APPLICATION NOTE

Atmel AT02657: XMEGA-E5 Xplained Software User Guide

Atmel AVR XMEGA E

Features

- OLED display with 128×32 pixels resolution
- Ambient light sensor
- CPU load
- Analog filter
- Quadrature Encoder with push button
- Digital I/O
  - Two mechanical buttons
  - Two user LEDs
  - Four expansion headers
- Board controller with USB interface
  - One power LED and one status LED

Description

The Atmel® AVR® XMEGA®-E5 Xplained evaluation kit demo software is created to showcase the Atmel AVR ATxmega32E5 device. The demo samples the light sensor connected on ADC and sent the values on the OLED display.

The demo is controlled through the mechanical quadrature encoder switch increasing or decreasing the ADC sampling rate of sensor acquisition.

This documentation describes the preloaded demo software and the software libraries available thru Atmel Software Framework (ASF) for XMEGA-E5 Xplained kit. The Atmel AT02667 application note describes the XMEGA-E5 Xplained hardware in detail.

Figure 1. The XMEGA-E5 Xplained kit.

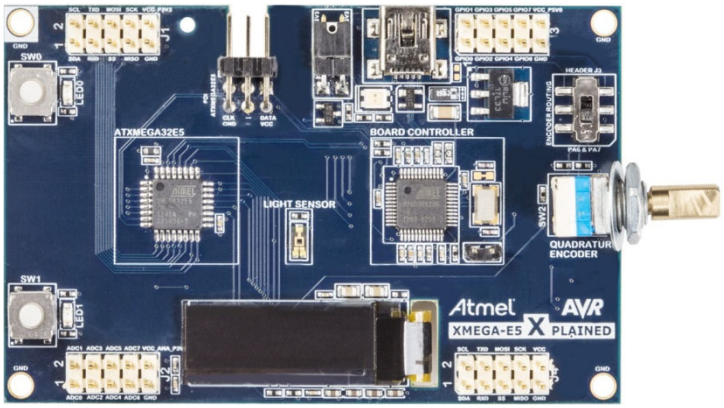


Table of Contents

1. Running Demo Application

1.1 Description

The Atmel XMEGA-E5 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-E5 Xplained”.

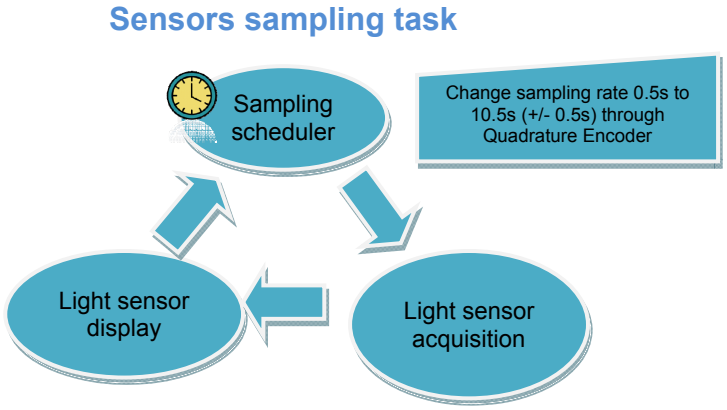
At power up, demo shows introduction screens, with explanation how to use the buttons and quadrature encoder. Then, the application starts with continuous sensor acquisition task scheduled with the real time counter (RTC). Light sensor values are displayed on the OLED. These values are also simultaneously sent to the PC using the USART to USB CDC gateway with the UC3B1 board controller.

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 two tasks:

1. Light sensor sampling task, managing the acquisition of the sensor.
2. CPU load task, displaying real time CPU activity (e.g. CPU time in active mode).

About Quadrature Encoder management, the application uses the XMEGA Quadrature Decoder feature based on Timer Counter and Event System hardware resources. Also, the XMEGA E device allows the rotary mode which is enabled for this application.



1.2 User interface

The Atmel XMEGA-E5 Xplained kit is powered through the USB connector. The demo application is controlled with the mechanical rotary encoder (SW1). The activity monitoring is done through the LEDs and the OLED display.

