APPLICATION NOTE

AT11480: Analog Comparator Application Examples

ATSAMD21J18A

Introduction

This application note explains the Analog Comparator concepts and its implementation in SAM D microcontrollers with the following application examples:

1. Level crossing detector.
2. Window mode operation.
3. Preventing false spike detection.
4. Gray signal frequency measurement.
5. SleepWalking with analog comparator.

For demonstration purpose a SAM D21 Xplained Pro board is used.

Table of Contents

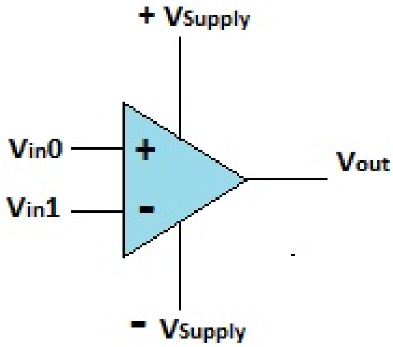
1 Introduction to Analog Comparator

Analog Comparator (AC) is a module that compares two analog input voltages and outputs a signal level indicating which of the inputs is greater or lesser. An analog comparator is basically an amplifier without feedback and thus has very high gain.

Figure 1-1 shows the basic symbol of an analog comparator. Typically an analog comparator compares voltage levels on two inputs and gives digital output based on the comparison. When the voltage on the positive input (Vin0) is greater than the voltage on the negative input (Vin1) then the output voltage (V_{OUT}) is saturated to its positive supply ($+V_{SUPPLY}$), otherwise the output is saturated to is negative supply ($-V_{SUPPLY}$).

In microcontrollers, since there is no negative supply voltage, GND (ground level) is taken as $–V_{SUPPLY}$ and V_{CC} level is taken as $+V_{SUPPLY}$.

Figure 1-1. Analog Comparator Equivalent Diagram

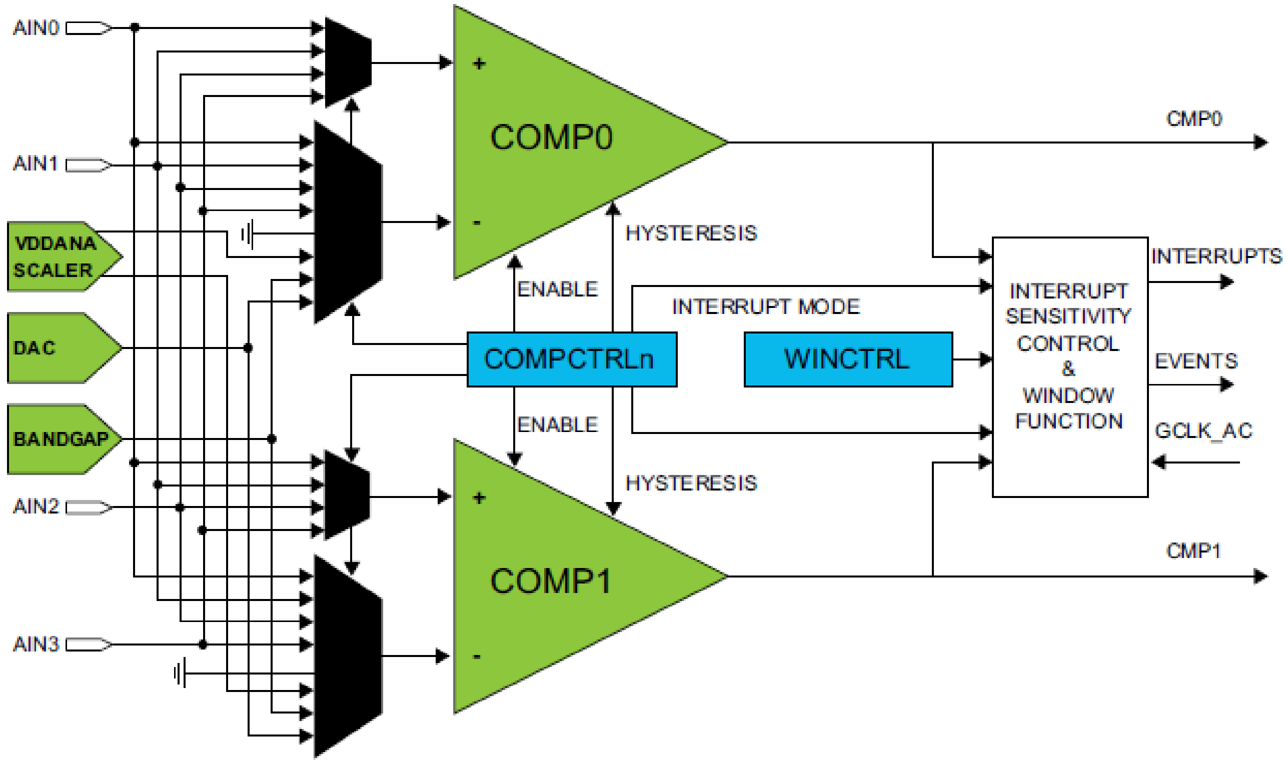


2 Analog Comparator in SAM D Devices

The Analog Comparator module in SAM D microcontrollers implements two individual comparators. Each comparator compares the voltage levels on two inputs and provides digital output based on the comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change.

Figure 2-1 shows the block diagram of the analog comparator in SAM D microcontrollers. Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of analog input pins or internal inputs, such as a bandgap reference voltage. The digital output from the comparator is one when the difference between the positive and the negative voltage is positive and zero otherwise.

Figure 2-1. SAM D21 Analog Comparator Block Diagram



The comparators are always grouped in pairs on each port. The AC module may implement one pair. These are called Comparator 0 (COMP0) and Comparator 1 (COMP1). They have identical behaviors, but separate control registers. Each comparator has one positive and one negative input.

2.2 Initialization and Basic Operation

2.2.1 Analog Comparator Register Summary

The analog comparator has two control registers; CTRLA and CTRLB. CTRLA has the enable bit to enable the module, run in standby bit to make the analog comparator functioning in standby sleep mode, software reset bit to reset all the comparator registers, and low power MUX bit that controls the input impedance of the analog input multiplexer. CTRLB has the start bits for comparator 0 and 1, which is used to trigger manual comparison in individual comparators.

