AVR1003: Using the XMEGA™ Clock System

Features

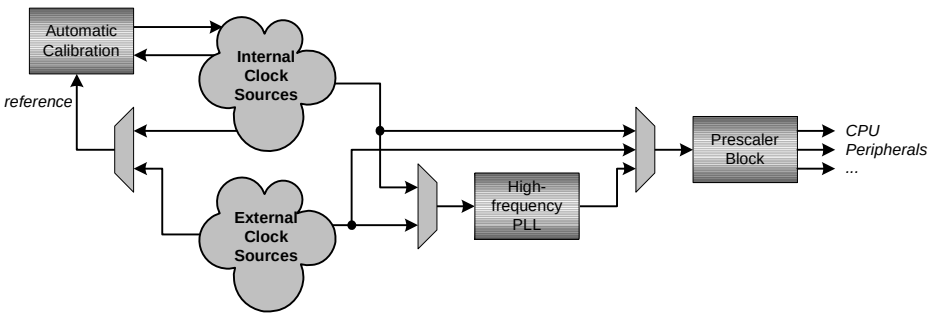
- **Internal 32 kHz, 2 MHz, and 32 MHz oscillators**
- **External crystal oscillator or clock input**
- **Internal PLL with multiplication factor 1x to 31x**
- **Safe clock source switching**
- **External oscillator failure detection**
- **1x to 2048x system clock prescaler option**
- **Automatic runtime calibration of internal oscillators**
- **Driver source code included**

1 Introduction

The XMEGA Clock System is a set of highly flexible modules that provides a large portfolio of internal and external clock sources. An internal high-frequency PLL and a flexible prescaler block provide a vast amount of possible clock source configurations, both for the CPU and peripherals. An external oscillator failure detector and optional automatic runtime calibration of internal oscillators reduce external component count and help designing for accuracy and safety.

This application describes the XMEGA Clock System with detailed configuration procedures and a ready-to-use software driver.

Figure 1-1. XMEGA Clock System Overview



8-bit AVR® Microcontrollers

Application Note

Rev. 8072E-AVR-11/09

2 Clock System Overview

The XMEGA Clock System provides a large portfolio of clock sources, both internal and external. In addition, an internal PLL can be used to multiply selected clock sources with a factor ranging from 1x to 31x.

In order to ease implementations, the default clock setting for the XMEGA is to start up running from an internal 2 MHz factory-calibrated source. In this way, if the default settings are sufficient, no external components or software configuration is required to start executing code.

The following sections describe the various clock sources and the available configuration options for using one of them as a main system clock.

2.1 Internal Clock Sources

There are five internal clock sources (including the internal PLL), ranging from an ultra low-power 32 kHz RC oscillator to a 32 MHz factory-calibrated ring oscillator with auto-calibration features. All but one can be used for the main system clock.

Any number of the internal sources can be enabled at any given time, even if none are used for the main system clock. Also, some clock sources might even be used for multiple purposes, such as the 32 kHz RC oscillator that can be used as a main system clock and as a clock source for the Real-time Counter module at the same time.

Some of the internal clock source can be used as a reference to the internal PLL in order to generate even higher frequencies. The PLL is covered in Section 2.3.

2.1.1 Ultra Low-power 32 kHz RC Oscillator

The Ultra Low-power 32 kHz internal RC oscillator (ULP32K) is mainly used for system purposes, such as startup delays, the Watchdog Timer and various internal timings. It can be used for the Real-time Clock module, but is not available as a system clock source. With a frequency accuracy of 30%, it is not intended as such either.

The ULP oscillator can be used as a source for the Real-time Counter module. Refer to the application note “AVR1314: Using the XMEGA Real-time Counter” for more details.

The ULP oscillator is automatically enabled by hardware when it is needed.

2.1.2 Calibrated 32 kHz RC Oscillator

The 32.768 kHz internal RC oscillator (RC32K) is factory-calibrated to 32 kHz with an accuracy of 1% at 3V and 25°C. The calibration value is stored in the calibration row and is automatically loaded into the oscillator’s calibration register (RC32KCAL) on reset. This value is read and write accessible for the user, but the oscillator should not be tuned outside recommended limits. Please refer to the electrical characteristics in the device datasheet for details.

The RC32K oscillator can be used as a system clock source directly and also as a source for the Real-time Counter module (actually the RC32K divided down to 1.024 kHz). Refer to the application note “AVR1314: Using the XMEGA Real-time Counter” for more details.