Figure 1-1. Hardware resources used in the XMEGA-E5 Xplained demo.

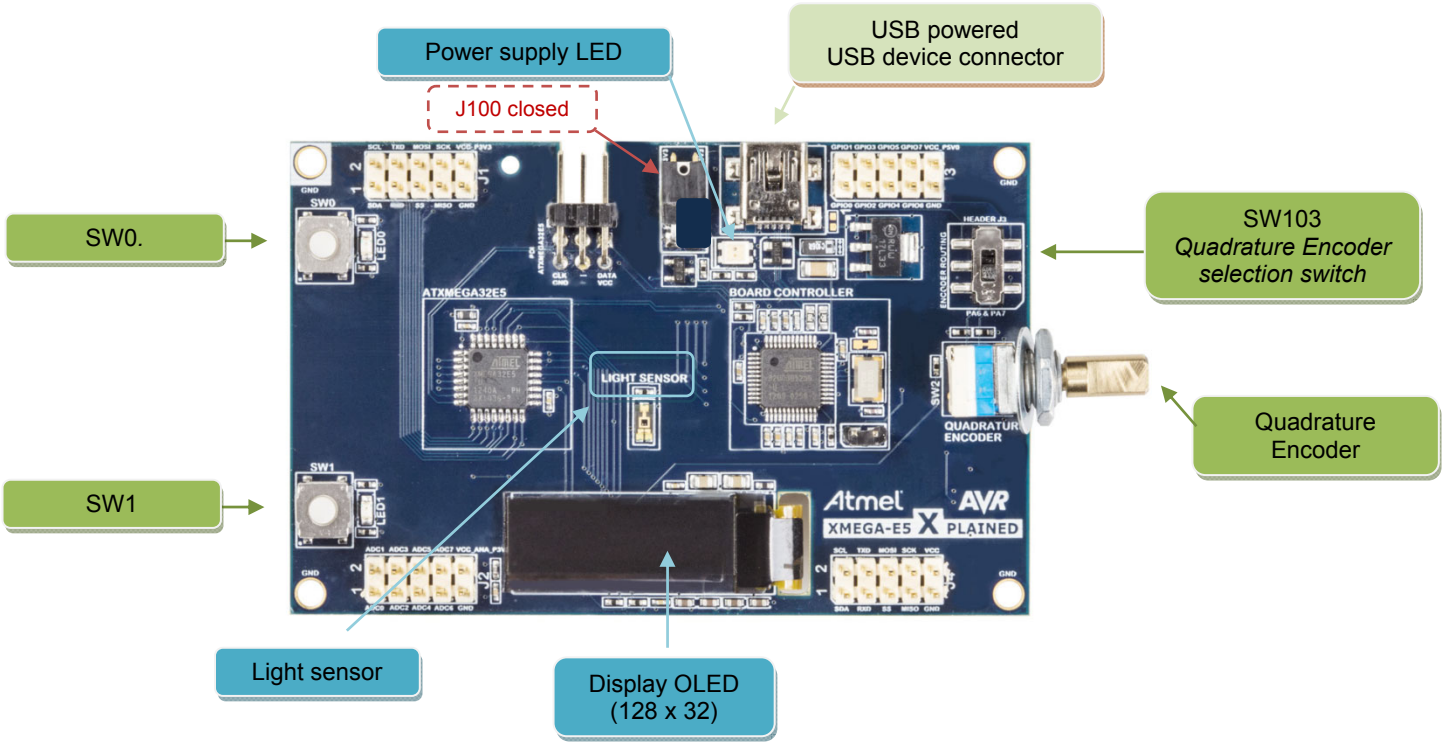
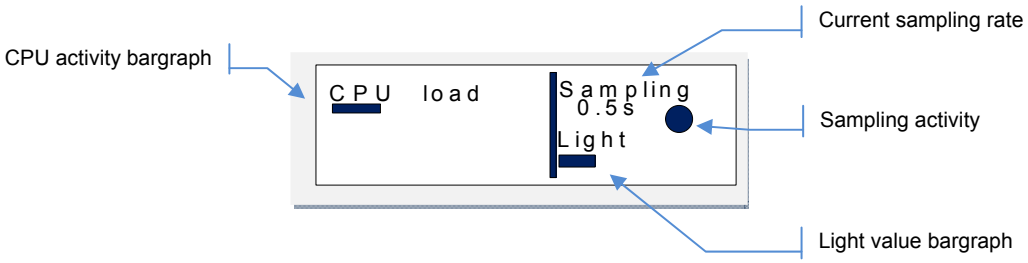