The module has an event control register (EVCTRL) that controls the input and output events. COMPEO0 and COMPEO1 bits control output events for two individual comparators and COMPEI0 and COMPEI1 bits control input events for the two individual comparators. Events can be configured such that comparator can act as event generator that outputs event to event system and can act as event user performing a comparison on an input event. Individual comparators, apart from operating in normal mode, can take up a common input forming a window mode operation. The event control register also supports window mode output event.

Three registers, interrupt enable set (INTENSET), interrupt enable clear (INTENCLR) and interrupt flag (INTFLAG) control the interrupts of the analog comparator module. COMP0 and COMP1 bits control individual comparator output interrupts and WIN0 bit controls window mode interrupt.

Three status registers, STATUSA, STATUSB, and STATUSC show the current status of the comparator outputs. STATUSA has two bits that show the current status of the individual comparators and one bit for window mode status. STATUSB has two bits for individual comparators that indicates whether the comparator's output is ready or not and one bit indicating synchronization busy status.

STATUSC is a copy of STATUSA (see STATUSA register), with the additional feature of automatically starting single-shot comparisons. A read of STATUSC will start a comparison on all comparators currently configured for single-shot operation. The read will stall the bus until all enabled comparators are ready. If a comparator is already busy with a comparison, the read will stall until the completion of current comparison and a new comparison will not be started.

The window control register (WINCTRL) has an enable bit to enable the window mode configuration and has two other bits that chooses a window mode option.

Two comparator control registers (COMPCTRL[0]/[1]) are available one for each comparator. This register has control bits like comparator enable, single shot mode enable, propagation delay setting, interrupt selection, positive and negative input selection, output mode selection, hysteresis mode enable, and filter mode enable for individual comparator.

There are two voltage scalers with configurable voltage outputs available for individual comparator. The voltage output level is set in voltage scaler registers (SCALER[0]/[1]).

2.2.2 Analog Comparator Initialization

Before enabling the AC, the input and output events must be configured in the Event Control register (EVCTRL). These settings cannot be changed while the AC is enabled.

Each individual comparator must also be configured by its respective Comparator Control register before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

The AC is enabled by writing a one to the Enable bit in the Control A register (CTRLA.ENABLE). The individual comparators must also be enabled by writing a one to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The AC is disabled by writing a zero to CTRLA.ENABLE. This will also disable the individual comparators, but will not clear their COMPCTRLx.ENABLE bits.

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and it is specified in the Electrical Characteristics section of the datasheet.

During the start-up time, the COMP output is not available. If the supply voltage is below 2.5V, the start-up time is also dependent on the internal voltage doubler.

The AC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC, except DEBUG, will be reset to their initial state, and the AC will be disabled.

Analog comparator module has selectable hysteresis and propagation delay, the hysteresis and propagation delay are two important properties of the comparator’s dynamic behavior. Both parameters can be adjusted to achieve the optimal operation for each application.

2.2.3 Voltage Doubler

The AC contains a voltage doubler that can reduce the resistance of the analog multiplexers when the supply voltage is below 2.5V. The voltage doubler is normally switched on/off automatically based on the supply level. When enabling the comparators, additional start-up time is required for the voltage doubler to settle. If the supply voltage is guaranteed to be above 2.5V, the voltage doubler can be disabled by writing the Low-Power MUX bit in the Control A register (CTRLA.LPMUX) to one. Disabling the voltage doubler saves power and reduces the start-up time.

2.2.4 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the single bit in the Comparator x Control register (COMPCTRLx.SINGLE).

- Continuous measurement
- Single-shot

Continuous Measurement

Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit of Status A register (STATUSA.STATEx). After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

Single-Shot

Single-shot operation is selected by writing COMPCTRLx.SINGLE to one. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing a one to the respective Start Comparison bit in the write only Control B register (CTRLB.STARTx). The comparator is enabled, and after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed. Writing a one to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

2.2.5 Selecting Comparator Inputs

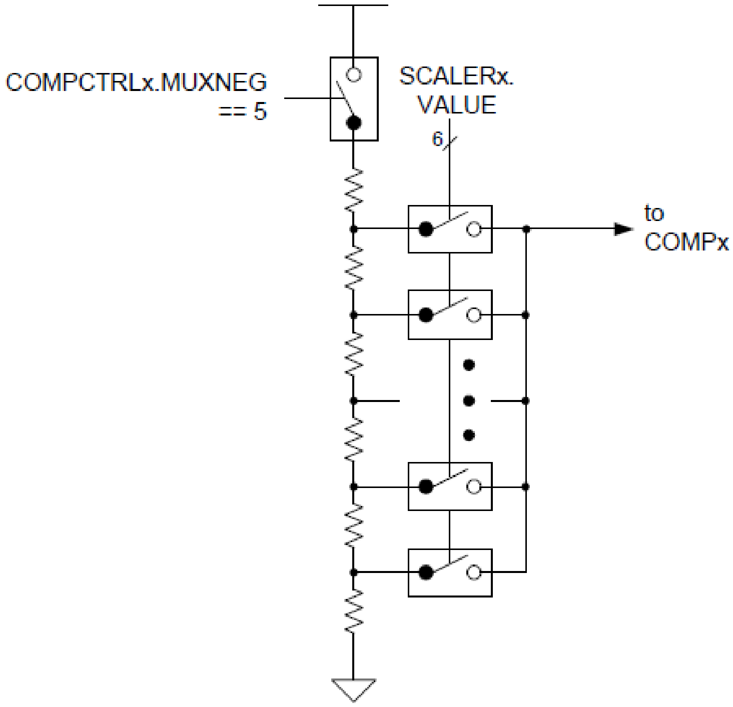
The input selection includes four shared analog port pins and several internal signals. The positive input is fed from an external input pin (AINx).The negative input can be fed either from an external input pin (AINx) or from one of the several internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS)
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG)

2.2.6 V_{DDANA} Scaler

The VDDANA scaler generates a reference voltage that is a fraction of the device’s supply voltage, with 64 levels. One independent voltage channel is dedicated for each comparator. The scalar is enabled when a comparator’s Negative Input MUX bit group in its Comparator Control register (COMPCTRLx.MUXNEG) is set to five and the comparator is enabled. The voltage of each channel is selected by the value bit group in the Scaler x registers (SCALERx.VALUES [5:0]).

