
Atmel AT01639: XMEGA-C3 Xplained Software User Guide

Atmel XMEGA C

Features

- OLED display
- Sensors readout
 - Light sensor
 - Temperature sensor
- CPU load
- QTouch® button demonstration
- microSD card
- Embedded file system
- USB device composite:
 - Mass Storage Class
 - Communication Device Class

Introduction

The Atmel® AVR® XMEGA®-C3 Xplained evaluation kit demo software is created to showcase the Atmel AVR ATxmega384C3 device. The demo samples the available sensors connected on ADC and sent the values through OLED display, USB or microSD card. The demo is controlled through mechanical switches and touch buttons.

This documentation describes the preloaded demo software and the software libraries available thru [Atmel Software Framework \(ASF\)](#) for XMEGA-C3 Xplained kit. The Atmel AT01636 application note describes the XMEGA-C3 Xplained hardware in detail.

Figure 1. The XMEGA-C3 Xplained kit.

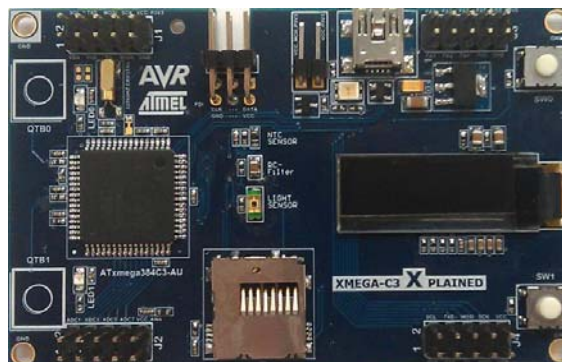


Table of Contents

1. Running Demo Application	3
1.1 Description	3
1.2 User interface	4
1.3 USB feature	5
1.3.1 USB Mass Storage Class	5
1.3.2 USB Communication Device Class	5
1.4 Running the demo application	6
2. Inside the Demo Application	7
2.1 Overview	7
2.2 CPU load task	8
2.3 Atmel QTouch task	9
2.4 microSD task	9
2.5 USB task	10
2.6 Sampling task	10
2.7 Footprint	11
3. Building a New Application on XMEGA-C3 Xplained	12
3.1 New project on XMEGA-C3 Xplained	12
3.2 New example project from ASF on XMEGA-C3 Xplained	12
4. References and Further Information	13
4.1 Device datasheet	13
4.2 Detailed hardware references (and associated errata)	13
4.3 Tools	13
5. Revision History	14

1. Running Demo Application

1.1 Description

The Atmel XMEGA-C3 Xplained kit comes with a demo application. However, it is available in ASF through Atmel Studio 6 by creating a new example from ASF, named “Demo for XMEGA-C3 Xplained”.

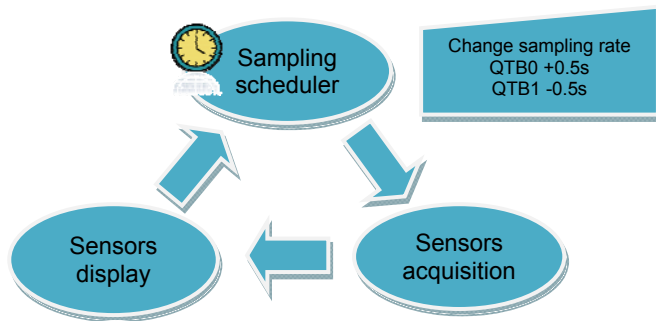
At power up, demo shows introduction screens, with explanation how to use the buttons. Then, the application starts with continuous sensors acquisition task scheduled with the real time counter (RTC). Sensor values are displayed on the OLED. Depending on user choice on mechanical buttons, these values can be either sent to the USB device Virtual COM port or stored on the microSD card.

The application benefits of the Low Power XMEGA architecture, and the CPU enters in sleep mode whenever possible. Thus, a task running in parallel to display CPU load which is the CPU time in active mode.

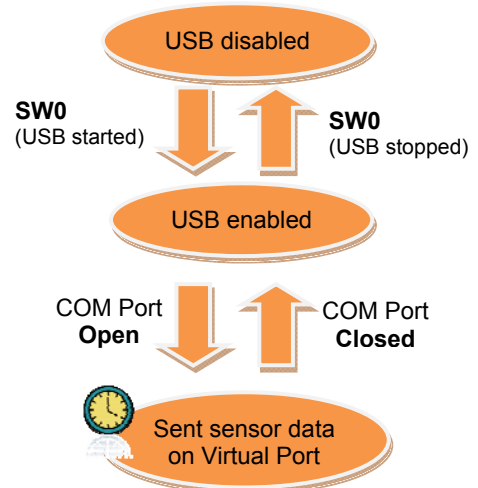
The software includes five tasks:

1. Sensors sampling task, managing the acquisition of the temperature and light sensors.
2. USB task, managing the connectivity with PC.
3. microSD task, managing data storage and accessibility.
4. QTouch task, scheduling the QTouch sensor measures.
5. CPU load task, displaying real time CPU activity (e.g. CPU time in active mode).