Figure 1-2. Information on the OLED display for the XMEGA-E5 Xplained demo.



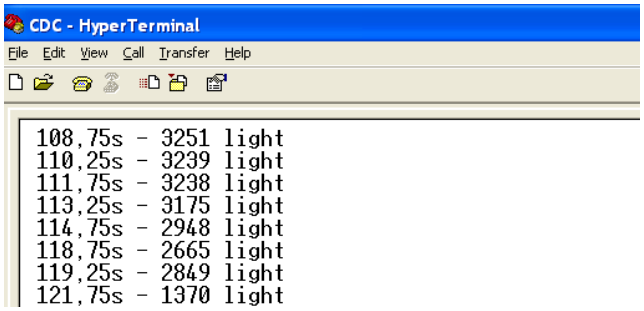
1.3 USB feature

When powering the kit through the USB connector, the Atmel XMEGA-E5 Xplained enumerates as a USB Communication Device Class (CDC). Actually this is the AT32UC3B1256 device, the board controller which does this USB-support by acting as a USART to USB CDC gateway.

1.3.1 USB Communication Device Class

The CDC interface uses the native driver from UNIX® O.S., but requires a specific one on Windows® O.S. available in the example. 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, however, the ATxmega32E5 device communicates with the AT32UC3B1 board controller on the USART at 38400 bauds, 1 stop bit, and no parity. So the terminal must be configured accordingly.

1.4 Running the demo application

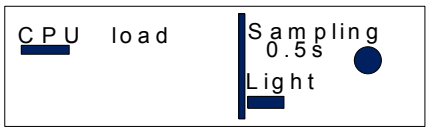
- Startup

The Atmel XMEGA-E5 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.

- Light sensor acquisition running

The application starts a sensor acquisition task scheduled by the real-time counter (RTC). The light sensor values are displayed on the OLED.

The Quadrature Encoder (SW2) is used to control the sampling rate. There are 20 positions to select a sampling rate from 0.5s to 10.5s per 0.5s step.



CPU activity varies according to actions on the Quadrature Encoder. When no change in the sampling rate, display shows a low CPU load while this will increase when changing this rate.

2. Inside the demo application

2.1 Overview

This demo application capitalizes on ASF modules available for the Atmel ATxmega32E5 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-E5 Xplained demo architecture.