Figure 2-2. V_{DDANA} **Scaler**



2.2.7 Comparator Output

Each comparator output state can also be output on a pin for use by external devices. The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from zero to one (rising edge), when the output changes from one to zero (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the single-shot mode to chain further events in the system, regardless of the state of the comparator outputs.

2.2.8 Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help to prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other. This feature is explained in Section 3.4 Preventing False Spike Detection.

2.2.9 Propagation Delay

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest possible consumption. The speed setting is configured for each comparator individually by the speed bit group in the Comparator x Control register (COMPCTRLx.SPEED). The speed bits select the amount of bias current provided to the comparator and as such will also affect the startup time.

3 Application Examples

This chapter explains five different applications demonstrating the AC module in SAM D microcontrollers. The applications have been added to the Atmel® Software Framework (ASF) and can be opened from New Example Project wizard in Atmel Studio.

This section assumes that the user has foreknowledge on programming/debugging a SAM D21 device using Atmel Studio IDE.

Following are the hardware requirements to test the AC applications.

- SAM D21 Xplained Pro board
- Function generator to apply analog input signal
- Oscilloscope to verify the outputs

The applications have been tested with Atmel Studio 6.2 SP2 with ASF version 3.22.0 and above.

3.1 Getting Started

The AC example project has to be opened from New Example Project option in Atmel Studio. Using the option *Atmel Studio → File → New → Example Project* will open the *New Example Project from ASF* or *Extensions* window. The project named *Analog Comparator Examples – SAM D21 Xplained Pro* has to be opened.

```
#define AC_MODE                AC_LEVEL_CROSS_DETECTOR
//#define AC_MODE              AC_WINDOW_DETECTOR
//#define AC_MODE              AC_FALSE_SPIKE_PREVENTER
//#define AC_MODE              AC_SLEEP_WALKING
```

The project has to be built with required application macro uncommented and other macros commented out. For example, the above definition selects the level cross detector application.

After selecting the required application the project can be built with option Atmel Studio → Build → Build Solution.

Once the project is built the SAM D21 Xplained Pro board can be connected and programmed by launching the debug session using option Atmel Studio → Debug → Continue. This option will program the target device and runs the application.

Details on how to provide inputs and how to verify output are given in the respective application sections below.

3.2 Level Crossing Detector

This application demonstrates a simple level crossing detector using a single comparator in the AC module. The comparator output is directly routed to a port pin (PA12) by enabling the AC alternate function for that pin.

This application is selected by uncommenting the following line in main.c and commenting out other application selection macros.

```
#define AC_MODE      AC_LEVEL_CROSS_DETECTOR
```

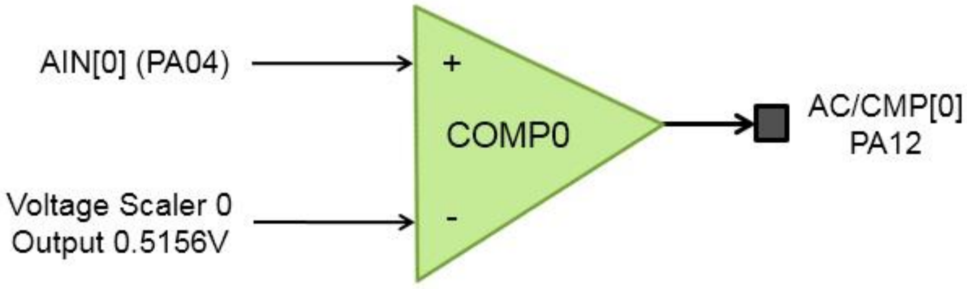
Following are the steps done to configure the AC as level crossing detector.

1. Enable alternate function H (AC/CMP[0]) for pin PA12 so that the comparator 0 output is directly routed to the pin.
2. Select GCLK Generator 0 as GCLK source for AC module. The GCLK Generator 0 is clocked from internal OSC8M oscillator whose output is set to 8MHz.

3. Select AIN[0] pin (PA04) as positive input and internal voltage scaler output as negative input for comparator 0. The voltage scaler register is set to a value of 9 corresponding to a voltage output of 0.5156V.
4. Set comparator 0 in continuous comparison mode with hysteresis and filter mode disabled.
5. Enable comparator 0 output to be routed to I/O pin and select interrupt mode as toggle so that comparator output toggles when positive input crosses the negative input (both during rising and falling).
6. Enable comparator 0 and then enable the AC module.

Figure 3-1 shows the analog comparator connections.

