AVR32 UC3 Software Framework

User Manual

Section 1

AVR32 UC3 Software Framework

1.1 Features

- Drivers for each AVR®32 UC3 peripheral
- Software libraries optimized for AVR32
- Hardware components drivers
- Demo applications that use all libraries
- RTOS-ready source code
- Complete software framework in C code
- Full projects compatible with GNU GCC and IAR Embedded Workbench®
- High level user documentation including examples, getting started and tutorials
- Designed to run on AVR32 UC3 evaluation kits and easily portable to any other hardware platform
- Designed to help develop software applications for AVR32 UC3 microcontroller

1.2 Introduction

This document describes the software framework developed for the Atmel AVR32 UC3 microcontrollers.

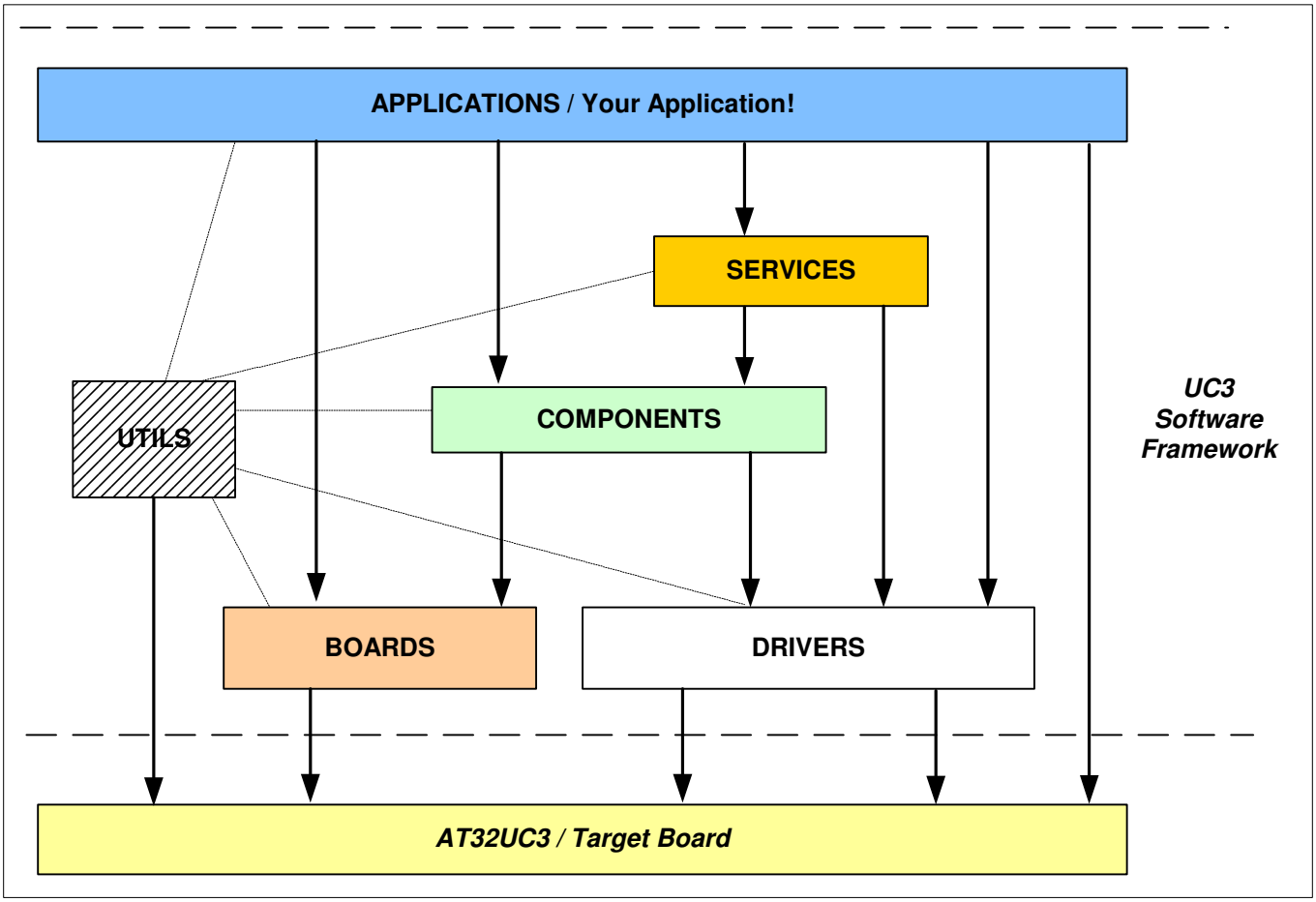
This framework provides software drivers and libraries to build any application for AVR32 UC3 devices.

