APPLICATION NOTE

# AVR32859: ATSAM4L-EK User Guide

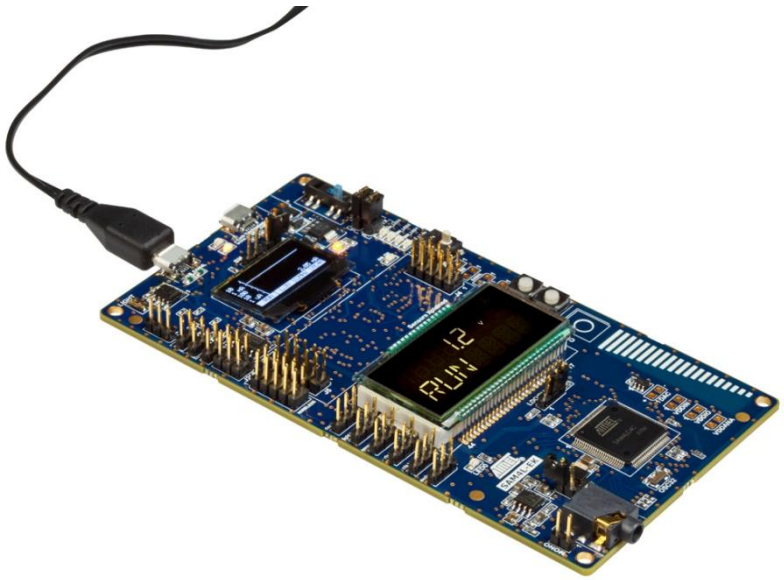
## Atmel SAM4L

## Features

- ATSAM4L-EK kit
- Board description
- Using the demonstration firmware

## Introduction

The Atmel® | SMART™ ATSAM4L-EK is a reference design and development system for the 32-bit ARM® Cortex™ -M4 ATSAM4LC4C microcontroller from Atmel Corporation. The kit is equipped with a rich set of peripherals that make the ATSAM4L-EK a perfect evaluation platform. This guide shows the user how to quickly get started with this kit.



# 1 SAM4L-EK Description

## 1.1 SAM4L-EK Board Features

**Table 1-1. SAM4L-EK Board Features**

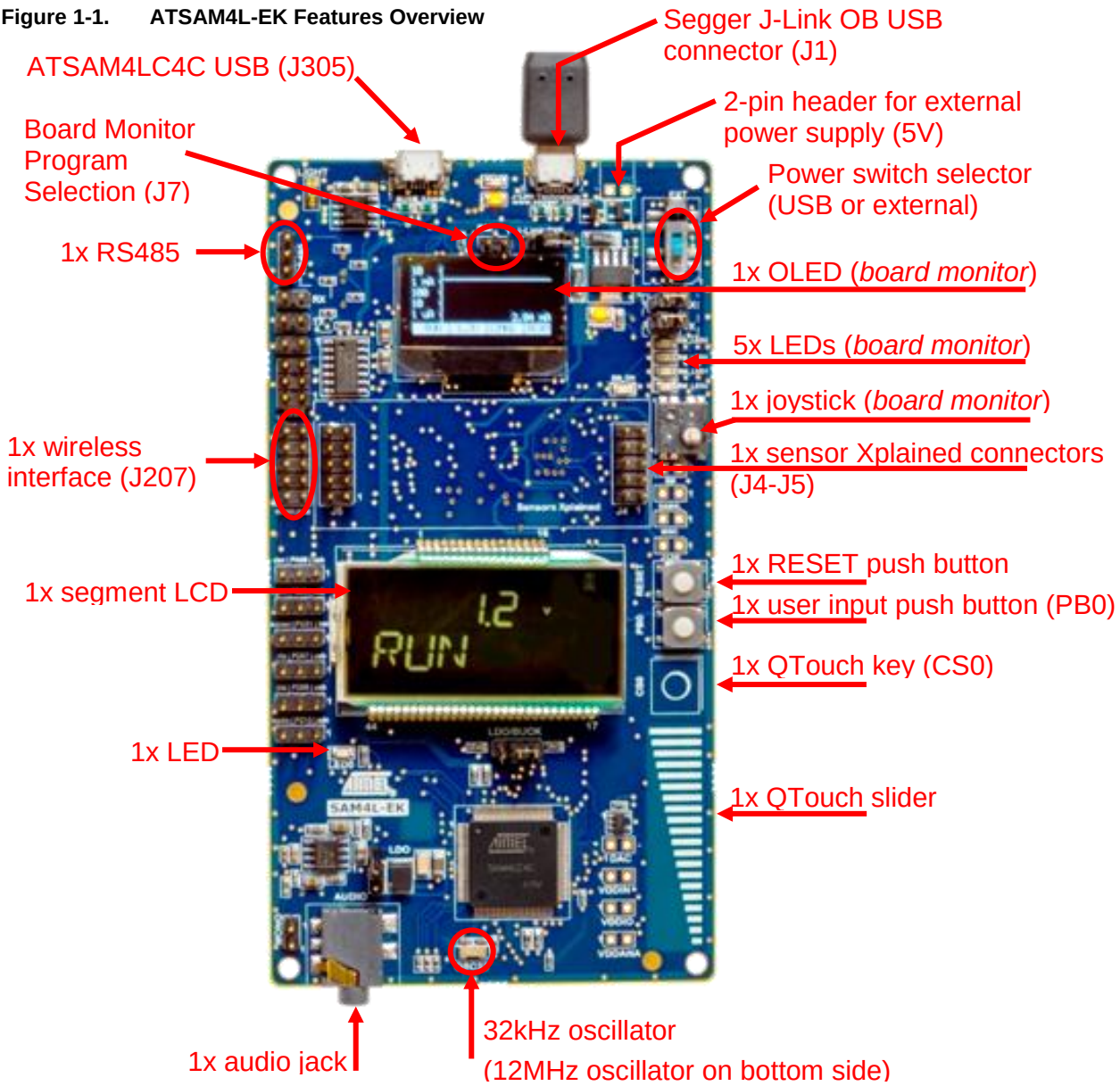
| Characteristics | Specification |
|---|---|
| MCU | ATSAM4LC4C (256KB flash, 32KB RAM), powered in 3.3V |
| Clock | 12MHz crystal<br>32.768kHz crystal |
| Connector | 1x Micro-AB (Micro-USB 2.0) for the main CPU ATSAM4LC4C<br>1x Micro-B (Micro-USB 2.0) for embedded debugger Segger J-Link-OB<br>1x Audio jack connector (3.5mm)<br>1x Sensors Xplained board connector (2x 10-pin headers)<br>1x RS485 header (3-pin) |
| Board power supply | 5V DC from ATSAM4LC4C USB<br>5V DC from Segger J-Link-OB USB<br>5V DC from a 2-pin header |
| Board monitor | Dedicated MCU for power measurement of the ATSAM4LC4C (VDDIN, VDDIO, VDDANA)<br>1x OLED Display (128x64)<br>5x LEDs<br>1x joystick<br>1x USART connected to the ATSAM4LC4C MCU<br>1x TWI connected to the ATSAM4LC4C MCU |
| 4x40 segment LCD | Connected to the ATSAM4LC4C LCD interface |
| Memory | 1x serial flash AT25DF641A |
| User interface for the ATSAM4LC4C | 1x QTouch® button<br>1x QTouch slider<br>1x RESET button<br>1x push button (PB0)<br>1x LED (LED0)<br>1x light sensor |