Figure 3-1. Analog Comparator Connections – Level Cross Detector

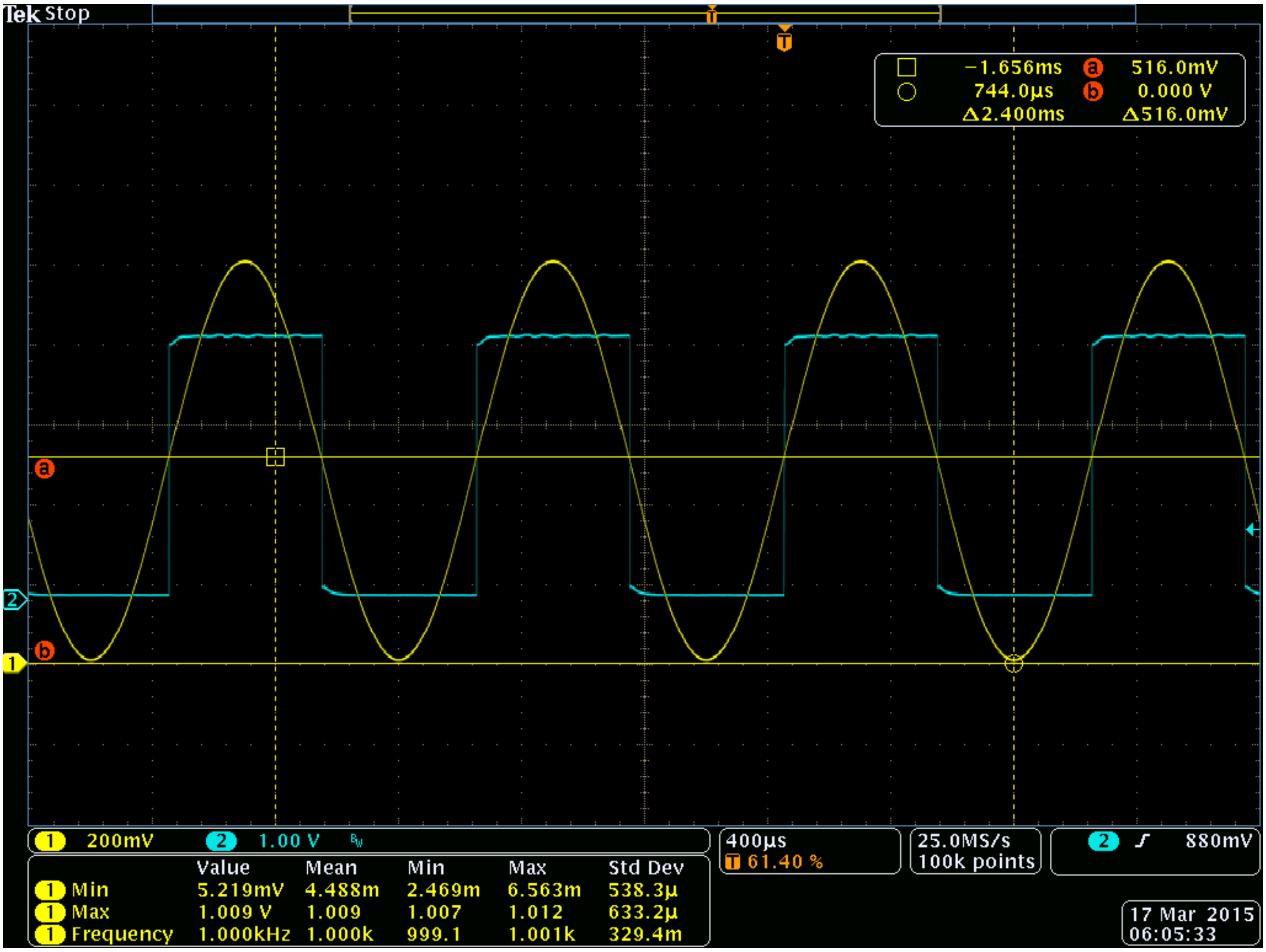


The output is verified by taking a sine wave of frequency in few tens of Hz and signal level of ~1.0312V peak-to-peak with an offset voltage of 0.5156V from a function generator and applying it to pin 17 (PA04) in EXT1 header of SAM D21 Xplained Pro board. This application is theoretically a zero cross detector but since the SAM D microcontroller doesn't accept negative voltages, the input sine wave is offset to be completely in the positive region.

The output is taken from port pin PA12 (pin 7 on EXT3 header in SAM D21 Xplained Pro board). The signal on both PA04 pin and PA12 pin are captured in an oscilloscope for easier comparison.

Figure 3-2 shows the scope shot of the level cross detector output.

Figure 3-2. Level Cross Detector – Output Scope Shot



	Value	Mean	Min	Max	Std Dev
1 Min	5.219mV	4.488m	2.469m	6.563m	538.3µ
1 Max	1.009 V	1.009	1.007	1.012	633.2µ
1 Frequency	1.000kHz	1.000k	999.1	1.001k	329.4m

Channel 1 (Yellow): Input sine wave signal applied on PA04.

Channel 2 (Blue): AC output taken from pin PA12.

Cursor a: AC detect threshold (516mV).

Cursor b: Ground.

3.3 Analog Comparator Window Mode

This application demonstrates the window mode operation using both the comparators in the AC module. The port pin (PA12) is not directly toggled by comparator output, instead it is manually toggled by application inside the window mode interrupt handler.

This application is selected by uncommenting the following line in main.c and commenting out other application selection macros.

```
#define AC_MODE      AC_WINDOW_DETECTOR
```

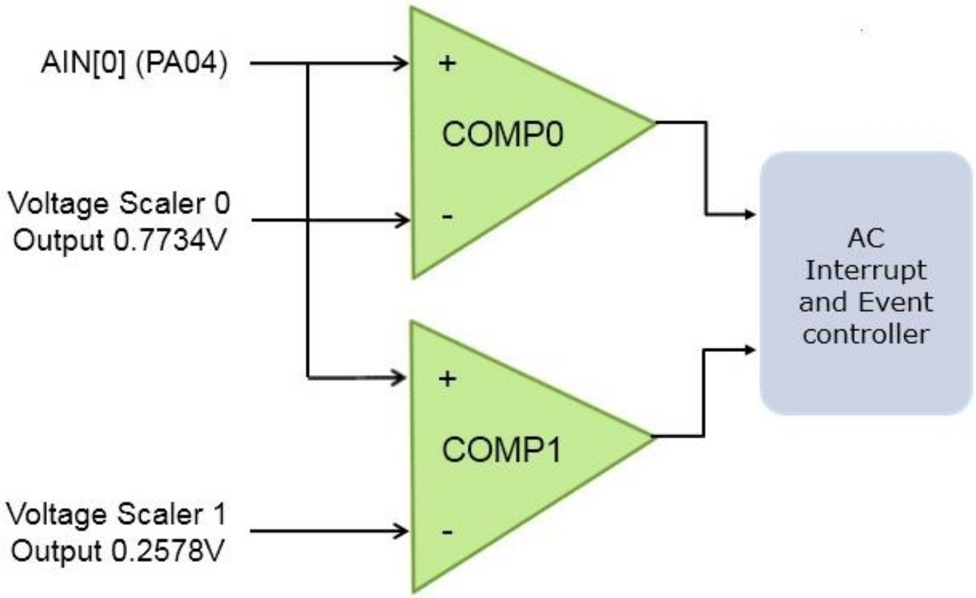
Following are the steps done to configure the AC as window detector.