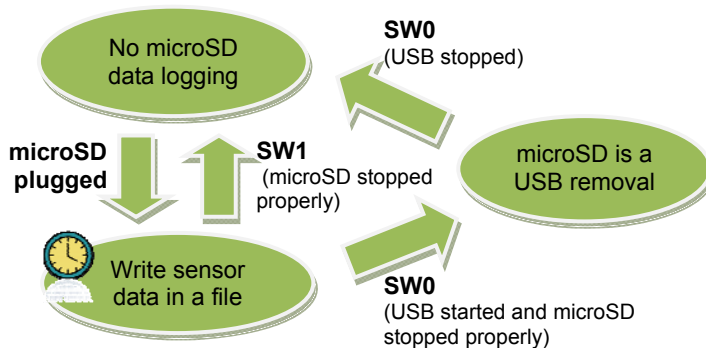
Sensors sampling task



USB task



microSD task



1.2 User interface

The Atmel XMEGA-C3 Xplained kit is powered through the USB connector. The demo application is controlled with the mechanical switches (SW0, SW1) and the QTouch buttons (QT™B0, QT B1). The activity monitoring is done through the LEDs and the OLED display.

Note: microSD is activated once plugged and must be removed from connector only after SW1 is pressed to close properly any data log file which may have been opened.

Figure 1-1. Hardware resources used in the XMEGA-C3 Xplained demo.

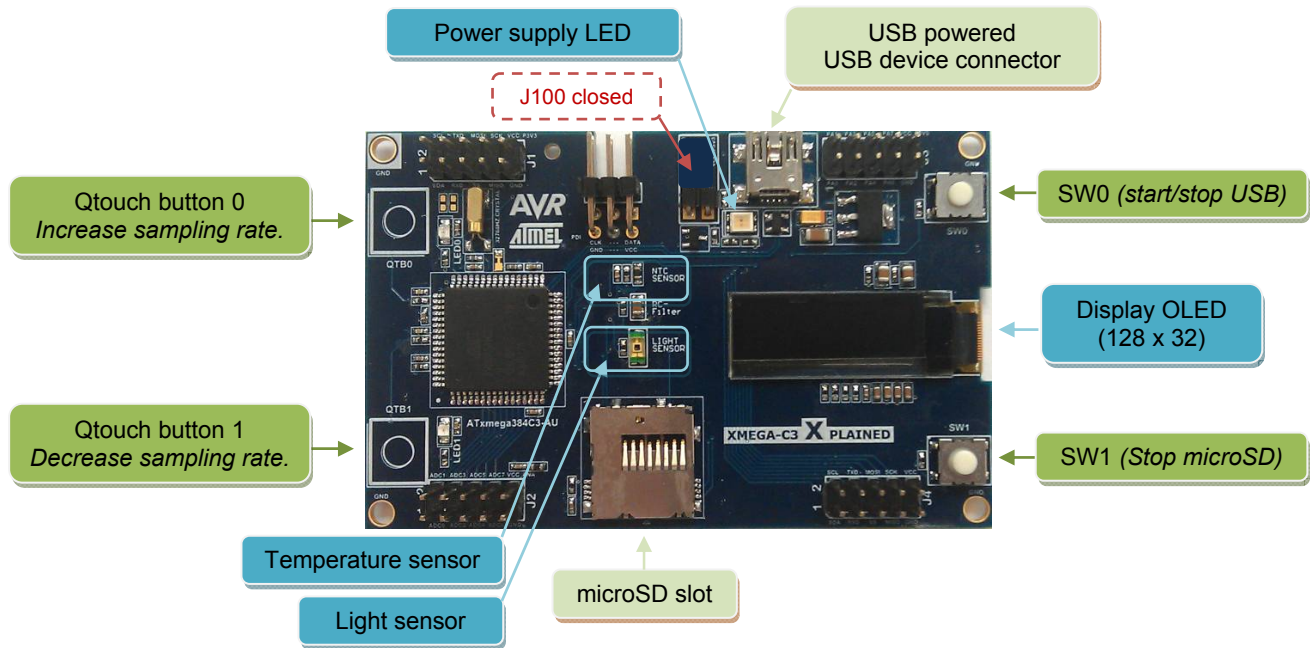
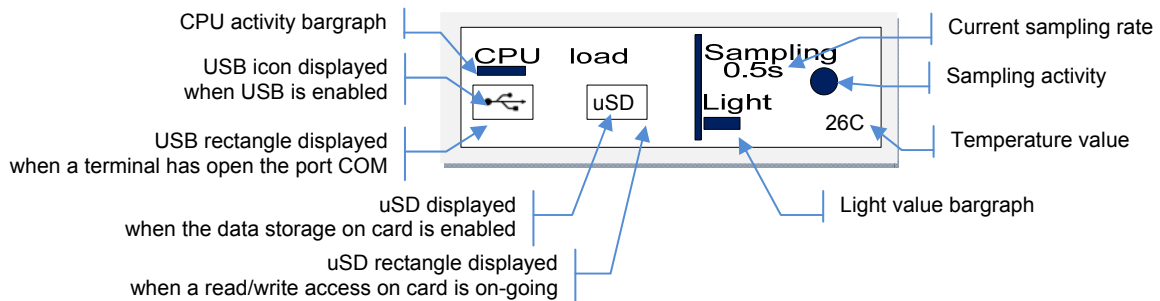


Figure 1-2. Information on the OLED display for the XMEGA-C3 Xplained demo.



1.3 USB feature

When powering the kit through the USB connector, the Atmel XMEGA-C3 Xplained starts without any USB PC enumeration. USB support is handled by embedded firmware when SW0 is pressed.

