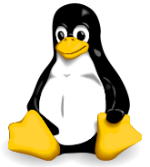


- **Scripts can parse user-defined command-line arguments**

```
import QtQuick 2.3
Item {
    Component.onCompleted: {
        if (Script.arguments.length > 0) {
            for (var i = 0; i < Script.arguments.length; i++)
            print("Arg[" + i + "] -> " + Script.arguments[i])
        } else {
            print("No arguments!");
        }
    }
}
```

See `examples/scripting/arguments.qml` for more information

SAM-BA QML Scripts Return codes



- **Scripts can return user-defined error code**

```
import QtQuick 2.3
Item {
     Component.onCompleted: {
          Script.returnCode = 42
     }
}
```

See `examples/scripting/return-code.qml` for more information

SAM-BA QML Scripts Examples

- **Sample scripts are provided for most devices and external memory types**

```
examples/
├── sama5d2
│   ├── boot-config
│   ├── nandflash
│   ├── qspiflash
│   ├── sdmmc
│   └── serialflash
├── sama5d4
│   ├── ledblink
│   ├── nandflash
│   └── serialflash
└── samv7
    └── flash
```

SAM-BA QML scripts

SAMA5D4-Xplained Serial flash Programing (1)

- **examples/sama5d4/serialflash/serialflash-usb.qml**

```
import SAMBA 3.1
import SAMBA.Connection.Serial 3.1
import SAMBA.Device.SAMA5D4 3.1

AppletLoader {
    connection: SerialConnection { }
    device: SAMA5D4 { board: "sama5d4-xplained" }
    onConnectionOpened: {
        appletInitialize("lowlevel")
        appletInitialize("serialflash")
        appletErase(0, connection.applet.memorySize)
        appletWrite(0x00000, "application.bin", true)
    }
}
```

Uses helper QML class AppletLoader to load and run applet

SAM-BA QML scripts

SAMA5D4-Xplained Serial flash Programing (2)

- **Parameters can be customized**
  - Default serial connection (autodetect USB port)

```
connection: SerialConnection { }
```

  - Serial connection on COM23

```
connection: SerialConnection { port: "COM23" }
```

  - SPI settings from board definition

```
device: SAMA5D4 { board: "sama5d4-xplained" }
```

  - Custom SPI settings

```
device: SAMA5D4 {
    config {
        spiInstance: 0
        spiIoset: 1
        spiChipSelect: 0
        spiFreq: 66
    }
}
```

SAM-BA QML scripts (hands on)

SAMA5D4-Xplained Nandflash Programing (1)

- **Enable access to the ROM Code SAM-BA Monitor (USB communication link)**
  - Close jumper BOOT_DIS : this disables Nandflash chip select signal
  - Connect the board to the computer using A5-USB-A port : this power on the board and enumerates the USB CDC link to the computer (SAM-BA Monitor)
  - Open jumper BOOT_DIS to reconnect the Nandflash chip select signal

- **Look at and run the following script**

  `examples/sama5d4/nandflash/nandflash-usb.qml`

  - **`sam-ba -x examples/sama5d4/nandflash/nandflash-usb.qml`**

SAM-BA applets

- ‘Applet’ is the name for a small embedded program that runs into the chip internal SRAM
- An applet is sent to the target using SAM-BA Monitor ‘Send Data’ command, which writes a bunch of data at a specific address into the chip
- At the top of the applet binary, a reserved area called Mailbox is used to pass parameters to the applet
- After sending the applet in the SRAM, SAM-BA fills this mailbox with the command to be executed by the applet, and any parameter that can by useful or necessary for this command
- Then SAM-BA sends a SAM-BA Monitor ‘Go’ command at the address where the applet is loaded in SRAM, and then the applet is executed and perform the required command

SAM-BA applets

- **Applet source code in included in SAMA5 softpack:**
  https://github.com/atmelcorp/atmel-software-package

- **Applets can be rebuilt using GCC**

```
cd samba_applets/serialflash
make TARGET=sama5d4-generic RELEASE=1
```

- **Or using IAR:**

```
cd samba_applets/serialflash
make TARGET=sama5d4-generic RELEASE=1 iar
```

  - Then open workspace applet-serialflash_sama5d4-generic.eww in IAR Workbench

SAM-BA applets (hands on)

- **Goal: Add new commands to an existing applet to power ON and power OFF the USER LED on the SAMA5D4-Xplained board**
  - **Find the sources for the lowlevel applet**
  - **In the main.c file, add two new commands and their associated functions. One to power ON the USER LED, and the other one to power OFF this LED. (tip: look at the getting_started example in the SoftPack to find code related to LED control)**
  - **Rebuild the lowlevel applet for SAMA5D4, and copy the resulting binary in the SAMA5D4 applets folder in SAM-BA**
  - **In SAM-BA**
    - **Find the QML file where the two commands have to be added in order to make the USER LED toggle with command lines like:**
      - `sam-ba -p serial -b sama5d4-xplained -a lowlevel -c user_led_on`
      - `sam-ba -p serial -b sama5d4-xplained -a lowlevel -c user_led_off`

Thank You!