1. Set pin PA12 as GPIO pin and configure the direction as output.
2. Select GCLK Generator 0 as GCLK source for AC module. The GCLK Generator 0 is clocked from internal OSC8M oscillator whose output is set to 8MHz.

3. Select AIN[0] pin (PA04) as positive input and internal voltage scaler 0 output as negative input for comparator 0. The voltage scaler 0 register is set to a value of 14 corresponding to a voltage output of 0.7734V, which is the upper threshold of the window.
4. Select AIN[0] pin (PA04) as positive input and internal voltage scaler 1 output as negative input for comparator 1. The voltage scaler 1 register is set to a value of 4 corresponding to a voltage output of 0.2578V which is the lower threshold of the window.
5. Set comparator 0 and 1 in continuous comparison mode with hysteresis and filter mode disabled.
6. Set interrupt mode selection as toggle for both the comparators.
7. Enable window mode operation and set window interrupt to get triggered when input voltage goes outside the threshold levels.
8. Enable window mode interrupt and register a callback function for the same.
9. Enable comparator 0 and 1 and then enable the AC module.
10. A Boolean variable is set to true inside the callback function indicating that the input signal has gone out of the threshold levels.
11. Inside while(1) loop in main() function the application waits for the Boolean variable to be set. Once set the variable is cleared and the pin PA12 is set indicating that the input has crossed the threshold level. The status register is read to check whether the signal has crossed above the upper threshold or crossed below the lower threshold. If the signal had gone above the upper threshold then the code will wait until it gets into the window and vice versa. Once the signal enters into the window the pin PA12 is cleared.

[redacted] shows the analog comparator connection.

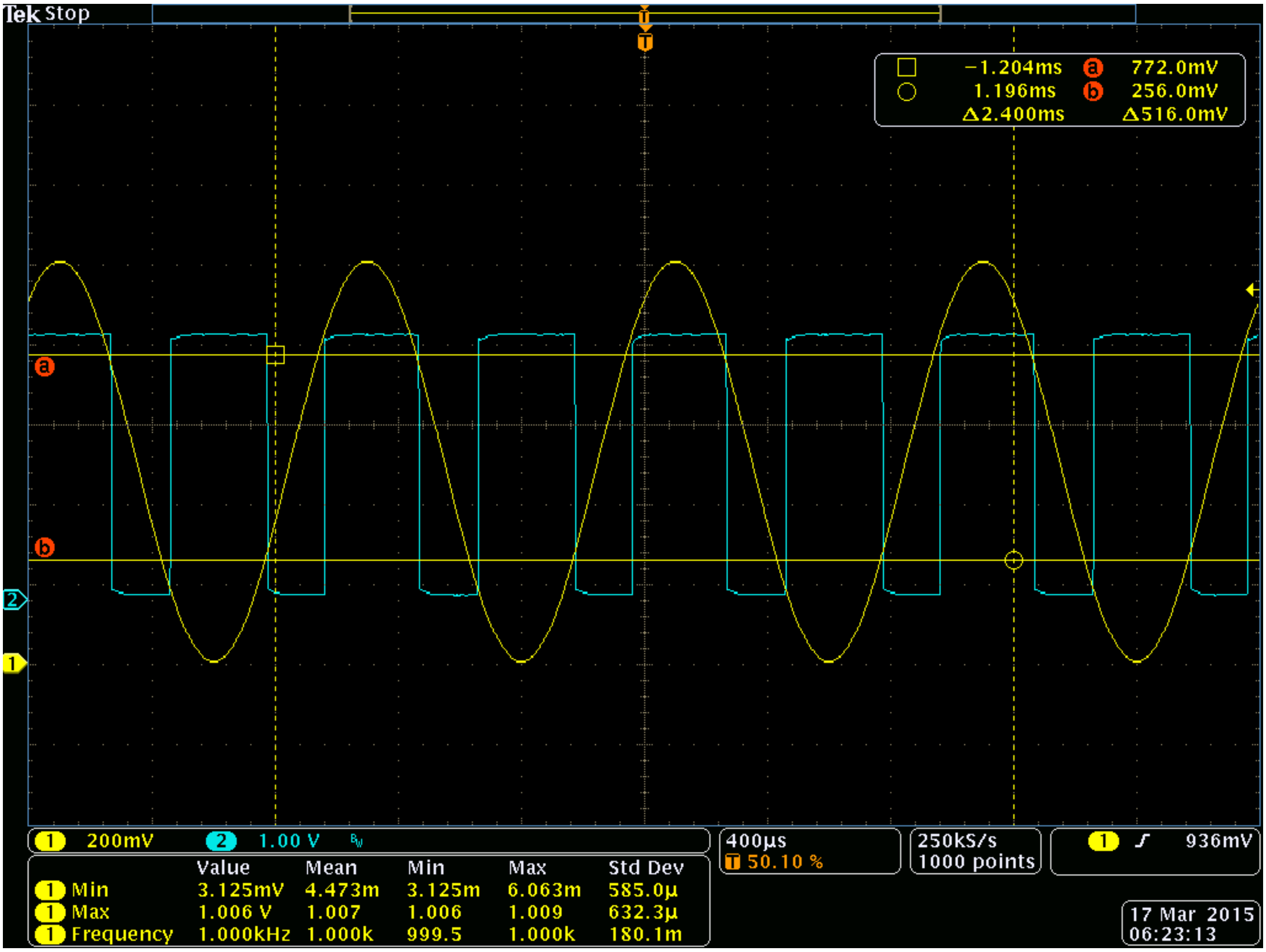
Figure 3-3. Analog Comparator Connections – Window Detector



The output is verified by taking a sine wave of frequency in few tens of Hz and signal level of ~1.0312V peak-to-peak with an offset voltage of 0.5156V from a function generator and applying it to pin 17 (PA04) in EXT1 header of SAM D21 Xplained Pro board. The output is taken from pin PA12. The signal on both PA04 pin and PA12 pin are captured in an oscilloscope for easier comparison.