## 1.2 Board Interface Connection

The following connections are supported by the board:

- Micro-B (Micro-USB) connector for the embedded debugger and serial debug COM port interface Segger J-Link OB (USB connector J1)
- Micro-AB (Micro-USB) connector for the ATSAM4LC4C USB interface (USB connector J305)
- 1x Wireless 10-pins interface (with support of both ATEXTBT and RF2xx interfaces) (10-pin headers J207)
- 1x audio jack connector
- 1x RS485 connector for industrial transmitter (RS485)
- 2-pin header for external power supply (5V DC)

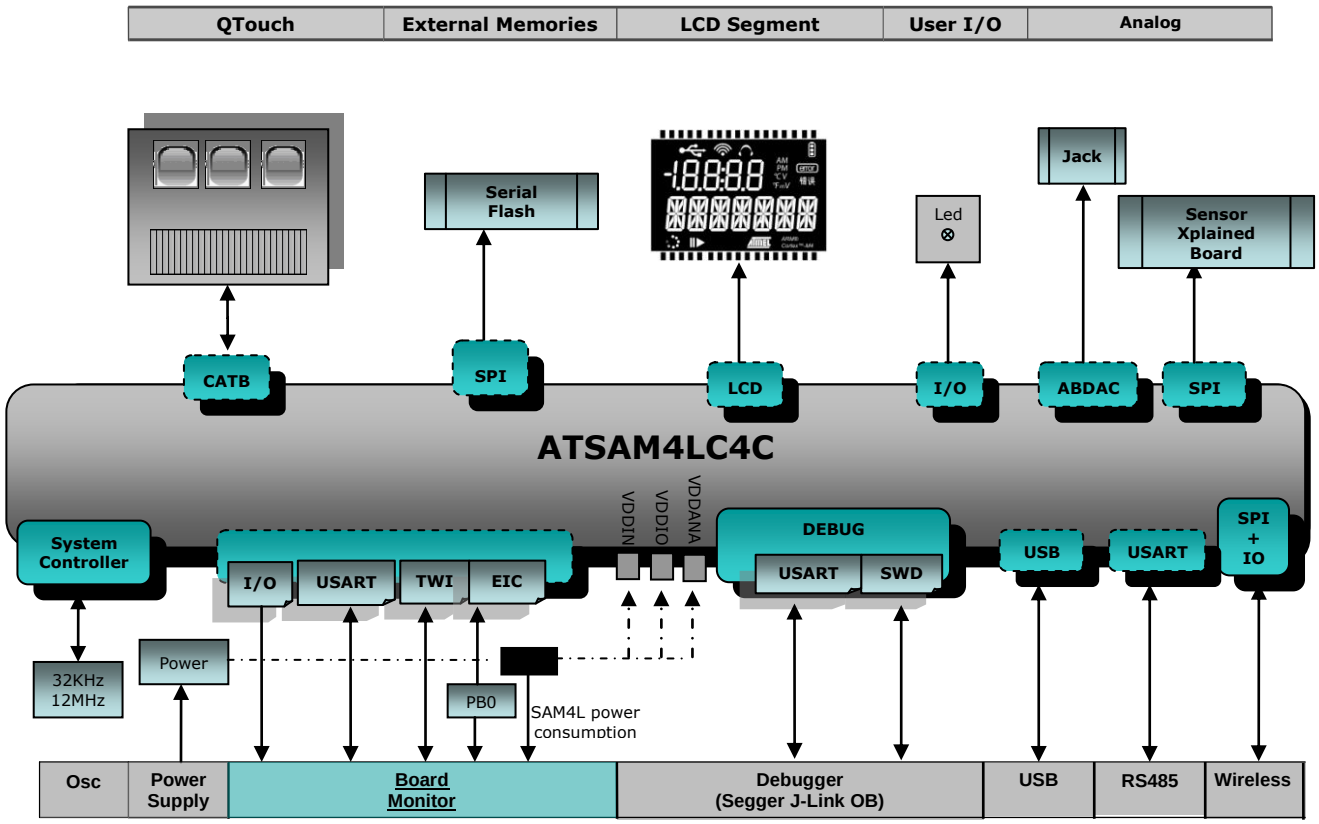
**Figure 1-1. ATSAM4L-EK Features Overview**