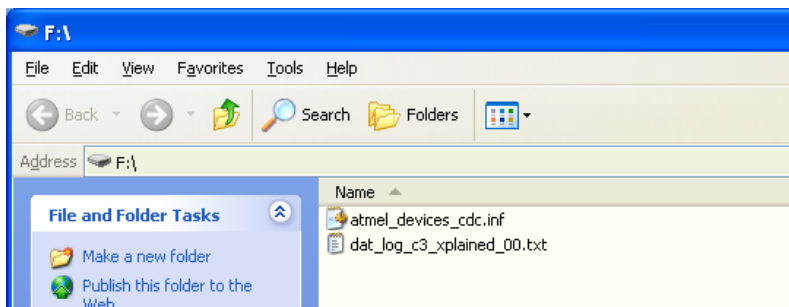
The USB feature provides a USB device composite including:

- a USB Mass Storage Class (MSC)
- a USB Communication Device Class (CDC)

1.3.1 USB Mass Storage Class

The MSC interface uses the native driver from USB host O.S. and does not require a specific setup. Thus, the microSD card is mounted as an external disk in O.S.

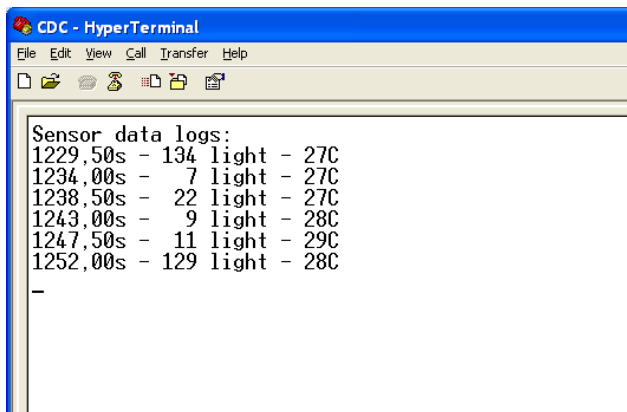
To avoid a file system corruption, the data logging is stopped on microSD while the USB is enabled. All sensor data logged on the microSD card are stored in a file named *dat_log_c3_xplained_xx.txt* where *xx* will be incremented from 00 for each new session of sensors data storage on the card.



1.3.2 USB Communication Device Class

The CDC interface uses the native driver from UNIX® O.S., but requires a specific one on Windows® O.S. When the USB feature is disabled, the Windows driver file is created by demo application on the microSD card. After USB MSC startup or through another SD card reader, the *atmel_devices_cdc.inf* file can be selected to install the new USB CDC interface.

After having installed the CDC driver, the Virtual COM port can be opened through a terminal.



Note: The Virtual COM port is not connected to a true RS232 COM port, thus the baud rate and parity parameters can be ignored.

1.4 Running the demo application

- Startup

The Atmel XMEGA-C3 Xplained kit is powered through the USB connector. The display first shows explanation of how to use the demonstration, SW1 pressed will skip this explanation.

- Sensors acquisitions running

The application starts a sensor acquisition task scheduled by the real-time counter (RTC). The temperature and light sensors values are displayed on the OLED. Pressing QTB0 increases sampling rate per 0.5s. Pressing QTB1 decreases sampling rate per 0.5s.

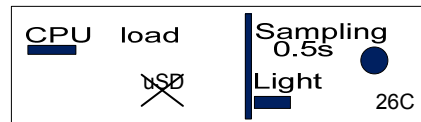


- microSD plug

When a microSD card is inserted, microSD icon is displayed. During microSD card installation, a rectangle shows the read/write access and the CPU load is increased consequently.



If the microSD installation fails, a cross is displayed on microSD icon.



After the file system mount, the CPU load falls and a read/write access is done sometimes when the application flushes the File System buffer on microSD.



- USB mode

Pressing SW0 activates USB, the USB icon is displayed.



When the USB Host opens the Virtual Port, the USB rectangle is displayed.



CPU activity varies according to demo modes. When USB is activated, display will show a CPU load light increase while USB Host accesses to microSD with multiple read/write operations will increase consequently this CPU load.

2. Inside the Demo Application

2.1 Overview

This demo application capitalizes on ASF modules available for the Atmel ATxmega384C3 device. [Figure 2-1](#) shows the specific application modules and the ASF modules used to build the demo application. All ASF modules are described in the [ASF online documentation](#).

Figure 2-1. XMEGA-C3 Xplained demo architecture.

