

---

**Atmel AT03260: SAM D20 System Driver (SYSTEM)**

---

**ASF PROGRAMMERS MANUAL**

## SAM D20 System Driver (SYSTEM)

---

This driver for SAM D20 devices 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 outline of this documentation is as follows:

- [Prerequisites](#)
- [Module Overview](#)
- [Special Considerations](#)
- [Extra Information](#)
- [Examples](#)
- [API Overview](#)

## Table of Contents

SAM D20 System Driver (SYSTEM) .....	1
Software License .....	3
1. Prerequisites .....	4
2. Module Overview .....	5
2.1. Voltage References .....	5
2.2. System Reset Cause .....	5
2.3. Sleep Modes .....	5
3. Special Considerations .....	6
4. Extra Information .....	7
5. Examples .....	8
6. API Overview .....	9
6.1. Function Definitions .....	9
6.1.1. System identification .....	9
6.1.2. Voltage references .....	9
6.1.3. Device sleep .....	9
6.1.4. Reset control .....	10
6.1.5. System initialization .....	11
6.2. Enumeration Definitions .....	11
6.2.1. Enum system_reset_cause .....	11
6.2.2. Enum system_sleepmode .....	11
6.2.3. Enum system_voltage_reference .....	11
7. Extra Information for SYSTEM Driver .....	13
7.1. Acronyms .....	13
7.2. Dependencies .....	13
7.3. Errata .....	13
7.4. Module History .....	13
Index .....	14
Document Revision History .....	15

## Software License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of Atmel may not be used to endorse or promote products derived from this software without specific prior written permission.
4. This software may only be redistributed and used in connection with an Atmel microcontroller product.

THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1. Prerequisites

There are no prerequisites for this module.

## 2. 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)

### 2.1 Voltage References

The various analog modules within the SAM D20 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 D20 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.

### 2.2 System Reset Cause

In some application 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.

### 2.3 Sleep Modes

The SAM D20 devices have several sleep modes, where 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 2-1: SAM D20 Device Sleep Modes on page 5](#) lists the clock settings of the different sleep modes.

**Table 2-1. SAM D20 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

To enter device sleep, one of the available sleep modes must be set, and the function to enter sleep called. The device will automatically wake up in response to an interrupt being generated or other device event.

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

### 3. **Special Considerations**

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

## 4. Extra Information

For extra information see [Extra Information for SYSTEM Driver](#). This includes:

- [Acronyms](#)
- [Dependencies](#)
- [Errata](#)
- [Module History](#)

## 5. Examples

For SYSTEM module related examples, please refer to the sub-modules listed in the [system module overview](#).



## 6. API Overview

### 6.1 Function Definitions

#### 6.1.1 System identification

##### 6.1.1.1 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.

#### 6.1.2 Voltage references

##### 6.1.2.1 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 6-1. Parameters**

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

##### 6.1.2.2 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 6-2. Parameters**

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

#### 6.1.3 Device sleep

##### 6.1.3.1 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 6-3. Parameters**

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

**Table 6-4. Return Values**

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

### 6.1.3.2 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.

## 6.1.4 Reset control

### 6.1.4.1 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 ALWAYS\_ON is set) and GCLK (if WRTLOCK is set).

### 6.1.4.2 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.

## 6.1.5 System initialization

### 6.1.5.1 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)

## 6.2 Enumeration Definitions

### 6.2.1 Enum `system_reset_cause`

List of possible reset causes of the system.

**Table 6-5. 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.
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).

### 6.2.2 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 6-6. 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.

### 6.2.3 Enum `system_voltage_reference`

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

**Table 6-7. Members**

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

## 7. Extra Information for SYSTEM Driver

### 7.1 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

### 7.2 Dependencies

This driver has the following dependencies:

- None

### 7.3 Errata

There are no errata related to this driver.

### 7.4 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 <a href="#">system_reset()</a> to reset the complete MCU with some exceptions
Added new <a href="#">system_get_device_id()</a> function to retrieved the device ID.
Initial Release

## Index

### E

#### Enumeration Definitions

- [system\\_reset\\_cause](#), 11
- [system\\_sleepmode](#), 11
- [system\\_voltage\\_reference](#), 11

### F

#### Function Definitions

- [system\\_get\\_device\\_id](#), 9
- [system\\_get\\_reset\\_cause](#), 10
- [system\\_init](#), 11
- [system\\_reset](#), 10
- [system\\_set\\_sleepmode](#), 9
- [system\\_sleep](#), 10
- [system\\_voltage\\_reference\\_disable](#), 9
- [system\\_voltage\\_reference\\_enable](#), 9

## Document Revision History

Doc. Rev.	Date	Comments
B	06/2013	Corrected documentation typos.
A	06/2013	Initial release



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

© 2013 Atmel Corporation. All rights reserved. / Rev.: 42120B-SAMD20-11/2013

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

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.