# 2 Board Description

## 2.1 Hardware

**Figure 2-1. ATSAM4L-EK Block Diagram**



## 2.2 Powering the Board

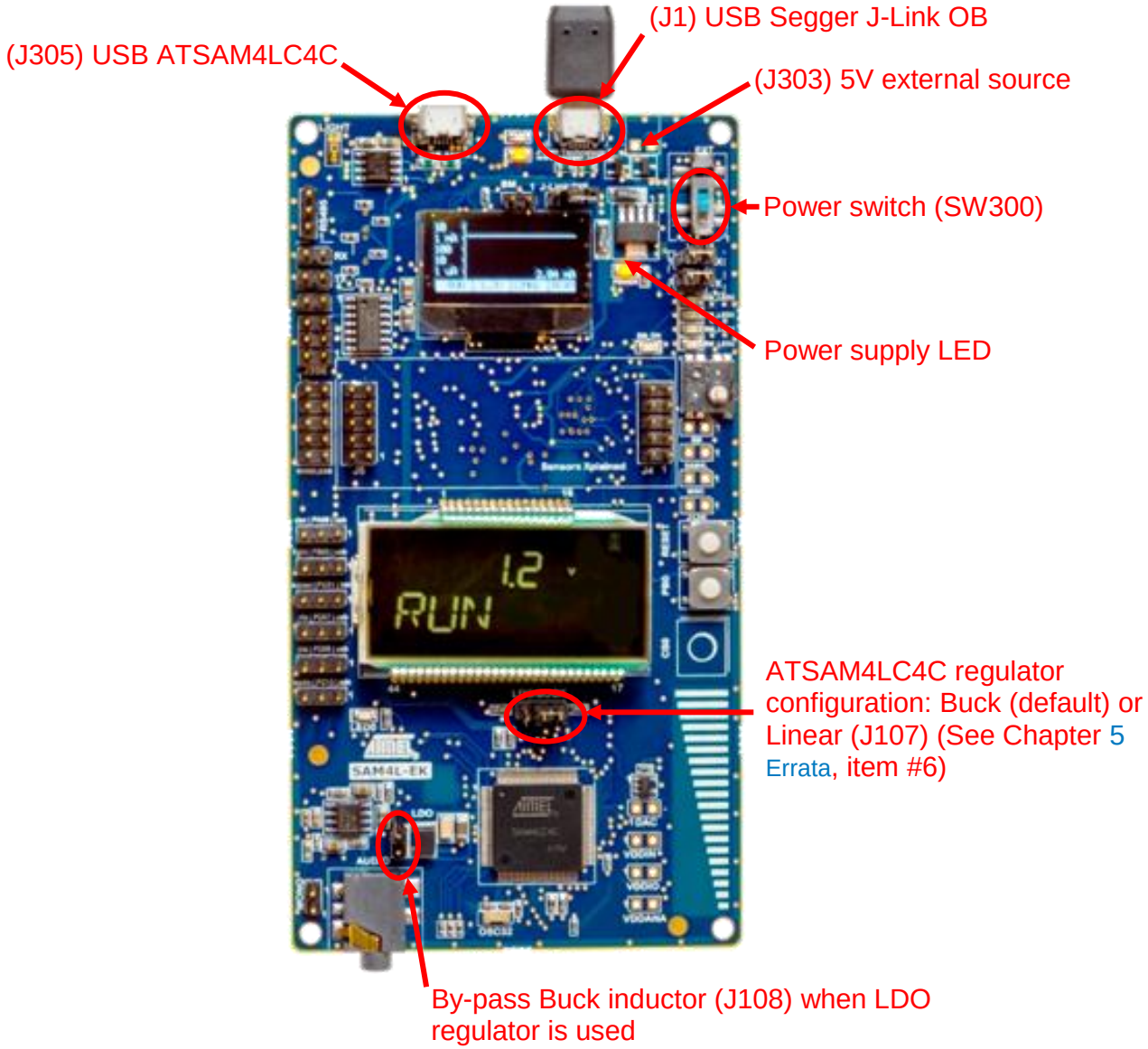
The ATSAM4L-EK offers three interfaces to power the board:

- USB embedded debugger Segger J-Link OB (J1)
- USB ATSAM4LC4C (J305)
- External 5V (DC) source connected to the J303 2-pin header

To select the USB or external power supply, a mechanical switch is used (SW300).

The default kit configuration is using ATSAM4LC4C buck regulator configuration, where ATSAM4LC4C is powered at 3.3V (VDDIN, VDDIO, VDDANA).

**Figure 2-2. Power Supply Sources**



## 2.3 Buttons/LED

The ATSAM4L-EK is equipped with one user push button and one LED. The push buttons consist of momentary push button switches mounted directly to the board. When any switch is pressed it will cause a low (zero) to appear at the associated input pin.

**Table 2-1. User I/O Mapping**

| GPIO | Feature |
|---|---|
| PC03 | Push button PB0 |
| PC10 | LED0 |

## 2.4 QTouch

The ATSAM4L-EK is equipped with one QTouch button and one QTouch slider using three channels.