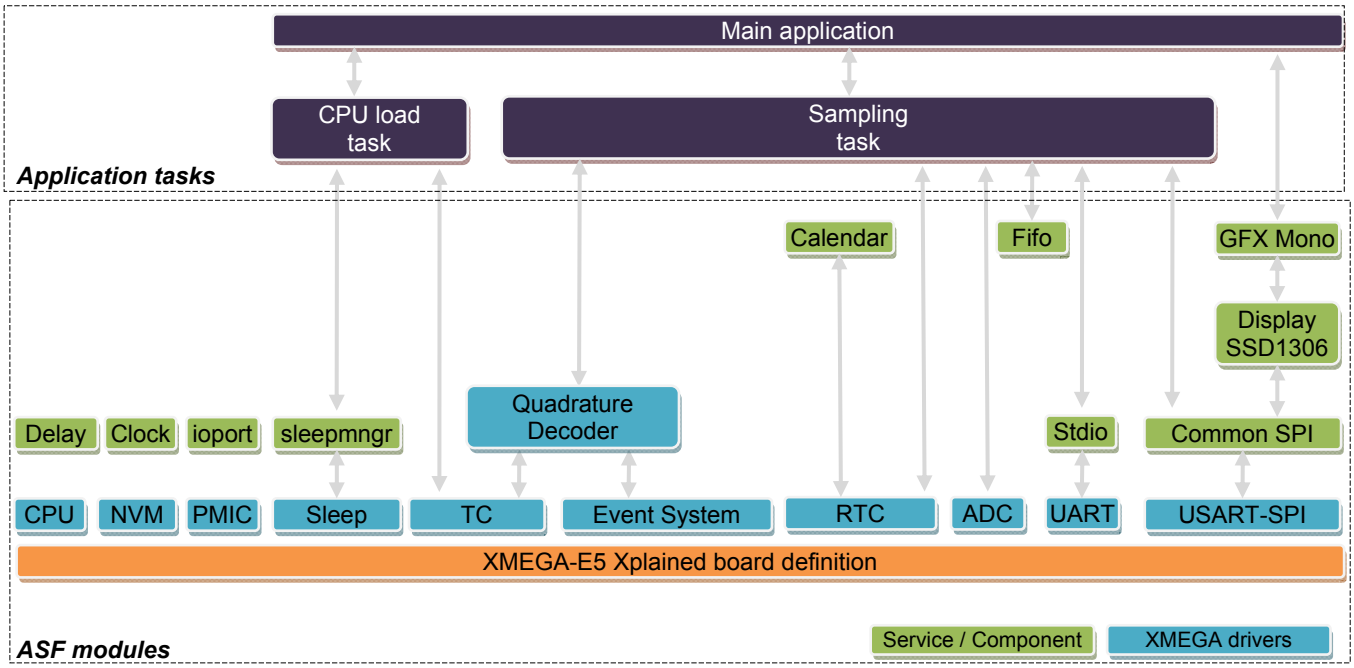


Table 2-1. Modules folder locations.

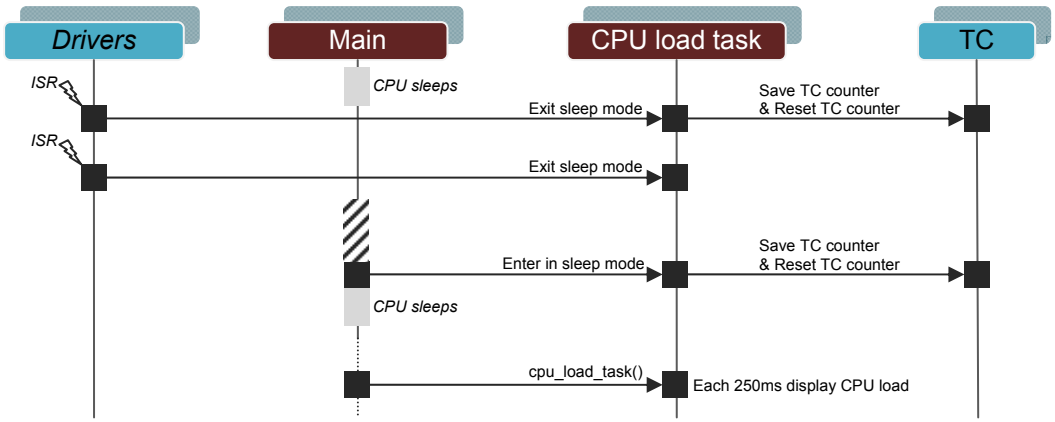
ASF module	Folder location
Application module files	**xmega/applications/xmega_e5_xplained_demo/**
Common SPI service	common/services/spi/
Sleep manager service	common/services/sleepmgr/
Ioport service	common/services/ioport/
GFX Mono service	common/services/gfx_mono/
Fifo service	common/services/fifo/
Delay service	common/services/delay/
Clock service	common/services/clock/
Calendar service	common/services/calendar/
SSD1306 display component	common/components/display/ssd1306/
Board definitions	xmega/boards/xmega_e5_xplained/
CPU driver	xmega/drivers/cpu/

ASF module	Folder location
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/tc45/
QDec driver	xmega/drivers/qdec/
USART-SPI driver	xmega/drivers/usart/

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.