Figure 3-4 shows the scope shot of the window mode detector output.

Figure 3-4. Window Detector – Output Scope Shot



	Value	Mean	Min	Max	Std Dev
1 Min	3.125mV	4.473m	3.125m	6.063m	585.0µ
1 Max	1.006 V	1.007	1.006	1.009	632.3µ
1 Frequency	1.000kHz	1.000k	999.5	1.000k	180.1m

Channel 1 (Yellow): Input sine wave signal applied on PA04.

Channel 2 (Blue): AC output taken from pin PA12.

Cursor a: AC window upper threshold (772mV).

Cursor b: AC window lower threshold (256mV).

3.4 Preventing False Spike Detection

This application demonstrates the usage of filter mode and hysteresis mode features of the AC module that helps in avoiding frequent toggling of the analog comparator when the positive input oscillates very close to the negative input level. This application is almost similar to the level crossing detector application. Additionally it has hysteresis mode enabled and filter mode set to majority 5 filter.

This application is selected by uncommenting the following line in main.c and commenting out other application selection macros.

```
#define AC_MODE      AC_FALSE_SPIKE_PREVENTER
```

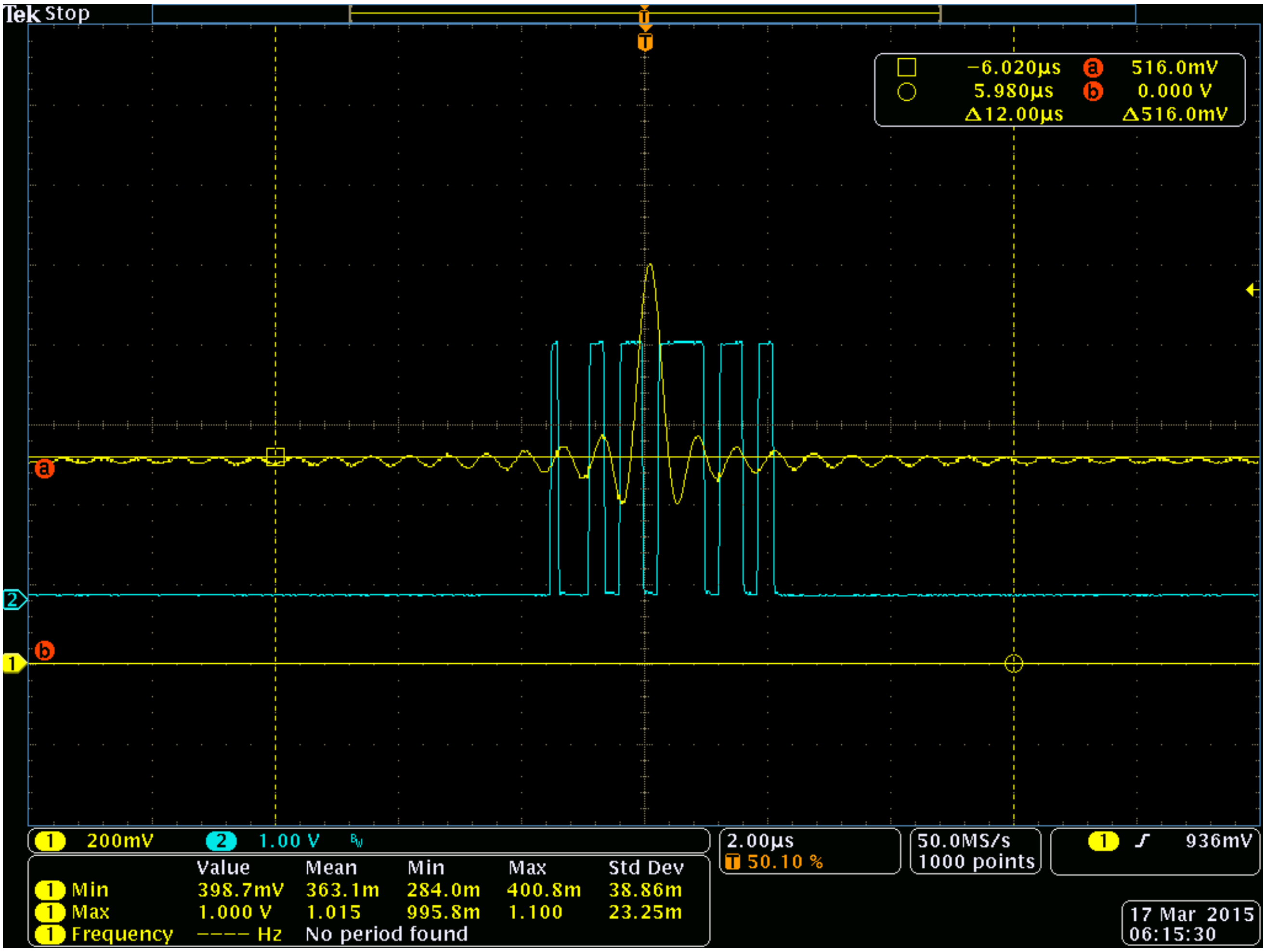
The comparator configuration steps and AC connection follows the same as level cross detector with only exception of hysteresis and filter mode being enabled in the configuration.

The output is verified by taking a sinc pulse of frequency in few tens of kHz and signal level of ~1.0312V peak-to-peak with an offset voltage of 0.5156V from a function generator and applying it to pin 17 (PA04) in EXT1 header of SAM D21 Xplained Pro board.

The output is taken from port pin PA12 (pin 7 on EXT3 header in SAM D21 Xplained Pro board). The signals on both PA04 pin and PA12 pin are captured with an oscilloscope for easier comparison.

Figure 3-5 shows the scope shot of the comparator output without enabling hysteresis and filter mode where the comparator output signal toggles several times for minor variation in the positive input signal around the negative input voltage level.