**Table 2-2. QTouch I/O Mapping**

| GPIO | Feature |
|---|---|
| PB04 | QTouch Button |
| PB02 | QTouch Slider Channel 0 |
| PA04 | QTouch Slider Channel 1 |
| PA05 | QTouch Slider Channel 2 |
| PB03 | QTouch Discharge Pin |

## 2.5 RS485

The ATSAM4L-EK is equipped with a RS485 interface. To enable the RS485 connector:

- Close J200 and J203 jumpers
- Open J600 and J601 jumpers

**Table 2-3. RS485 I/O Mapping**

| GPIO | Feature | Jumper |
|---|---|---|
| PB06 | USART_RTS | |
| PA08 | USART_CTS | J104.2 to J104.3 |
| PC02 | USART_RXD | Close J200 and J203 |
| PA07 | USART_TXD | Open J600 and J601 |

## 2.6 Light Sensor (ADC)

The ATSAM4L-EK is equipped with one light sensor connected to one of the ADC channel. To enable it, close the jumper J101.2 to J101.3.

**Table 2-4. Light Sensor I/O Mapping**

| GPIO | Feature | Jumper |
|---|---|---|
| PB05 | ADC Channel | Close J101.2 to J101.3 |

## 2.7 Serial Flash

The ATSAM4L-EK is equipped with one serial flash AT25DF641, connected through SPI.

**Table 2-5. Serial Flash I/O Mapping**

| GPIO | Feature |
|---|---|
| PC04 | SPI_MISO |
| PC05 | SPI_MOSI |
| PC06 | SPI_SCK |
| PC00 | SPI_CS |

## 2.8 Audio

The ATSAM4L-EK is equipped with one audio amplifier. To enable it, close J105.2 to J105.3 jumper.

**Table 2-6. Audio I/O Mapping**

| GPIO | Feature | Jumper |
|---|---|---|
| PC09 | Audio ABDAC Channel 0 | Close J105.2 to J105.3 |
| PC13 | Audio ABDAC Channel 1 | Close J105.2 to J105.3 |

## 2.9 SAM4LC4C USB

The ATSAM4L-EK is equipped with one USB connector for the SAM4LC4C USB.

**Table 2-7. USB I/O Mapping**

| GPIO | Feature | Jumper |
|---|---|---|
| PA25 | USB DM | |
| PA26 | USB DP | |
| PB05 | USB ID pin (GPIO) | Close J101.1 to J101.2 |
| PC07 | USB Over current Detected (GPIO) | Close J103.1 to J103.2 |
| PC08 | USB Over current Enabled (GPIO) | Close J104.1 to J104.2 |

## 2.10 Wireless

The ATSAM4L-EK is equipped with one 10-pin Wireless extension connector, to connect other Atmel boards like ATEXTBT or AT86RF2xxx.

**Table 2-8. Wireless I/O Mapping**

| GPIO | Feature |
|---|---|
| PC04 | SPI_MISO |
| PC05 | SPI_MOSI |
| PC06 | SPI_SCK |
| PA02 | SPI_CS |

## 2.11 Sensors Xplained Connection

The ATSAM4L-EK is equipped with one Sensor Xplained extension connector. To enable it, close J102.2 to J102.3 jumpers.

**Table 2-9. Sensors Xplained I/O Mapping**

| GPIO | Feature | Jumper |
|---|---|---|
| PC04 | SPI_MISO | |
| PC05 | SPI_MOSI | |
| PC06 | SPI_SCK | |
| PC01 | SPI_CS | Close J102.2 to J102.3 |

## 2.12 Embedded Debugger

The embedded debugger J-Link OB Module is provided by Segger, featuring:

- 1x JTAG interface to the ATSAM4L4C4C
- 1x UART interface to the ATSAM4L4C4C
- 1x USB interface for a PC host

The USART connection between the ATSAM4L4C4C and the J-Link OB Module is configured as follow:

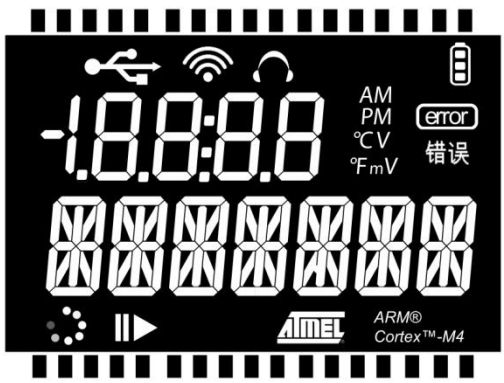
**Table 2-10. Embedded Debugger I/O Mapping**

| GPIO | Feature |
|---|---|
| PC11 | USART_RX |
| PC12 | USART_TX |

## 2.13 Custom Glass Display

This board is equipped with a 4x40 segment LCD, connected to the ATSAM4LC4C device through the LDCA controller.

**Figure 2-3. ATSAM4LC4C On-Board Custom Glass Display**



## 2.14 Board Monitor

The board monitor features:

- 1x OLED Display (128x64)
- 5x LEDs
- 1x joystick
- 1x USART connected to the ATSAM4LC4C MCU
- 1x TWI connected to the ATSAM4LC4C MCU

The USART connection between ATSAM4L and the board monitor is configured as:

**Table 2-11. Board Monitor I/O Mapping**

| GPIO | Feature | Jumper |
|---|---|---|
| PC02 | USART_RX | Close J600.1 to J600.2 |
| PA07 | USART_TX | Close J601.1 to J601.2 |

# 3 Using the Preloaded Firmware

The SAM4L-EK firmware demo features:

- Low power architecture:
  - Display ATSAM4LC4C power consumption on the board monitor OLED display
  - Use of the Power Save Mode
  - Use of the Power Scaling Mode (PS0, PS1, or PS2)
- Segment LCD controller:
  - Text display text and text scrolling
  - Hardware automatic animations
- Hardware touch (QTouch) support:
  - One capacitive button and one slider

## 3.1 Power-On the Board

- Power the board through the USB connector J1. The board starts the embedded debugger J-Link-OB USB enumerations



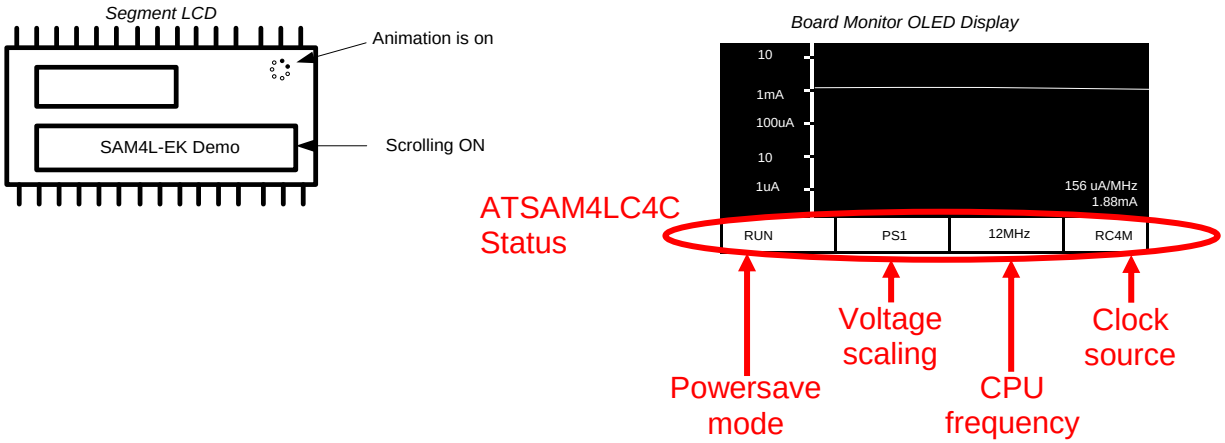

- If you do not have installer Atmel Studio 6, install the Segger J-Link OB driver available here: http://www.segger.com/jlink-software.html

Do not plug any USB cable on USB connector J305 as it will increase power consumption figures for the demo.

## 3.2 Running the Demonstration Firmware

### 3.2.1 Mode 1: “startup, RUN mode” (After Power-On Reset)

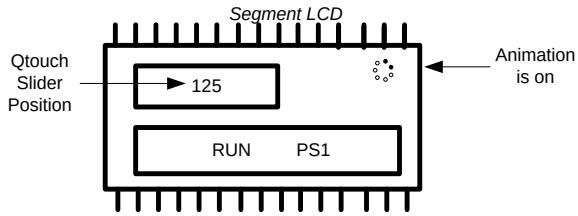
- ATSAM4LC4C is in active mode (12MHz on RC4M) with segment LCD and QTouch enabled. The ATSAM4LC4C starts with power scaling enabled (PS1)
- ATSAM4LC4C displays on the segment LCD:
  - A scrolling text message “SAM4L-EK Demo”
  - Scrolling animation to show SAM4L is in active mode
- ATSAM4LC4C sends through the USART its internal status to the board monitor (power save mode, voltage scaling, CPU frequency, and clock source)
- The board monitor displays on the OLED display the power consumption, power save mode, power scaling mode, CPU frequency, and clock source of ATSAM4LC4C

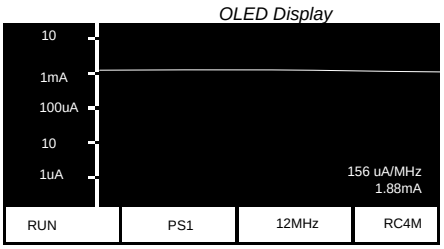



- After five seconds, the demo moves to Mode 2

### 3.2.2 Mode 2 “RUN mode with all features”

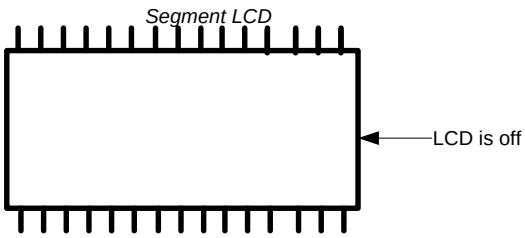
- The ATSAM4LC4C is in RUN mode. ATSAM4LC4C displays the voltage scaling mode (PS1, PS2, or PS0) on the segment LCD and its active mode (RUN)
- Using QTouch capacitive button CS0 will change voltage scaling configuration (PS1, PS2, or PS0)
- The QTouch slider position will be displayed (0 … 255) if QTouch slider event is detected
- The ATSAM4LC4C sends its internal status through USART to the board monitor (power save mode, voltage scaling, CPU frequency, and clock source) each time it changes
- Using PB0 button will disable QTouch and segment LCD, the demo move to Mode 3

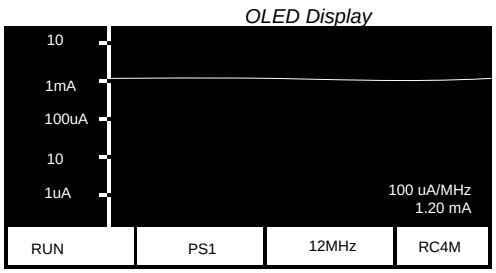



### 3.2.3 Mode 3 “RUN mode with no features”

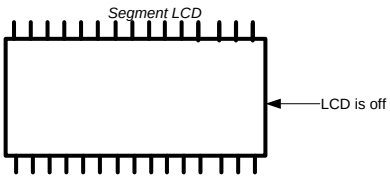
- The ATSAM4LC4C in active mode (12MHz on RC4M) with power scaling mode PS1. QTouch and LCD are disabled. The AST is still running with external OSC32K oscillator
- The ATSAM4LC4C is executing the Fibonacci algorithm
- Using PB0 button will enter the WAIT Power Save Mode and move the demo to Mode 4