The RC32K oscillator cannot be used as a source for the internal PLL.

Apart from being a system clock source alternative, the RC32K can also be used as a reference for the auto-calibration feature of the 2 MHz and 32 MHz internal oscillators. The two oscillators and the auto-calibration feature are covered in later sections.

The *Internal 32kHz RC Oscillator Enable* bit (RC32KEN) in the *Oscillator Control* register (OSC.CTRL) controls this oscillator, while the *Internal 32kHz RC Oscillator Ready* bit (RC32KRDY) in the *Oscillator Status* register (OSC.STATUS) can be polled to check if it is stable and ready to be used as a system clock source.

2.1.3 Calibrated 2 MHz RC Oscillator

The 2 MHz internal RC oscillator (RC2M) is factory-calibrated to 2 MHz with an accuracy of 1% at 3V and 25°C. The calibration value is stored in the calibration row and is automatically loaded into the oscillator's internal calibration register on reset. The oscillator can be further tuned and calibrated using the auto-calibration feature covered in Section 2.8.

The RC2M oscillator can be used as a system clock source directly or through the internal PLL to generate even higher system frequencies. Note that when using the PLL, the user is responsible for not exceeding recommended frequency limits for the CPU and peripherals.

The *Internal 2MHz RC Oscillator Enable* bit (RC2MEN) in the *Oscillator Control* register (OSC.CTRL) controls this oscillator, while the *Internal 2MHz RC Oscillator Ready* bit (RC2MRDY) in the *Oscillator Status* register (OSC.STATUS) can be polled to check if it is stable and ready to be used as a system clock source.

2.1.4 Calibrated 32 MHz Ring Oscillator

The 32 MHz internal ring oscillator (R32M) is factory-calibrated to 32 MHz with an accuracy of 1% at 3V and 25°C. The calibration value is stored in the calibration row and is automatically loaded into the oscillator's internal calibration register on reset. The oscillator can be further tuned and calibrated using the auto-calibration feature covered in Section 2.8.

The R32M oscillator can be used as a system clock source directly or through the internal PLL to generate even higher system frequencies. Note that when using the PLL, the user is responsible for not exceeding recommended frequency limits for the CPU and peripherals.

The *Internal 32MHz Ring Oscillator Enable* bit (R32MEN) in the *Oscillator Control* register (OSC.CTRL) controls this oscillator, while the *Internal 32MHz Ring Oscillator Ready* bit (R32MRDY) in the *Oscillator Status* register (OSC.STATUS) can be polled to check if it is stable and ready to be used as a system clock source.

2.2 External Clock Sources

There are several possible external clock sources, all sharing the XTAL1 and XTAL2 pins. Of course, this means that only one source can be enabled at any given time. Hardware makes sure that no two external sources are enabled simultaneously. Attempts to do so will be disregarded by the hardware.

An exception from this is the Real-time Counter Oscillator using the TOSC1 and TOSC2 pins with an external watch crystal as a source to the Real-time Clock module. The Real-time Counter Oscillator can be used as a system clock source

directly, but not the 1 kHz prescaled signal as this only goes to the RTC. Refer to the application note “AVR1314: Using the XMEGA Real-time Counter” for more details on the RTC.

2.2.1 Crystals and Resonators

The XTAL1 and XTAL2 pins are input and output, respectively, of an inverting amplifier that serves as an on-chip oscillator using external crystals or resonators from 0.4 MHz to 16 MHz. This Crystal Oscillator (XOSC) can be configured for one the following options:

- 32.768 kHz watch crystal oscillator with a low-power option
- 0.4 – 2 MHz low-swing crystal or resonator oscillator
- 2 – 9 MHz low-swing crystal or resonator oscillator
- 9 – 12 MHz low-swing crystal or resonator oscillator
- 12 – 16 MHz low-swing crystal or resonator oscillator

Actually, this is implemented as two crystal oscillators, one for 32.768 kHz crystals and one for high-frequency crystals.

When using external 32 kHz watch crystals, the *32kHz Crystal Oscillator Low-power Mode* bit (X32KLPM) in the *External Oscillator Control* register (OSC.XOSCCTRL) enables low-power mode. This only applies to 32 kHz watch crystals and should only be used with high quality crystals.

The *External Oscillator Selection* bitfield (XOSCSEL) in the *External Oscillator Control* register (OSC.XOSCCTRL) selects the external oscillator type and start-up time.