This framework has been carefully developed to help develop and glue together the different components of a software design. It also has been designed to be easily integrable into an operating system (OS) as well as to operate in a stand-alone way.

1.3 Framework Description

The framework is divided into several modules. Each module is provided with **full source code**, **example of usage, rich HTML documentation** and ready to use project for the **IAR® EWAVR32** and the **GNU GCC** compilers.

Figure 1. Block Diagram Overview



1.3.1 UC3 Drivers (directory /DRIVERS)

This directory contains software drivers such as ADC, GPIO or Timer peripherals.

Each driver is composed of a driver.c and driver.h file that provides low level functions to access the peripheral.

1.3.2 Software Services (directory /SERVICES)

This directory provides application-oriented piece of software such as a USB mass storage class, a FAT file system and an optimized DSP library.

1.3.3 Hardware Components (directory /COMPONENTS)

This directory provides software drivers to access hardware components such as external memory (e.g. Atmel Dataflash® or SDRAM) or LCD.

1.3.4 C/C++ Utilities (directory /UTILS)

This directory provides several linker script files and C/C++ files with general usage defines, macros and functions.

1.3.5 Demo Applications (directory /APPLICATIONS)

This directory provides application examples that are based on services, components and drivers modules.

1.4 Framework Usage

1.4.1 Installation

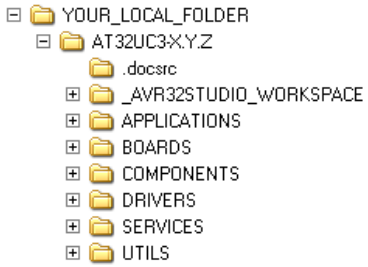
- Download the AT32UC3X-SoftwareFramework-x.y.z.zip.
- Extract it to a local directory.
- Open AVR32_Readme.html file in the root directory. The framework contains rich html documentation for all directories and files.
- Navigate through the documentation. Check the installation section first.

Note: There is a uc3softwareframework-releasenote.txt file in the root directory. This file contains all information regarding the revision management of the software framework.

1.4.2 Directory Architecture

Below is an example on what the complete AVR32 UC3 software framework folder should look like after a successful and complete installation.

Figure 2. AVR32 UC3 Software Framework Architecture



Here is the content of the ADC driver directory to introduce the typical organization of a software driver.