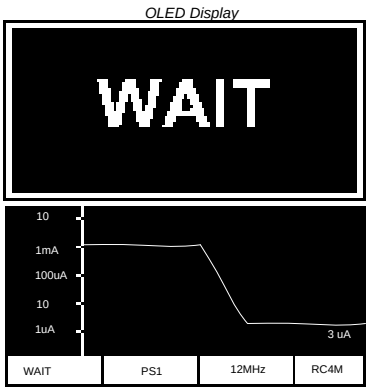


### 3.2.4 Mode 4 “WAIT mode”

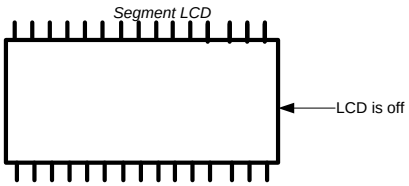
- The ATSAM4LC4C is in WAIT mode with power scaling mode PS1. The AST is still running with external OSC32K oscillator
- The ATSAM4LC4C sends its new internal status through USART to the board monitor (power save mode, voltage scaling, CPU frequency, and clock source). The Board monitors displays a splash screen indicating the new Power Save Mode (WAIT here)
- Using PB0 button will enter the RETENTION Power Save Mode and move the demo to Mode 5

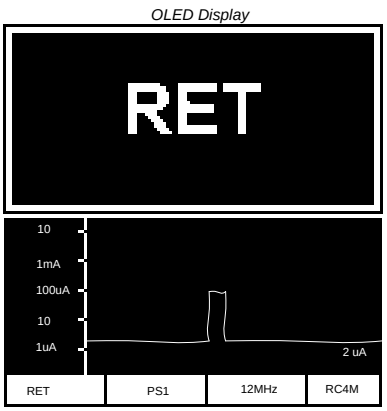



### 3.2.5 Mode 5 “RETENTION mode”

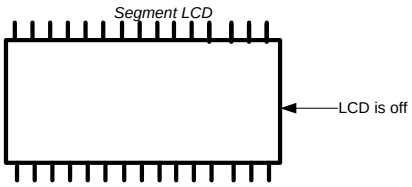
- The ATSAM4LC4C in RET mode with power scaling configuration PS1. The external OSC32K oscillator is still running
- The ATSAM4LC4C sends its new internal status through USART to the board monitor (power save mode, voltage scaling, CPU frequency, and clock source)
- Using PB0 button will enter the BACK-UP Power Save Mode and move the demo to Mode 6

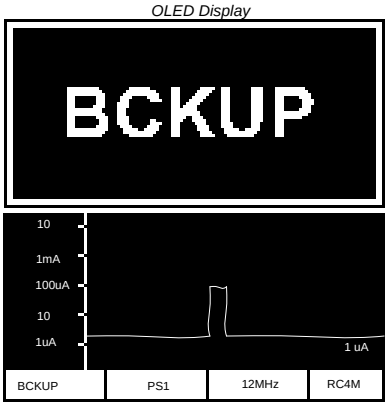


### 3.2.6 Mode 6 “BACKUP mode”

- The ATSAM4LC4C is in backup mode with power scaling mode PS1. The external OSC32K oscillator is still running
- The ATSAM4LC4C sends its new internal status through USART to the board monitor (power save mode, voltage scaling, CPU frequency, and clock source)
- Using PB0 button will go back to Mode 1 in RUN mode



# 4 Board Monitor

## 4.1 Measurement Stage

The board monitor is in charge of current consumption measurement of the ATSAM4LC4C (VDDIN, VDDIO, VDDANA). The current measurement stage works in the range 1µA to 50mA. The sampling rate of the current measurement stage is 8kHz. Each current measure is filtered at 50Hz.

The accuracy is:

- 10% of error on the measure in the range 1µA to 10µA
- 1% of error on the measure in the range 10µA to 50mA

## 4.2 Commands Between SAM4L and the Board Monitor

### 4.2.1 Register Map

| Offset | Name | Type | Description | Confirm available in firmware version |
|---|---|---|---|---|
| 0x01 | BM_POINTER_CTRL | W | Enable/disable the board monitor mouse-like pointer | 1.2 and above |
| 0x02 | BM_POINTER_MOVE | W | Send new mouse pointer position | 1.2 and above |
| 0x03 | BM_CTRL | W | Enable/disable the board monitor | 1.2 and above |
| 0x04 | BM_LED_SET | W | Turn-on a LED of the board monitor | 1.2 and above |
| 0x05 | BM_LED_CLR | W | Turn-off a LED of the board monitor | 1.2 and above |
| 0x06 | BM_LED_TGL | W | Toggle a LED of the board monitor | 1.2 and above |
| 0x07 | BM_MCU_STATUS | W | Send MCU power saving information to the board monitor, in order to get them displayed on the OLED screen | 1.2 and above |
| 0x08 | BM_PULLUP_TWI | W | Enable/disable the pull-up on TWI lines | 1.3* and above |
| 0x09 | BM_PICOUART_SEND | W | Send PicoUart Frame | 1.3* and above |
| 0x0A | BM_MCU_GET_CURRENT | W | Send Current Consumption Measured Request | 1.3* and above |
| 0x0B | BM_TGL_BUTTON | W | Force Toggle of Button Line | 1.3* and above |
| 0x0C | BM_MCU_RET_CURRENT | R | Return Current Consumption Meas-ured | 1.3* and above |
| 0x0D | BM_MCU_GET_FIFO_FREE_SIZE | W | Send free size (in byte) of the board monitor command fifo request | 1.3* and above |
| 0x0E | BM_MCU_RET_FIFO_FREE_SIZE | R | Return the free size (in byte) of the board monitor command fifo | 1.3* and above |
| 0x0F | BM_PRINT_TEXT | W | Print Text On Board Monitor | 1.3* and above |
| 0x10 | BM_PRINT_CLEAR | W | Clear Print Text Area | 1.3* and above |
| 0x11 | BM_GET_FIRMWARE_VERSION | W | Send Firmware Version Request | 1.3* and above |
| 0x12 | BM_RET_FIRMWARE_VERSION | R | Return Firmware Version value | 1.3* and above |