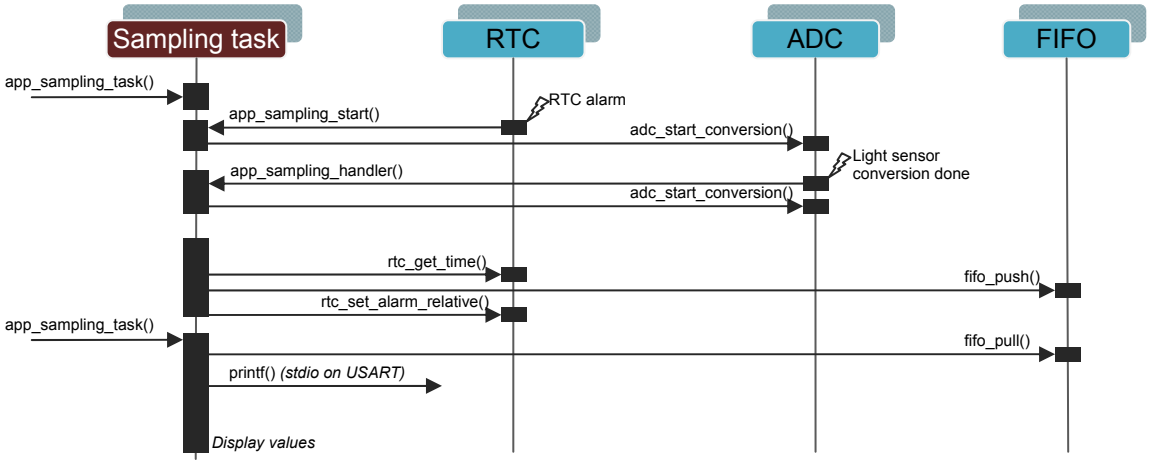
Figure 2-2. CPU load task behavior.



2.3 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 USART and display values without timing constraint. The implementation is available in *app_sampling.c/.h* files.

Figure 2-3. Sampling task behavior.



2.4 Footprint

Figure 2-4 and Figure 2-5 show the CODE and RAM spaces needed for each module used by the demo application.

Figure 2-4. Atmel XMEGA-E5 Xplained demo CODE footprint (unit Kbyte).

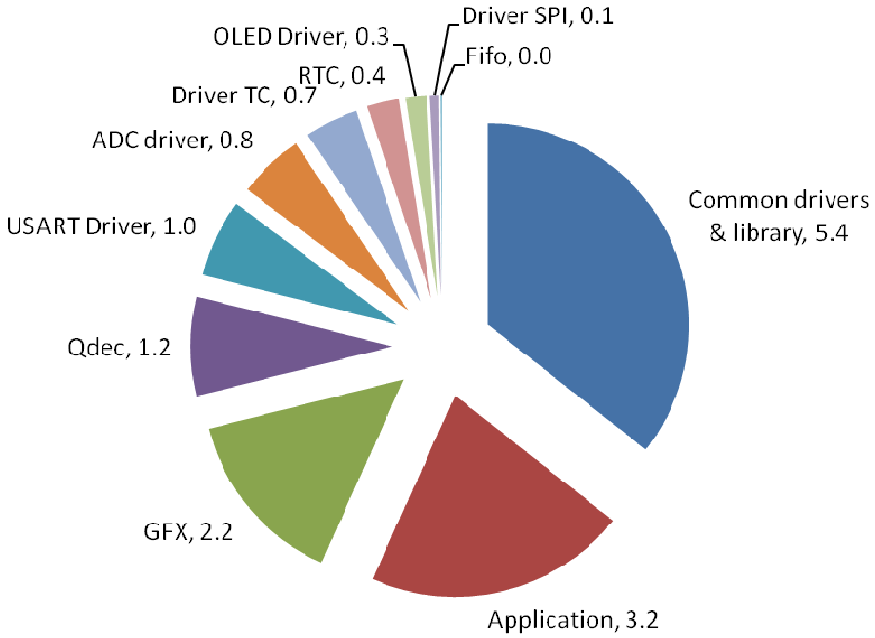
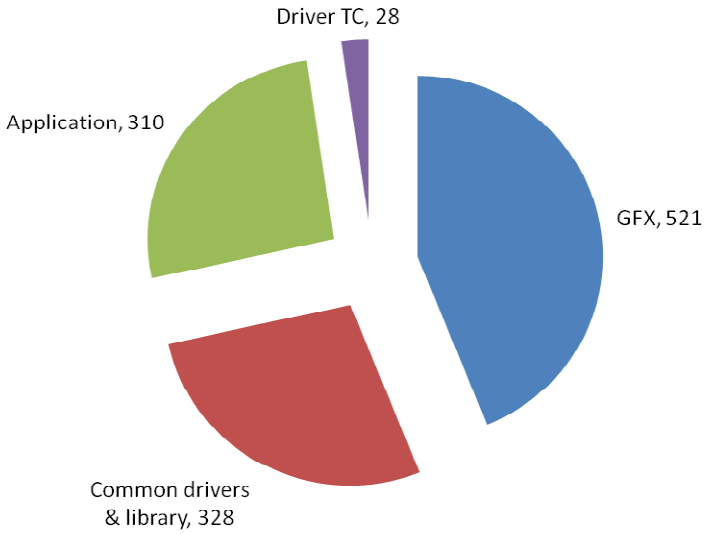