The *External Oscillator Enable* bit (XOSCEN) in the *Oscillator Control* register (OSC.CTRL) controls this oscillator, while the *External Oscillator Ready* bit (XOSCRDY) in the *Oscillator Status* register (OSC.STATUS) can be polled to check if it is stable and ready to be used as a system clock source.

The oscillator must be configured before enabling it and the configuration cannot be changed without disabling it first. Hardware will disregard any attempts to change configuration while it is enabled.

How to connect external crystals and the recommended capacitor values are shown in Figure 2-1 and

Table 2-1 below.

Figure 2-1. Crystal Oscillator Connection

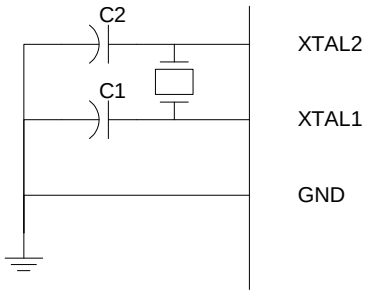


Table 2-1. Recommended Capacitor Values

Frequency Range	Capacitor Value (C1 and C2)
0.4 – 2 MHz	100 pF
2 – 9 MHz	15 pF
9 – 12 MHz	15 pF
12 – 16 MHz	10 pF

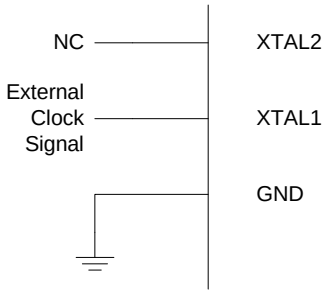
Note that the crystal oscillator is a low-swing type oscillator, which means that the XTAL pins cannot be used to provide a clock signal to other devices. Instead, one of the Timer/Counter modules could be used to generate a clock signal for external devices. Refer to the application note “AVR1306: Using the XMEGA Timer/Counters”.

2.2.2 External Clock

If the hardware design provides an external clock signal, this can be connected to the XTAL1 pin and the on-chip crystal oscillator can be disabled. External clock is one of the possible selections in the *External Oscillator Selection* bitfield (XOSCSEL) in the *External Oscillator Control* register (OSC.XOSCCTRL) and always use a startup time of six clock cycles before being reported to be stable with the *External Oscillator Ready* bit (XOSCRDY) in the *Oscillator Status* register (OSC.STATUS).

Figure 2-2 below shows how to connect an external clock signal. To ensure correct operation, the clock frequency should not change more than 2% from one cycle to another. If the clock frequency needs to change more rapidly, the XMEGA should be kept in a reset state during the change.

Figure 2-2. External Clock Connection



2.3 High-frequency PLL Option

If higher frequencies are required, a built-in PLL can be used to multiply a clock source by a factor from 1 to 31. The following clock sources can be used as input to the PLL:

- Internal 2 MHz RC oscillator
- Internal 32 MHz ring oscillator divided internally by four
- External 0.4 – 16 MHz crystal oscillator
- External clock

The hardware will not allow using the external oscillator as a PLL source if it is configured for 32 kHz operation. If the external clock is used as input, it must be

above 0.4 MHz. Please refer to the electrical characteristics for accurate minimum values.

The correct procedure for enabling the PLL is as follows:

1. Select multiplication factor using the *Multiplication Factor* bitfield (PLLFAC) and input clock source using the *Clock Source* bitfield (PLLSRC) in the *PLL Control Register* (OSC.PLLCTRL).
2. Enable the PLL using the *PLL Enable* bit (PLLEN) in the *Oscillator Control* register (OSC.CTRL).
3. Wait for the PLL to stabilize by checking the *PLL Ready* bit (PLLRDY) in the *Oscillator Status* register (OSC.STATUS).

The user should make sure that the input clock source is stable before trying to enable the PLL. When enabled, the PLL typically needs 64 reference cycles to stabilize.

The PLL configuration cannot be changed without disabling it first. Hardware will disregard any attempts to change the configuration while it is enabled.