Figure 3-5. Analog Comparator Output with Hysteresis and Filter Mode Disabled



	Value	Mean	Min	Max	Std Dev
1 Min	398.7mV	363.1m	284.0m	400.8m	38.86m
1 Max	1.000 V	1.015	995.8m	1.100	23.25m
1 Frequency	---- Hz	No period found			

Channel 1 (Yellow): Input sinc pulse applied on PA04.

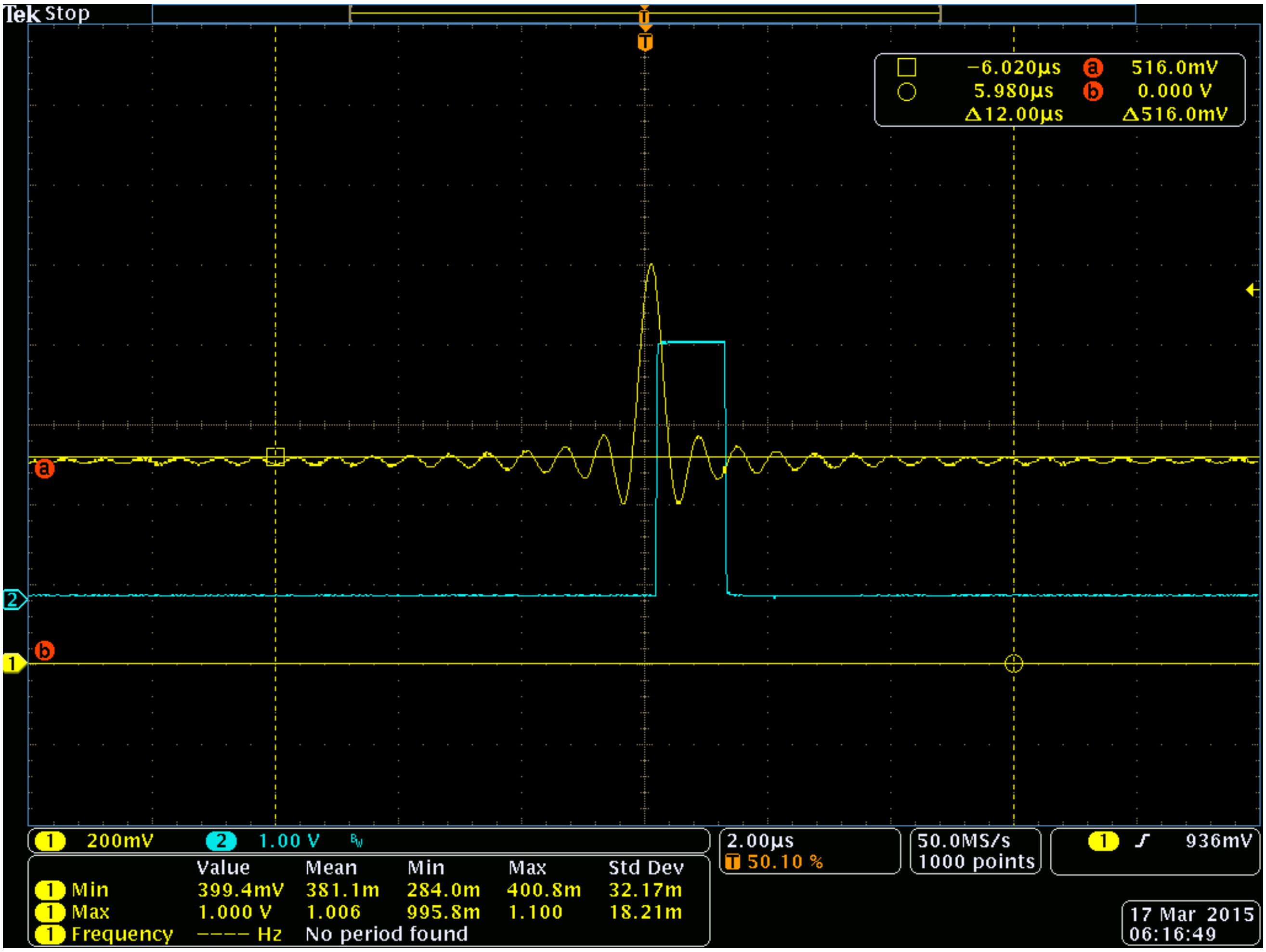
Channel 2 (Blue): AC output taken from pin PA12.

Cursor a: AC detect threshold (516mV).

Cursor b: Ground.

Figure 3-6 shows the scope shot of the comparator output with hysteresis enabled and filter mode set where the comparator output toggles only once for one sinc pulse.

Figure 3-6. Analog Comparator Output with Hysteresis and Filter Mode Enabled



	Value	Mean	Min	Max	Std Dev
1 Min	399.4mV	381.1m	284.0m	400.8m	32.17m
1 Max	1.000 V	1.006	995.8m	1.100	18.21m
1 Frequency	---- Hz	No period found			

Channel 1 (Yellow): Input sinc pulse applied on PA04.

Channel 2 (Blue): AC output taken from pin PA12.

Cursor a: AC detect threshold (516mV).

Cursor b: Ground.

3.5 Grey Signal Frequency Measurement

Frequency measurements are typically done by applying an input signal to an I/O pin of a microcontroller and either by reading the digital input register to detect the voltage level or by detecting the rising or falling edges of the input signal to estimate the time delay between them.

The time stamping will be proper only when the input signal meets the VIL and VIH specification of the I/O pins. So in order to measure the frequency of an input signal that falls in the gray region (between VIL and VIH) an external amplification has to be done to make the signal meet the VIL and VIH limits.

The external amplification can be avoided in such cases by using the analog comparator. The gray signal of interest can be applied to the positive input and an appropriate threshold can be set as negative input for the comparator.

The level crossing detector application itself can be used for this purpose, which enables the frequency measurement of a signal that falls in the gray area.

3.6 SleepWalking with Analog Comparator

This application explains how the analog comparator can be effectively used with RTC and event system modules to perform comparisons in standby sleep mode periodically and to wake the device only when the input signal crosses the threshold. On detection the I/O pin PA12 will be toggled.

This application is selected by uncommenting the following line in main.c and commenting out other application selection macros.