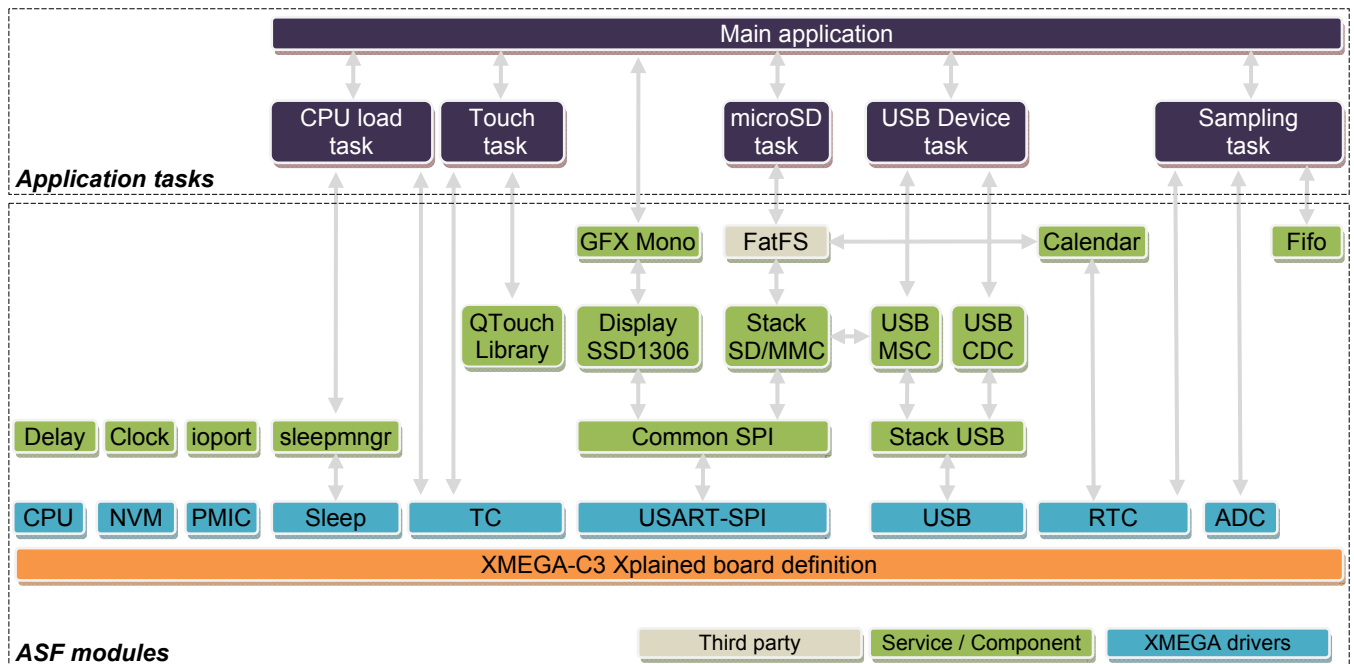


Table 2-1. Modules folder locations.

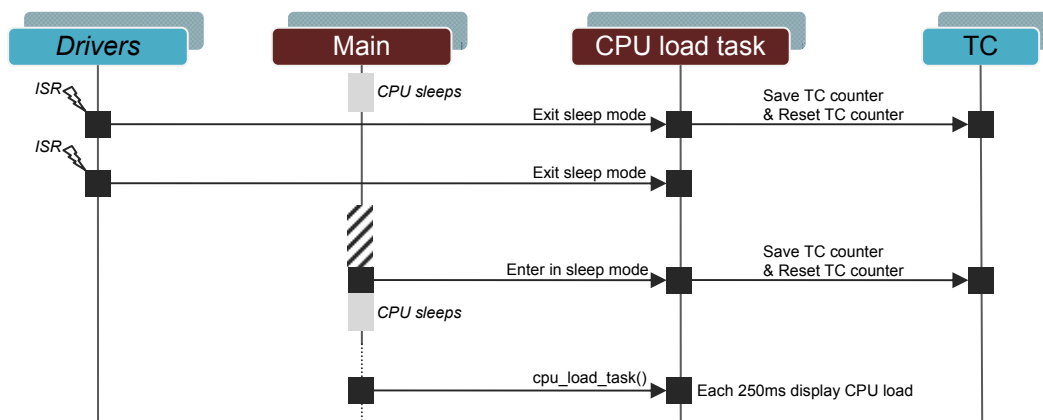
ASF module	Folder location
Application module files	xmega/applications/xmega_c3_xplained_demo/
FatFS third party	thirdparty/fatfs/
QTouch library	thirdparty/qtouch/generic/avr8/qtouch/
Common SPI service	common/services/spi/
USB Device service	common/services/usb/
USB CDC service	common/services/usb/class/cdc/
USB Mass storage service	common/services/usb/class/msc/
Sleep manager service	common/services/sleepmgr/
Ioport service	common/services/ioport/
GFX Mono service	common/services/gfx_mono/
Fifo service	common/services/fifo/
Delay service	common/services/delay/
Clock service	common/services/clock/

ASF module	Folder location
Calendar service	common/services/calendar/
SD/MMC stack component	common/components/memory/sd_mmc/
SSD1306 display component	common/components/display/ssd1306/
Board definitions	xmega/boards/xmega_c3_xplained/
CPU driver	xmega/drivers/cpu/
NVM driver	xmega/drivers/nvm/
PMIC driver	xmega/drivers/pmic/
Sleep driver	xmega/drivers/sleep/
ADC driver	xmega/drivers/adc/
RTC driver	xmega/drivers/rtc/
TC driver	xmega/drivers/tc/
USART-SPI driver	xmega/drivers/uart/
USB driver	xmega/drivers/usb/

2.2 CPU load task

The CPU load task monitors the CPU active time and CPU sleep time through a TC counter. The counter is saved and reset when the CPU enters and exits of the sleep mode. Each 250ms, the ratio is displayed through a bargraph on the OLED display. The implementation is available in *app_cpu_load.c/h* files.