Figure 3. Module File Architecture

```
ADC/                                    : the name of the controller is usually used as the directory name.
|-- adc.c                               : the implementation of the software driver API.
|-- adc.h                               : the software driver API.
|
|-- EXAMPLE                             : the EXAMPLE directory holds one example of usage of this software driver API.
|   |
|   |-- AT32UC3A                        : this directory holds all AT32UC3A family build files.
|   |   |-- GCC                         : this directory contains gcc-specific files.
|   |   |   |-- Makefile                : gcc make makefile.
|   |   |   |-- uc3a0512-adc_example.elf  : target downloadable elf file. Use avr32program.
|   |   |   |-- config.mk               : make configuration file(included by the Makefile file,
|   |   |   |                             contains the list of source files and compilation/link options.
|   |   |   `-- gdb_cmdfile.txt         : configuration file for the avr32-gdb program
|   |   |                                 (Usage: avr32-gdb -x gdb_cmdfile.txt or ddd -debugger avr32-gdb --command=gdb_cmdfile.txt
|   |   |
|   |   `-- IAR                         : this directory holds IAR-specific files.
|   |       |-- adc_example.ewd         : IAR EWAVR32 debug configuration file.
|   |       |-- adc_example.ewp         : IAR EWAVR32 project file.
|   |       `-- adc_example.eww         : IAR EWAVR32 workspace file.
|   |
|   |-- DOC                             : this directory holds all automatically-generated documentation.
|   |   |-- ....html
|   |   |-- ...
|   |   |-- ...
|   |   `...
|   |
|   |-- adc.jpg                         : a image that will be integrated in the automatically generated documentation.
|   |-- adc_example.c                   : the implementation of the example.
|   |-- documentation.h                 : the entry point source of doxygen documentation.
|   |-- doxyfile.doxygen                : the doxygen configuration file
|   `-- readme.html                     : the entry point of all automatically-generated documentation.
`.
5 directories, 50 files
```

1.4.3 Building an Application with the Framework

Step1. Defining the application

Let's take an example to build an application: the Atmel AT45DB dataflash is a memory that is present on EVK1100 evaluation kit. It is connected to the SPI interface of the AVR32 UC3A microcontroller. We want to access this memory and perform a basic sanity check.

Step2. Identifying the blocks

- AT45DB: We are going to use the AT45DB driver, it is a hardware component: it will be under the /COMPONENTS directory. It is a memory component: it will be stored under the /COMPONENTS/MEMORY directory. It is a dataflash memory type. It will be located under the /COMPONENTS/MEMORY/DATA_FLASH directory. It is an AT45DBX type of dataflash. The driver will be located under the /COMPONENTS/MEMORY/DATA_FLASH/AT45DBX directory.
- Power manager: to start the external oscillator clock. It is a AVR32 UC3 peripheral. The power manager driver is located under the /DRIVERS/PM/ directory.
- RS232: the USART will be used as an output for the memory test. It is located under the /DRIVERS/USART directory.
- GPIO: we will need to configure the UC3 IO to be used by the peripherals. It is located under the /DRIVERS/GPIO directory.
- Debug: we will need format printing functions to print messages to the RS232. The debug module is located under the /UTILS/DEBUG directory.
- The application itself will be located under the /APPLICATIONS/MYAPPLICATION.

Step3. Building the project

We need to create a new project for this application. In order not to start from scratch, we are going to reuse a simple project. Copy the /DRIVERS/GPIO/EXAMPLE content to the /APPLICATIONS/MYAPPLICATION directory.

Note: The /DRIVERS/USART/EXAMPLE is also a simple ready-to-use project that you can start from.

- Rename the gpio_example.c file to myapplication.c file.
- Implement the application as defined in step1.
- Modify the GCC or IAR project to match this new application.

Note: You will find one or several of examples in each module:

1. /DRIVERS/PM/EXAMPLE1 and /DRIVERS/PM/EXAMPLE2
2. /DRIVERS/USART/USART_EXAMPLE and /DRIVERS/USART/MODEM_EXAMPLE
3. /DRIVERS/GPIO/EXAMPLE

Note: The /COMPONENTS/MEMORY/DATA_FLASH/AT45DBX/EXAMPLE directory contains an example of this application.

1.5 Tools

1.5.0.1 IAR

All IAR projects are compatible with the latest available version of IAR Embedded Workbench for AVR32.

1.5.0.2 GNU GCC and AVR32Studio

All GNU GCC projects are provided with stand alone makefile.

1.5.0.3 Evaluation Kit

The projects are designed to run on AVR32 UC3 evaluation kits. All the projects contain board definition files to ease portability to any other hardware platform that use a AVR32 UC3 microcontroller.

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 436-4270
FAX 1(408) 436-4314

Microcontrollers
Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 436-4270
FAX 1(408) 436-4314

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Atmel Smart Card ICs
Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Atmel Heilbronn
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom
Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

---

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

7732A–AVR32–05/07 /xM