Note: Board Monitor firmware is available here: http://www.atmel.com/tools/SAM4L-EK.aspx?tab=documents.

### 4.2.2 APIs Available for SAM4L

#### 4.2.2.1 BM_POINTER_CTRL – Board Monitor Pointer Control

**Prototype:**

```
void bm_mouse_pointer_ctrl(bool state);
```

**Parameter(s):**

state true to enable, false to disable.

**Return Value:**

**None.**

#### 4.2.2.2 BM_POINTER_MOVE – Board Monitor Pointer Move

**Prototype:**

```
void bm_mouse_pointer_move(uint32_t x, uint32_t y);
```

**Parameter(s):**

x x position. Range is [0; 127].

y y position range is [0; 63].

**Return Value:**

**None.**

#### 4.2.2.3 BM_CTRL – Board Monitor Control

**Prototype:**

```
void bm_ctrl(bool state);
```

**Parameter(s):**

state true to enable, false to disable.

**Return Value:**

**None.**

#### 4.2.2.4 BM_LED_SET – Board Monitor Led Set

**Prototype:**

```
void bm_led_set(uint32_t led);
```

**Parameter(s):**

led led number. Range is [0; 3].

**Return Value:**

**None.**

#### 4.2.2.5 BM_LED_CLR – Board Monitor Led Clear

**Prototype:**

```
void bm_led_clr(uint32_t led);
```

**Parameter(s):**

led led number. Range is [0; 3].

**Return Value:**

**None.**

#### 4.2.2.6 BM_LED_TGL – Board Monitor Led Toggle

**Prototype:**

```
void bm_led_tgl(uint32_t led);
```

**Parameter(s):**

led led number. Range is [0; 3].

**Return Value:**

**None.**

#### 4.2.2.7 BM_MCU_STATUS – Board Monitor Status

**Prototype:**

```
void bm_send_mcu_status(uint32_t voltage_scaling, uint32_t
sleep_mode, uint32_t cpu_freq, uint32_t cpu_src);
```

**Parameter(s):**

voltage_scaling Voltage scaling.

sleep_mode Sleep mode.

cpu_freq CPU frequency.

CPU source clock.

**Return Value:**

**None.**

#### 4.2.2.8 BM_PULLUP_TWI – Board Monitor Pull-up TWI

**Prototype:**

```
void bm_pullup_twi(bool state);
```

**Parameter(s):**

voltage_scaling Voltage scaling.

sleep_mode Sleep mode.

cpu_freq CPU frequency.

CPU source clock.

**Return Value:**

**None.**

#### 4.2.2.9 BM_PICOUART_SEND – Board Monitor PicoUart Send

**Prototype:**

```
void bm_send_picouart_frame(uint8_t frame, uint32_t
timeout_ms);
```

**Parameter(s):**

frame Frame to be sent.

timeout timeout_ms Value in ms.

**Return Value:**

**None.**

#### 4.2.2.10 BM_MCU_GET_CURRENT – Board Monitor Get Current

**Prototype:**

```
bool bm_get_mcu_current(uint32_t* sleep_mode, float*
current);
```

**Parameter(s):**

sleep_mode Sleep Mode Desired.

current Current Value Measured.

**Return Value:**

**True: Current Value is available.**

#### 4.2.2.11 BM_TGL_BUTTON – Board Monitor Toggle Button

**Prototype:**

```
bool bm_get_mcu_current(uint32_t* sleep_mode, float*
current);
```

**Parameter(s):**

sleep_mode Sleep Mode Desired.

current Current Value Measured.

**Return Value:**

**True: Current Value is available.**

#### 4.2.2.12 BM_MCU_GET_FIFO_FREE_SIZE – Board Monitor Get Free Fifo Size

**Prototype:**

```
bool bm_get_fifo_free_size(uint16_t* free_size);
```

**Parameter(s):**

free_size free size in byte.

**Return Value:**

**True: Current Fifo Size is available.**

#### 4.2.2.13 BM_PRINT_TEXT – Board Monitor Print Text

**Prototype:**

```
void bm_print_txt(uint8_t* str, uint8_t str_length);
```

**Parameter(s):**

str String pattern.

str_length String length pattern.