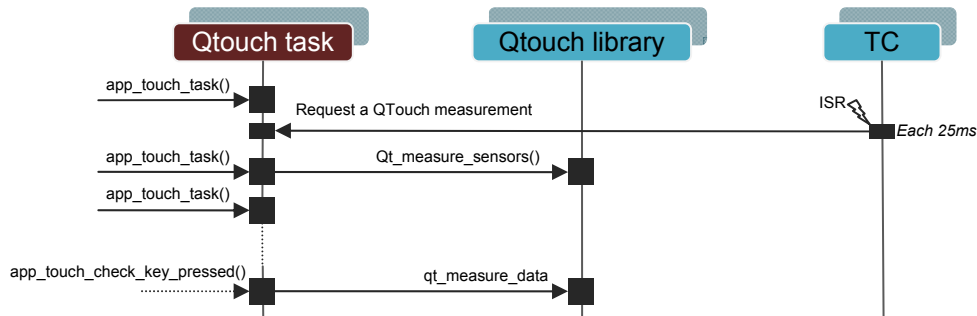
Figure 2-2. CPU load task behavior.



2.3 Atmel QTouch task

The QTouch is managed by a library, and the QTouch sensor measurement must be called each 25ms by the main task. In this demo application a TC counter is used to schedule it. The implementation is available in `app_touch.c/.h` files.

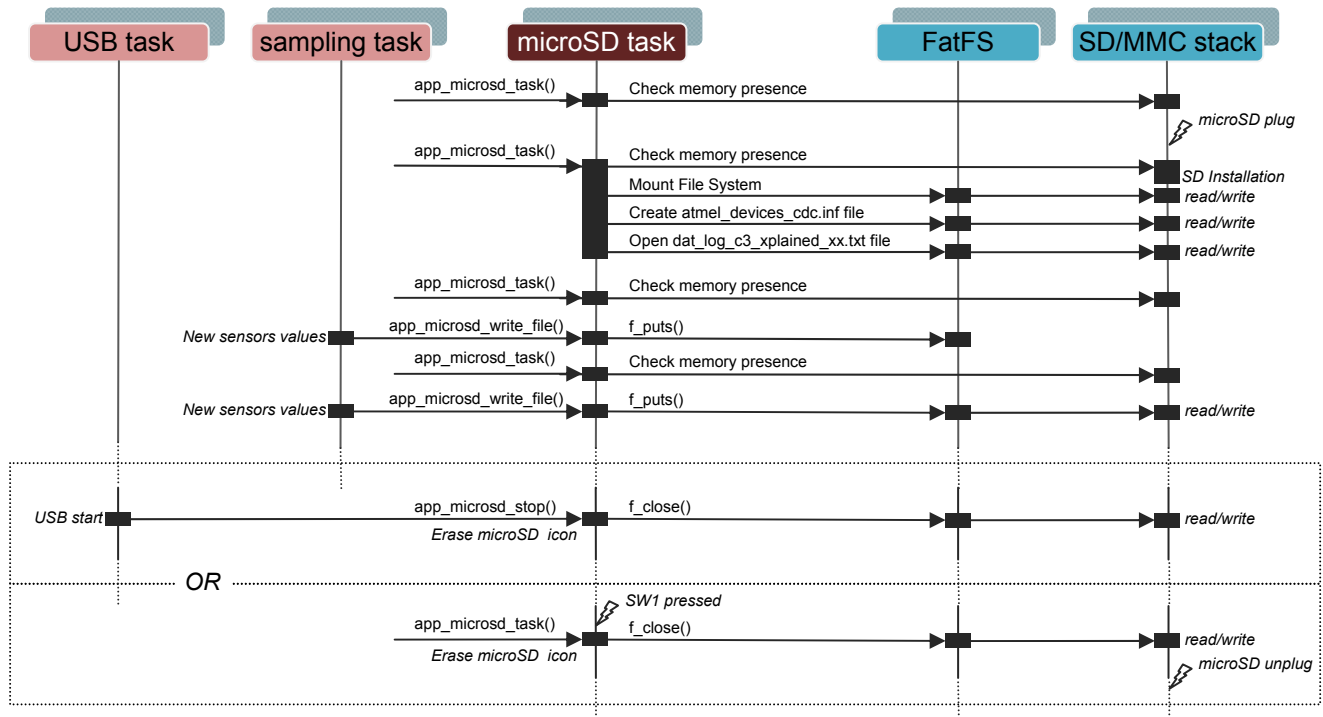
Figure 2-3. QTouch task behavior.



2.4 microSD task

The microSD detection, installation, mount, and file controls are performed in the microSD task. This task is enabled when the USB MSC is disabled, because an embedded file system (here FatFS) and a USB device MSC cannot run in same time on a same memory. The implementation is available in `app_microsd.c/.h` files.

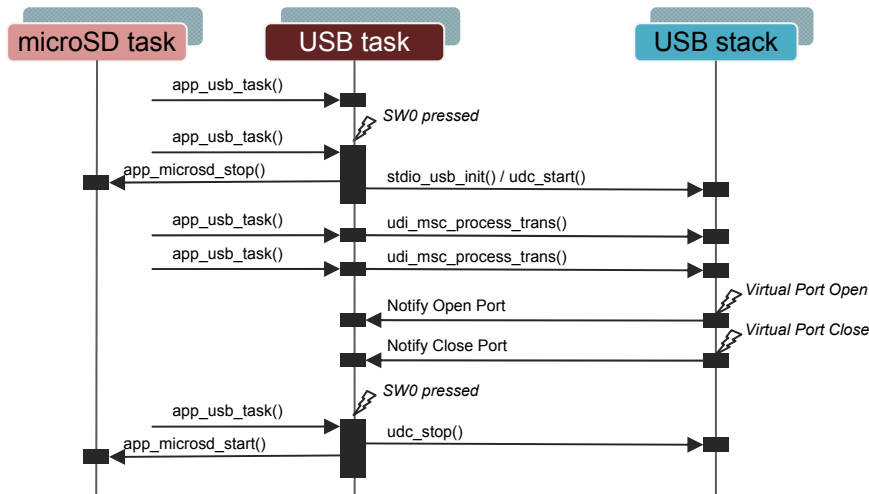
Figure 2-4. microSD task behavior.