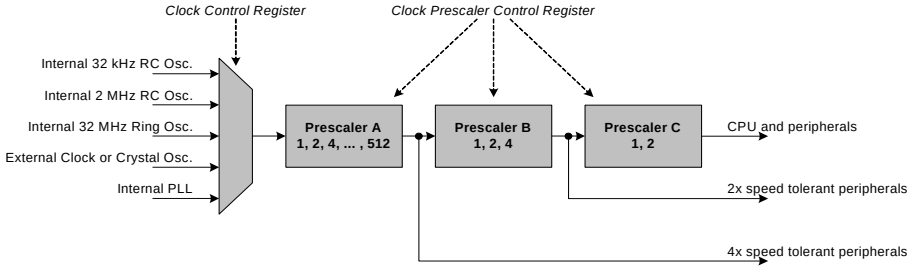
The user is responsible for providing a minimum input clock frequency of 0.44 MHz to the PLL and make sure that the output never exceeds 200 MHz. The minimum input frequency requirement applies when using an external clock signal as input to the PLL.

2.4 System Clock Selection and Prescalers

To provide flexible options for the main system clock, the XMEGA Clock System contains a MUX that selects one of the clock sources or the PLL. It feeds the selected clock through a highly configurable prescaler block that can divide the clock signal by a factor from 1 to 2048 before it is routed to the CPU and peripherals. It can also provide clock signals two and four times the CPU clock frequency for peripherals than operate at a higher frequency than the CPU, for instance the EBI module that can operate at twice the CPU frequency.

Figure 2-3 below illustrates the MUX and prescalers. The output from the MUX goes through three prescaler stages. The first stage, prescaler A, can divide by a factor of 1 to 512. Then prescalers B and C can be configured to either pass the signal through or divide by up to four combined. See datasheet for details on this restriction. Prescaler C output is routed to the CPU and peripherals.

Figure 2-3. System Clock Selection and Prescalers



The default setup after a reset is to select the internal 2 MHz RC oscillator and pass the clock signal undivided through all three prescalers. Prescaler A can be used to

lower the system frequency by a large factor in order to reduce power consumption, while prescaler B and C is primarily intended to provide clock signals one, two, or four times the CPU frequency to selected peripherals.

The prescaler settings can be changed safely at any time. The hardware will ensure a glitch-free transition between frequencies. When changing clock sources, hardware will prevent any attempts to change to an unstable clock source.

The procedure for changing system clock and prescalers is as follows:

1. Load the *Protect IO Register* signature (byte value `0xD8`) into the *Configuration Change Protection* register (`CCP`). This will automatically disable all interrupts for the next four CPU instruction cycles.
2. Set the desired configuration for the prescaler or the system clock.

Note that writing the signature to the CCP register only leaves time for reconfiguring either the prescaler or the system clock. Repeat the CCP write if both needs reconfiguration. Study the example software for details.

2.5 Clock Configuration Locking

As a safety precaution, it is possible to lock the current Clock System configuration until the next reset. If your application is going to run from one clock configuration, it could be wise to protect that configuration from accidental changes.

The procedure for locking the configuration is as follows:

1. Set up the XMEGA Clock System to desired configuration
2. Load the *Protect IO Register* signature (byte value `0xD8`) into the *Configuration Change Protection* register (`CCP`). This will automatically disable all interrupts for the next four CPU instruction cycles.
3. Set the *Clock System Lock* bit (`LOCK`) in the *Clock System Lock* register (`CLK.LOCK`) to logic one.
4. The clock configuration is now locked until the next system reset.

If the External Oscillator Failure Detector is enabled, a failure will unlock the Clock System configuration. See Section 2.7 below for details.

2.6 Sleep Modes and Clock Sources

All oscillators and clock generation will be disabled in Power-down and Power-save sleep modes. The exception is if the Real-time Counter is enabled in Power-save mode, which leaves the RTC clock source running. If the RTC is not used, Power-down mode is recommended instead of Power-save mode. With these two sleep modes, start-up delays behave as if the oscillators were disabled.

In Standby and Extended Standby sleep modes, the oscillators are kept running, enabling the CPU to wake up without extensive delays.

In Idle mode, the oscillators are kept running and the CPU will wake up from sleep without any delay.

Please refer to the device datasheet for details on sleep modes and wakeup conditions.

2.7 External Oscillator Failure Detector

When using external crystals or external clock sources, there is always a slight probability of the source failing. As a safety precaution, the XMEGA Clock System has an External Oscillator Failure Detector that monitors the external clock source and reacts if it stops.

If the external source is selected as the main system clock source and it fails, the failure detector switches to the internal 2 MHz RC oscillator and issues a Non-maskable Interrupt (NMI). Please refer to the application note “AVR1305: XMEGA Interrupts and the Programmable Multilevel Interrupt Controller” for more details on NMIs.