**Return Value:**

**None.**

#### 4.2.2.14 BM_PRINT_CLEAR – Board Monitor Clear

**Prototype:**

```
void bm_print_clear(void);
```

**Parameter(s):**

**None.**

**Return Value:**

**None.**

#### 4.2.2.15 BM_GET_FIRMWARE_VERSION – Board Monitor get Firmware Version

**Prototype:**

```
void bm_get_firmware_version(uint8_t *fw_version_minor,
uint *fw_version_major);
```

**Parameter(s):**

```
fw_version_minor Minor number of firmware version.
fw_version_major Major number of firmware version.
```

**Return Value:**

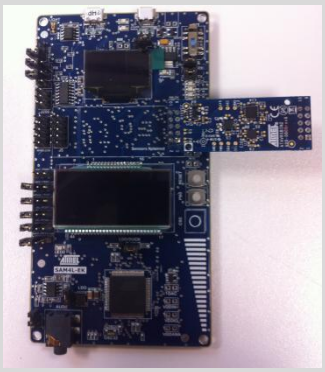
**None.**

## 4.3 Re-Program the Board Monitor

1. Open dos command.
2. Go into <Atmel Studio Installation Path>/atbackend.
3. Unzip ATSAM4L-EK_BoardMonitor.zip in this folder.
4. Close J7 header to select Board Monitor programming (see Figure 1-1).
5. Type (replace below “vx.y” to real version):

   *atprogram -t samice -i swd -d ATSAM3N4A program -c -f sam.applications.sam4l_ek_demo.board_monitor.sam3n_ek_vx.y.bin -o 0x400000*

# 5 Errata

| Bug | Description | Errata | Impacted version |
|---|---|---|---|
| #1 | Audio jack connection is wrong (J202). | Rework J202 connection by cutting off the connection of J202.5: connect J202.1 to left side of C202 and connect J202.2, J202.3, and J202.5 to GND. | Rev 2 |
| #2 | Q301 is mounted in the wrong way. | None. | Rev 2 |
| #3 | USART CTS/RTS signals are not correctly routed to RS485 transceiver (U200). | Follow this connection to make RS485 connection works:<br>• PA06 (USART0_RTS),<br>• PC08 (USART0_CTS),<br>• PC02 (USART0_RXD),<br>• PA07 (USART0_TXD)<br>To do so:<br>• Connect J100.2 to J103.3 with a wire,<br>• Close J104.2 to J104.3,<br>• Close J200,<br>• Close J203,<br>• Open the J600 and J601 | Rev 2 |
| #4 | J4 and J5 connectors must be swapped in order to correctly follow Xplained Board Pinout. | In order to use Sensors Xplained board, plug J1 header of Sensors Xplained board on J4 header of SAM4L-EK.<br> | All |
| #5 | Vbus monitor through PA06 cannot be used. | None. | All |
| #6 | Current consumption in Linear (LDO) mode is higher than in Switch-ing(Buck) mode for all sleep modes | The LDO/Buck selection jumper pulls down the PA02 pin which is internally pulled-up by the SAM4L demo software. This leads to additional 70µA consumption on the total chip current. To get the expected power numbers, remove the LDO/Buck selection jumper after powering up the board.<br>Note: LDO/Buck selection is latched in the device at POR, resetting the device by other means (Button, WDT) won't change the regulator mode. | All |

# 6 Reference and Further Information

## 6.1 Device Datasheet

The device datasheet contains block diagrams of the peripherals and details about implementing firmware for the device. It also contains the electrical specifications and expected characteristics of the device.

The datasheet is available on http://www.atmel.com/ in the Datasheets section of the product page.

## 6.2 Detailed Hardware Reference (and Associated Errata)

More detailed hardware information for this kit can be found in the file ATSAM4L-EK_Hardware-References.zip available on the Atmel web page dedicated to this kit: www.atmel.com/

The ATSAM4L family of devices is specified in the SAM4L Series datasheet. Always use this document as a reference throughout the development life cycle of an application destined to run on a SAM4L device.

## 6.3 Tools

To be able to develop applications for 32-bit ARM devices and build binaries for ARM targets and program a 32-bit ARM device, Atmel and its partners provide several tools supported on multiple host targets.

- Atmel Studio 6 is the integrated development environment (IDE) for developing and debugging Atmel ARM Cortex-M and Atmel AVR® microcontroller (MCU) based applications. The Atmel Studio 6 IDE gives you a seamless and easy-to-use environment to write, build, and debug your applications written in C/C++ or assembly code.

  http://www.atmel.com/microsite/atmel_studio6/default.aspx
- IAR Embedded Workbench®: IAR™ Embedded Workbench with its optimizing C and C++ compiler provides full support and generates very compact and efficient code for ARM device.

  http://www.iar.com/en/Products/IAR-Embedded-Workbench/ARM/

# 7 Revision History

| Doc Rev. | Date | Comments |
|---|---|---|
| 42026F | 08/2014 | Added PS2 mode support for the SAM4L-EK demo.<br>Updated board monitor (v1.4) to support PS2 display. |
| 42026E | 04/2013 | The logo is added.<br>Fixed some typos. |
| 42026D | 03/2013 | Added Board Monitor firmware link path.<br>Errata update for linear (LDO) mode current measurement in sleep modes. |
| 42026C | 01/2013 | Errata added. |
| 42026B | 12/2012 | • “sleep mode” has been replaced by “power save mode”<br>Section 3.2 is updated:<br>• Updated OLED display screenshots<br>• Added comment for QTouch slider management in the demo<br>Chapter 4 is added. |
| 42026A | 09/2012 | Initial document release. |




Atmel | 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**

© 2014 Atmel Corporation. / Rev.:Atmel-42026F-ATSAM4L-EK-User-Guide-ApplicationNote_082014.

Atmel®, Atmel logo and combinations thereof, AVR®, DataFlash®, Enabling Unlimited Possibilities®, QTouch®, and others are registered trademarks or trademarks of Atmel Corporation in U.S. and other countries. ARM®, ARM Connected® logo, Cortex™, and others are the registered trademarks or trademarks of ARM Ltd. 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.

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.