2.5 USB task

The USB device stack and the CDC/MSC interfaces are interrupt driven and do not require a task except to perform a memory transfer outside interrupt routine. Also, the task checks the switch SW0 to enable and disable USB feature. The implementation is available in *app_usb.c/.h* files.

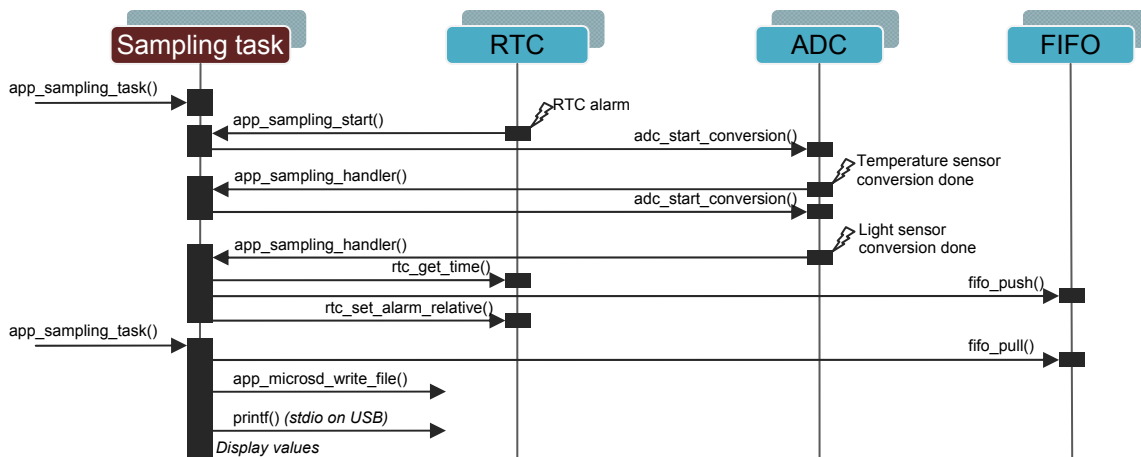
Figure 2-5. USB task behavior.



2.6 Sampling task

The schedule and the ADC conversions are done only by interrupt (RTC and ADC) to guarantee a constant acquisition. The FIFO service is used to save value, thus the sampling task can read FIFO, send values through USB, writes values on microSD and display values without timing constraint. The implementation is available in *app_sampling.c/.h* files.

Figure 2-6. Sampling task behavior.



2.7 Footprint

The figures below show the CODE and RAM spaces needed for each module used by the demo application.

Figure 2-7. Atmel XMEGA-C3 Xplained demo CODE footprint (unit Kbyte).