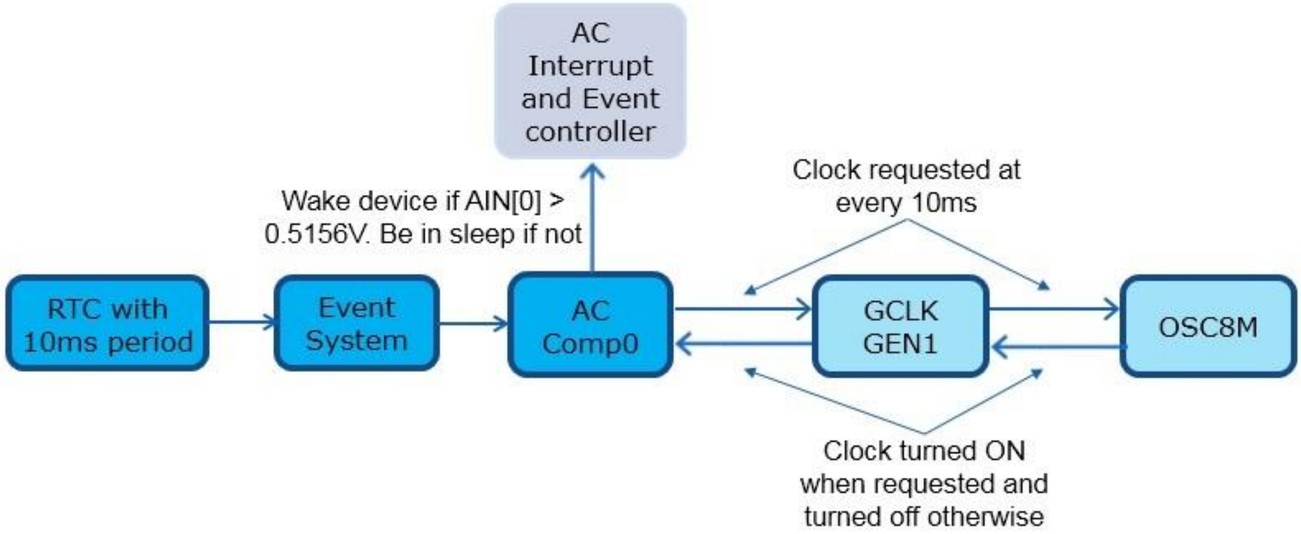
```
#define AC_MODE      AC_SLEEP_WALKING
```

Following are the steps to configure the AC in SleepWalking mode:

1. Set pin PA12 as GPIO pin and configure the direction as output.
2. Select GCLK Generator 1 as GCLK source for AC module. The GCLK Generator 1 is clocked from internal OSC8M oscillator whose output is set to 8MHz.
3. Enable the run in standby bit (RUNSTDBY) for the AC so that the analog comparator runs also in standby sleep mode.
4. Select AIN[0] pin (PA04) as positive input and internal voltage scaler output as negative input for comparator 0. The voltage scaler register is set to a value of 9 corresponding to a voltage output of 0.5156V.
5. Set comparator 0 in single shot comparison mode with hysteresis and filter mode disabled.
6. Set interrupt mode selection as toggle for comparator 0.
7. Enable comparator 0 interrupt and register a callback function for the same.
8. Configure the event control register of AC such that comparator 0 performs a comparison on an incoming event (comparator is configured as event user).
9. Enable comparator 0 and then enable the AC module.
10. Toggle pin PA12 inside the callback function.
11. Configure the RTC module in 16-bit mode to run from GCLK Generator 2 (sourced by OSCULP32K oscillator). The prescaler for GCLK Generator 2 is set to 32 such that a 1024Hz clock is applied to the RTC module.
12. The RTC period is set to 10 (for ~10ms period) and the RTC event is enabled such that an event will be generated on RTC overflow (RTC is configured as event generator).
13. An event system channel (0) is configured with RTC overflow as event generator and AC start comparison function as event user.
14. Sleep mode is set to standby sleep mode and the device is entered into standby sleep all the time.

Figure 3-7 shows the peripheral connection inside the microcontroller during SleepWalking.

Figure 3-7. Analog Comparator SleepWalking – Internal Peripheral Connections


AC Interrupt and Event controller
Wake device if AIN[0] > 0.5156V. Be in sleep if not
Clock requested at every 10ms
RTC with 10ms period
Event System
AC Comp0
GCLK GEN1
OSC8M
Clock turned ON when requested and turned off otherwise

When the application runs the device will enter into standby sleep mode after all the peripheral configuration. RTC will generate an event every 10ms, which is routed through the event channel to the AC module. The AC module will now request the GCLK to make a comparison. The GCLK will then request the OSC8M to supply the clock and the comparison happens. If the input signal is not above the threshold there is no need to wake the device and hence the clock request will be withdrawn making the clock source (OSC8M) to off. On the other hand, if the input signal is above the threshold the device will be woken up, pin PA12 will be toggled, and the device again enters into standby sleep mode.

The output is verified by applying a DC input voltage to pin 17 (PA04) in EXT1 header of SAM D21 Xplained Pro board.

The output is taken from pin PA12. When the input signal is above 0.5156V pin PA12 will be toggled.

The main advantage of using SleepWalking approach is to reduce power consumption by doing AC comparison activity in standby sleep mode without waking up the CPU. Following are current consumption numbers measured without using SleepWalking approach (device kept in active mode all the time) and with using SleepWalking approach.

Table 3-1. Current Consumption

SleepWalking Mode	Condition	Current Consumption
No SleepWalking	Device in active mode always	~1.3mA
With SleepWalking	AC positive input always lesser than negative input (device not waken up but still analog comparison performed every 10ms)	~19.3µA
	AC positive input goes above negative input once in every 100ms (device wakes up every 100ms with analog comparison performed every 10ms)	~26.5µA

4 References

1. ATSAMD21J18A Datasheet (http://www.atmel.com/Images/Atmel-42181-SAM-D21_Datasheet.pdf).

5 Revision History

Doc Rev.	Date	Comments
42473A	06/2015	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.