SMART ARM-based Microcontrollers

AT03260: SAM D/R System (SYSTEM) Driver

APPLICATION NOTE

Introduction

This driver for Atmel® | SMART ARM®-based microcontrollers provides an interface for the configuration and management of the device's system relation functionality, necessary for the basic device operation. This is not limited to a single peripheral, but extends across multiple hardware peripherals.

The following peripherals are used by this module:

- SYSCTRL (System Control)
- PM (Power Manager)

The following devices can use this module:

- Atmel | SMART SAM D20/D21
- Atmel | SMART SAM R21
- Atmel | SMART SAM D09/D10/D11
- Atmel | SMART SAM DA1

The outline of this documentation is as follows:

- Prerequisites
- Module Overview
- Special Considerations
- Extra Information
- Examples
- API Overview

Table of Contents

1. Software License

2. Prerequisites

There are no prerequisites for this module.

3. Module Overview

The System driver provides a collection of interfaces between the user application logic, and the core device functionality (such as clocks, reset cause determination, etc.) that is required for all applications. It contains a number of sub-modules that control one specific aspect of the device:

- System Core (this module)
- System Clock Control (sub-module)
- System Interrupt Control (sub-module)
- System Pin Multiplexer Control (sub-module)

3.1. Voltage References

The various analog modules within the SAM devices (such as AC, ADC, and DAC) require a voltage reference to be configured to act as a reference point for comparisons and conversions.

The SAM devices contain multiple references, including an internal temperature sensor and a fixed band-gap voltage source. When enabled, the associated voltage reference can be selected within the desired peripheral where applicable.

3.2. System Reset Cause

In some applications there may be a need to execute a different program flow based on how the device was reset. For example, if the cause of reset was the Watchdog timer (WDT), this might indicate an error in the application, and a form of error handling or error logging might be needed.

For this reason, an API is provided to retrieve the cause of the last system reset, so that appropriate action can be taken.

3.3. Sleep Modes

The SAM devices have several sleep modes. The sleep mode controls which clock systems on the device will remain enabled or disabled when the device enters a low power sleep mode. Table 3-1 SAM Device Sleep Modes on page 5 lists the clock settings of the different sleep modes.

Table 3-1. SAM Device Sleep Modes

Sleep mode	CPU clock	AHB clock	APB clocks	Clock sources	System clock	32KHz	Reg mode	RAM mode
Idle 0	Stop	Run	Run	Run	Run	Run	Normal	Normal
Idle 1	Stop	Stop	Run	Run	Run	Run	Normal	Normal
Idle 2	Stop	Stop	Stop	Run	Run	Run	Normal	Normal
Standby	Stop	Stop	Stop	Stop	Stop	Stop	Low Power	Source/ Drain biasing

Before entering device sleep, one of the available sleep modes must be set. The device will automatically wake up in response to an interrupt being generated or upon any other sleep mode exit condition.

Some peripheral clocks will remain enabled during sleep, depending on their configuration. If desired, the modules can remain clocked during sleep to allow them continue to operate while other parts of the system are powered down to save power.

4. Special Considerations

Most of the functions in this driver have device specific restrictions and caveats; refer to your device datasheet.

5. Extra Information

For extra information, see Extra Information for SYSTEM Driver. This includes:

- Acronyms
- Dependencies
- Errata
- Module History

6. Examples

For SYSTEM module related examples, refer to the sub-modules listed in the Module Overview.

7. API Overview

7.1. Function Definitions

7.1.1. Voltage References

7.1.1.1. Function system_voltage_reference_enable()

Enable the selected voltage reference.

```
void system_voltage_reference_enable(
        const enum system_voltage_reference vref)
```

Enables the selected voltage reference source, making the voltage reference available on a pin as well as an input source to the analog peripherals.

Table 7-1. Parameters

Data direction	Parameter name	Description
[in]	vref	Voltage reference to enable

7.1.1.2. Function system_voltage_reference_disable()

Disable the selected voltage reference.

```
void system_voltage_reference_disable(
        const enum system_voltage_reference vref)
```

Disables the selected voltage reference source.

Table 7-2. Parameters

Data direction	Parameter name	Description
[in]	vref	Voltage reference to disable

7.1.2. Device Sleep Control

7.1.2.1. Function system_set_sleepmode()

Set the sleep mode of the device.

```
enum status_code system_set_sleepmode(
        const enum system_sleepmode sleep_mode)
```

Sets the sleep mode of the device; the configured sleep mode will be entered upon the next call of the system_sleep() function.

For an overview of which systems are disabled in sleep for the different sleep modes, see Sleep Modes.