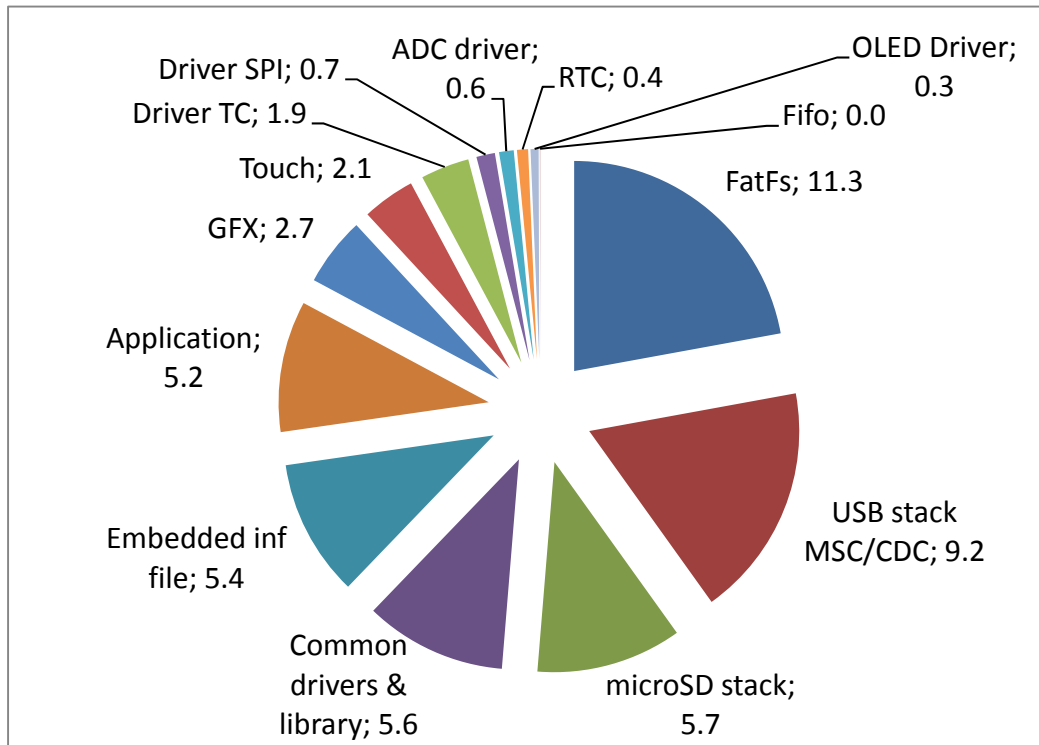
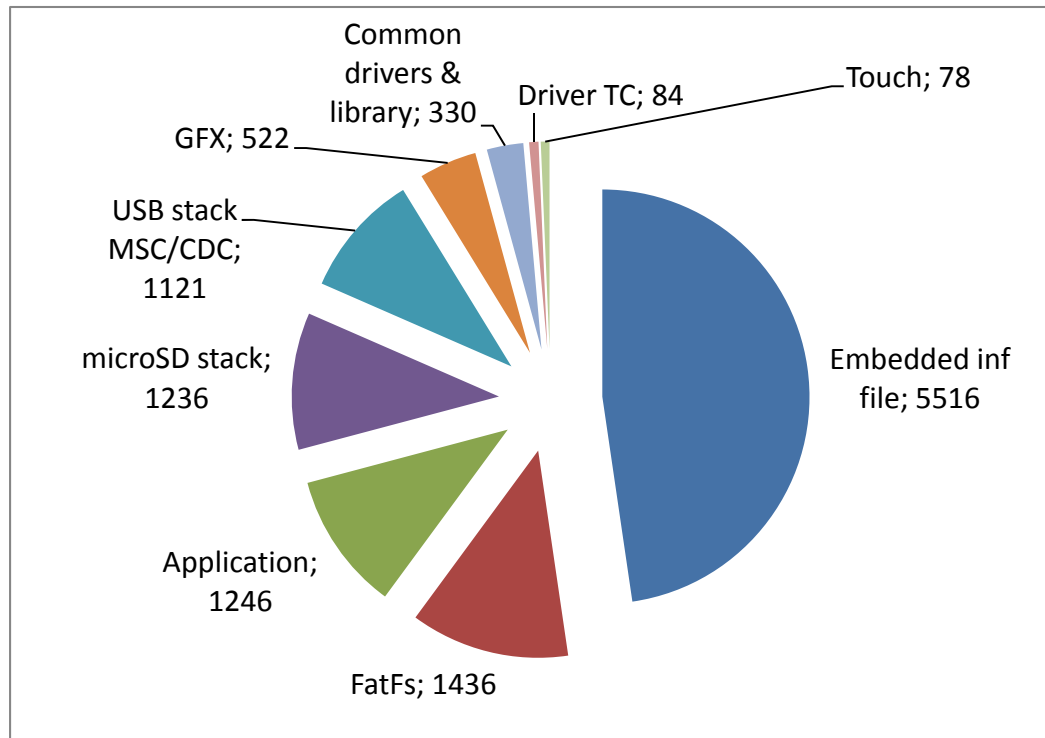


Figure 2-8. XMEGA-C3 Xplained demo RAM footprint (unit Byte).

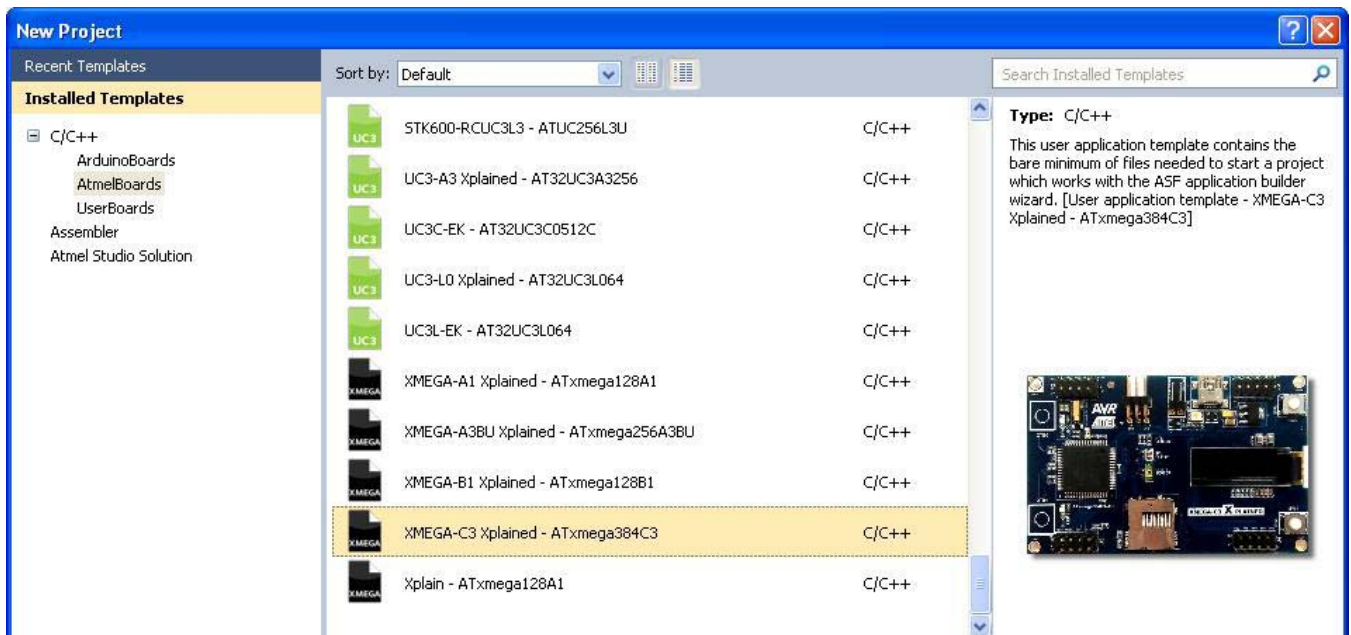


3. Building a New Application on XMEGA-C3 Xplained

The main ways to create a new application on XMEGA-C3 Xplained kit with Atmel Studio 6 and ASF are to create a “New Project” or to create a “New Example Project from ASF”.