If the external source is enabled, but is not selected as the main system clock source, the external source will only be disabled, but the NMI will still be issued.

The failure detector uses the ULP oscillator to monitor the external source, and reacts if the frequency falls below 32 kHz. Note that the ULP accuracy is 30%, so it could be possible to run below 32 kHz, but it is not recommended. 32 kHz operation is guaranteed not to cause a failure detection.

The failure detector is not enabled by default. To avoid accidental enabling, a special sequence is required to enable it. The procedure is as follows:

1. Load the *Protect IO Register* signature (byte value `0xD8`) into the *Configuration Change Protection* register (`CCP`). This will automatically disable all interrupts for the next four CPU instruction cycles.
2. Set the *Failure Detection Enable* bit (`XOSCFDEN`) in the *External Oscillator Failure Detection* register (`OSC.XOSCFAIL`) to logic one.
3. The failure detector is now enabled until the next system reset.

Since the failure detector issues a Non-maskable Interrupt if the external source fails when used as the main system clock source, it is important to implement an interrupt handler for this interrupt. Non-maskable interrupts cannot be disabled in any way, hence the name “Non-maskable”. Please study the example software for details.

2.8 Automatic Runtime Calibration of Internal Oscillators

The XMEGA Clock System provides two Digital Frequency-locked Loops (DFLLs), one for the 2 MHz RC oscillator and one for the 32 MHz ring oscillator. The DFLLs can be configured individually to use either the internal 32 kHz RC oscillator or an external 32 kHz watch crystal as a reference for the calibration process.

Once enabled, a DFLL provides continuous calibration of its oscillator based on the clock reference. When entering sleep mode, the current state is frozen and the calibration loop continues from where it stopped when exiting from sleep mode again.

If a DFLL is disabled, the current calibration value for the oscillator will remain in effect until the DFLL is enabled again and the calibration process continues.

The *DFLL Control* register (`OSC.DFLLCTRL`) contains two control bits, *DFLL 32MHz Calibration Reference Selection* (`R32MCREF`) and *DFLL 2MHz Calibration Reference Selection* (`RC2MCREF`), which select the clock reference for the DFLLs. A logic zero selects the internal 32 kHz RC oscillator, while a logic one selects the external watch crystal oscillator. Hardware will disregard any attempts to use an unstable clock source.

The DFLLs themselves require a three-byte counter reference value in the corresponding *DFLL Counter* registers (OSC.DFLLx.OSCCNTn). Correct counter reference values are loaded automatically on reset and should not be changed by the user.

The DFLLs also have a two-byte calibration register each (OSC.DFLLx.CALn), which contains an enable bit and a calibration value, which is loaded automatically by hardware on reset. The calibration value serves as a starting point for the calibration loop and should not be changed by the user.

The correct procedure for enabling a DFLL is as follows:

1. Select internal or external clock reference with the DFLL's control bit in the *DFLL Control* register (OSC.DFLLCTRL).
2. Set the *DFLL Enable* bit (ENABLE) in the DFLL's *Control Register* (DFLLx.CTRL) to logic one to enable the DFLL.
3. The calibration process starts immediately.

There are no runtime calibration options for the internal 32 kHz RC oscillator, but if required, it can be fine-tuned by changing its calibration register (RC32KCAL). However, great care should be taken not to tune any oscillator outside recommended operating limits.

3 Examples

This application note includes a source code package with a basic Clock System driver implemented in C. It is written for the IAR Embedded Workbench® compiler.

Note that this Clock System driver is not intended for use with high-performance code. It is designed as a library to get started with the XMEGA Clock System. For timing and code space critical application development, you should access the Clock System registers directly. Please refer to the driver source code and device datasheet for more details.

3.1 Files

The source code package consists of three files:

- *clksys_driver.c* – Clock System driver source file
- *clksys_driver.h* – Clock System driver header file
- *main.c* – Example code using the driver

For a complete overview of the available driver interface functions and their use, please refer to the source code documentation.

3.2 Doxygen Documentation

All source code is prepared for automatic documentation generation using Doxygen. Doxygen is a tool for generating documentation from source code by analyzing the source code and using special keywords. For more details about Doxygen please visit http://www.doxygen.org. Precompiled Doxygen documentation is also supplied with the source code accompanying this application note, available from the *readme.html* file in the source code folder.

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

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

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

8072E-AVR-11/09