Table 7-3. Parameters

Data direction	Parameter name	Description
[in]	sleep_mode	Sleep mode to configure for the next sleep operation

Table 7-4. Return Values

Return value	Description
STATUS_OK	Operation completed successfully
STATUS_ERR_INVALID_ARG	The requested sleep mode was invalid or not available

7.1.2.2. Function system_sleep()

Put the system to sleep waiting for interrupt.

```
void system_sleep( void )
```

Executes a device DSB (Data Synchronization Barrier) instruction to ensure all ongoing memory accesses have completed, then a WFI (Wait For Interrupt) instruction to place the device into the sleep mode specified by system_set_sleepmode until woken by an interrupt.

7.1.3. Reset Control

7.1.3.1. Function system_reset()

Reset the MCU.

```
void system_reset( void )
```

Resets the MCU and all associated peripherals and registers, except RTC, all 32KHz sources, WDT (if ALWAYSON is set) and GCLK (if WRTLOCK is set).

7.1.3.2. Function system_get_reset_cause()

Return the reset cause.

```
enum system_reset_cause system_get_reset_cause( void )
```

Retrieves the cause of the last system reset.

Returns
An enum value indicating the cause of the last system reset.

7.1.4. System Debugger

7.1.4.1. Function system_is_debugger_present()

Check if debugger is present.

```
bool system_is_debugger_present( void )
```

Check if debugger is connected to the onboard debug system (DAP).

Returns
A bool identifying if a debugger is present.

Table 7-5. Return Values

Return value	Description
true	Debugger is connected to the system
false	Debugger is not connected to the system

7.1.5. System Identification

7.1.5.1. Function system_get_device_id()

Retrieve the device identification signature.

```
uint32_t system_get_device_id( void )
```

Retrieves the signature of the current device.

Returns
Device ID signature as a 32-bit integer.

7.1.6. System Initialization

7.1.6.1. Function system_init()

Initialize system.

```
void system_init( void )
```

This function will call the various initialization functions within the system namespace. If a given optional system module is not available, the associated call will effectively be a NOP (No Operation).

Currently the following initialization functions are supported:

- System clock initialization (via the SYSTEM CLOCK sub-module)
- Board hardware initialization (via the Board module)
- Event system driver initialization (via the EVSYS module)
- External Interrupt driver initialization (via the EXTINT module)

7.2. Enumeration Definitions

7.2.1. Enum system_reset_cause

List of possible reset causes of the system.

Table 7-6. Members

Enum value	Description
SYSTEM_RESET_CAUSE_SOFTWARE	The system was last reset by a software reset
SYSTEM_RESET_CAUSE_WDT	The system was last reset by the watchdog timer
SYSTEM_RESET_CAUSE_EXTERNAL_RESET	The system was last reset because the external reset line was pulled low

Enum value	Description
SYSTEM_RESET_CAUSE_BOD33	The system was last reset by the BOD33
SYSTEM_RESET_CAUSE_BOD12	The system was last reset by the BOD12
SYSTEM_RESET_CAUSE_POR	The system was last reset by the POR (Power on reset)

7.2.2. Enum system_sleepmode

List of available sleep modes in the device. A table of clocks available in different sleep modes can be found in Sleep Modes.

Table 7-7. Members

Enum value	Description
SYSTEM_SLEEPMODE_IDLE_0	IDLE 0 sleep mode
SYSTEM_SLEEPMODE_IDLE_1	IDLE 1 sleep mode
SYSTEM_SLEEPMODE_IDLE_2	IDLE 2 sleep mode
SYSTEM_SLEEPMODE_STANDBY	Standby sleep mode

7.2.3. Enum system_voltage_reference

List of available voltage references (VREF) that may be used within the device.

Table 7-8. Members

Enum value	Description
SYSTEM_VOLTAGE_REFERENCE_TEMPSENSE	Temperature sensor voltage reference
SYSTEM_VOLTAGE_REFERENCE_BANDGAP	Bandgap voltage reference

8. Extra Information for SYSTEM Driver

8.1. Acronyms

Below is a table listing the acronyms used in this module, along with their intended meanings.

Acronym	Definition
PM	Power Manager
SYSCTRL	System control interface

8.2. Dependencies

This driver has the following dependencies:

- None

8.3. Errata

There are no errata related to this driver.

8.4. Module History

An overview of the module history is presented in the table below, with details on the enhancements and fixes made to the module since its first release. The current version of this corresponds to the newest version in the table.

Changelog
Added new `system_reset()` to reset the complete MCU with some exceptions
Added new `system_get_device_id()` function to retrieved the device ID
Initial Release

9. Document Revision History

Doc. Rev.	Date	Comments
42120E	12/2015	Added support for SAM DA1 and SAM D09
42120D	12/2014	Added support for SAM R21 and SAM D10/D11
42120C	01/2014	Added support for SAM D21
42120B	06/2013	Corrected documentation typos
42120A	06/2013	Initial document release

Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (“Safety-Critical Applications”) without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.