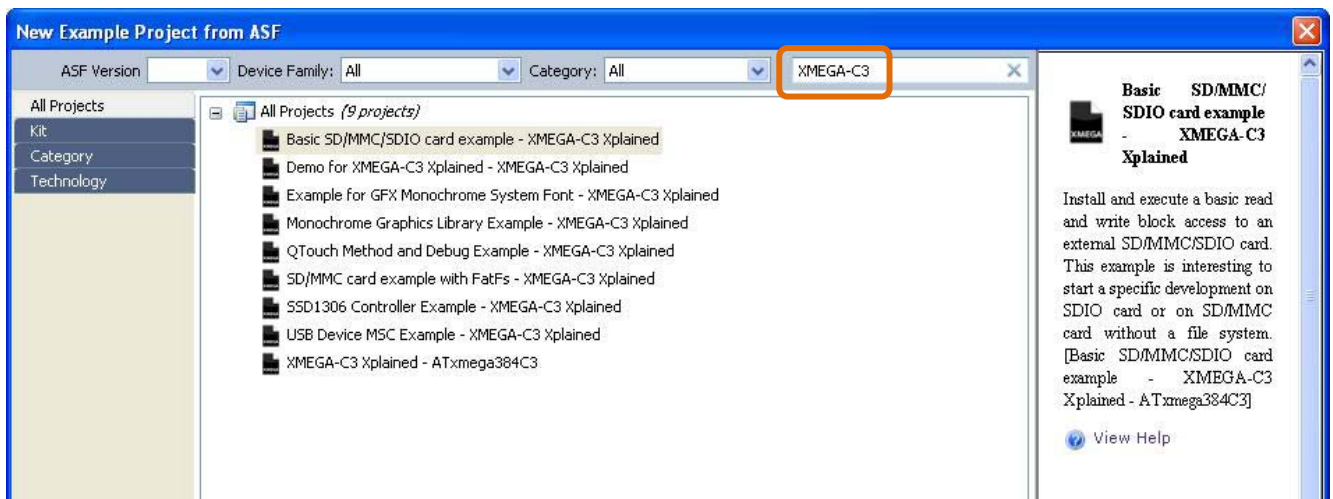
3.1 New project on XMEGA-C3 Xplained

When “New Project” is selected in Atmel Studio 6, the XMEGA-C3 Xplained kit is available in the Atmel board list. Thus, an empty project with the XMEGA-C3 Xplained kit definitions is created and allows adding several ASF modules (drivers, components and services) through *ASF Wizard*.



3.2 New example project from ASF on XMEGA-C3 Xplained

Several XMEGA-C3 Xplained examples are available through the [Atmel Software Framework \(ASF\)](#). Atmel Studio 6 includes ASF, and the examples are available as a new example project naming “XMEGA-C3 Xplained”. These examples can be used to start and build a new application.



4. References and Further Information

4.1 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.

4.2 Detailed hardware references (and associated errata)

More detailed hardware information for this kit can be found in the file XMEGA-C3 Xplained_Hardware-References.zip available on the Atmel web page dedicated to this kit: www.atmel.com/.

The Atmel XMEGA C family of devices is specified in the XMEGA C manual and device datasheet. Always use this document as a reference throughout the development life cycle of an application destined to run on a XMEGA C device.

4.3 Tools

To be able to develop applications for 8-bit Atmel AVR devices and build binaries for AVR targets and program an 8-bit AVR 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/studio>
- IAR Embedded Workbench[®]: IAR[™] Embedded Workbench with its optimizing C and C++ compiler provides full support and generates very compact and efficient code for AVR device
<http://www.iar.com/en/Products/IAR-Embedded-Workbench/AVR>

5. Revision History

Doc. Rev.	Date	Comments
42090A	02/2013	Initial document release

**Atmel Corporation**

1600 Technology Drive
San Jose, CA 95110
USA

Tel: (+1)(408) 441-0311

Fax: (+1)(408) 487-2600

www.atmel.com

Atmel Asia Limited

Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG

Tel: (+852) 2245-6100

Fax: (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY

Tel: (+49) 89-31970-0

Fax: (+49) 89-3194621

Atmel Japan G.K.

16F Shin-Osaki Kangyo Building
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN

Tel: (+81)(3) 6417-0300

Fax: (+81)(3) 6417-0370

© 2013 Atmel Corporation. All rights reserved. / Rev.: 42090A-AVR-02/2013

Atmel®, Atmel logo and combinations thereof, AVR®, Enabling Unlimited Possibilities®, QTouch®, XMEGA®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® is a registered trademark of Microsoft Corporation in U.S. and or other countries. ARM®, Cortex™ and others are 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.