Figure 2-5. XMEGA-E5 Xplained demo RAM footprint (unit Byte).

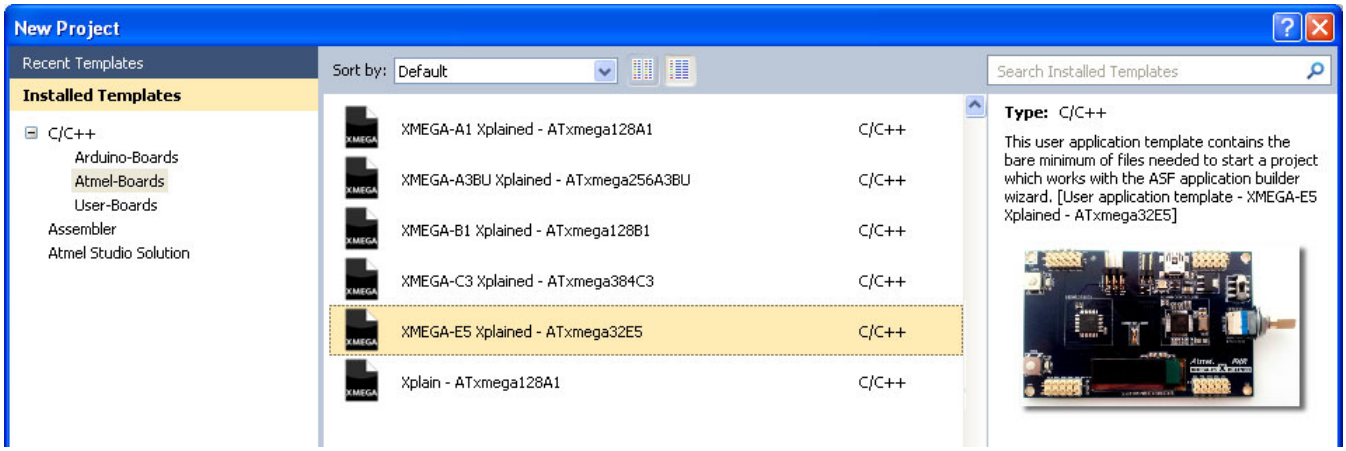


3. Building a New Application on the XMEGA-E5 Xplained

The main ways to create a new application on XMEGA-E5 Xplained kit with Atmel Studio 6 and ASF are to create a “New Project” or to create a “New Example Project from ASF”.

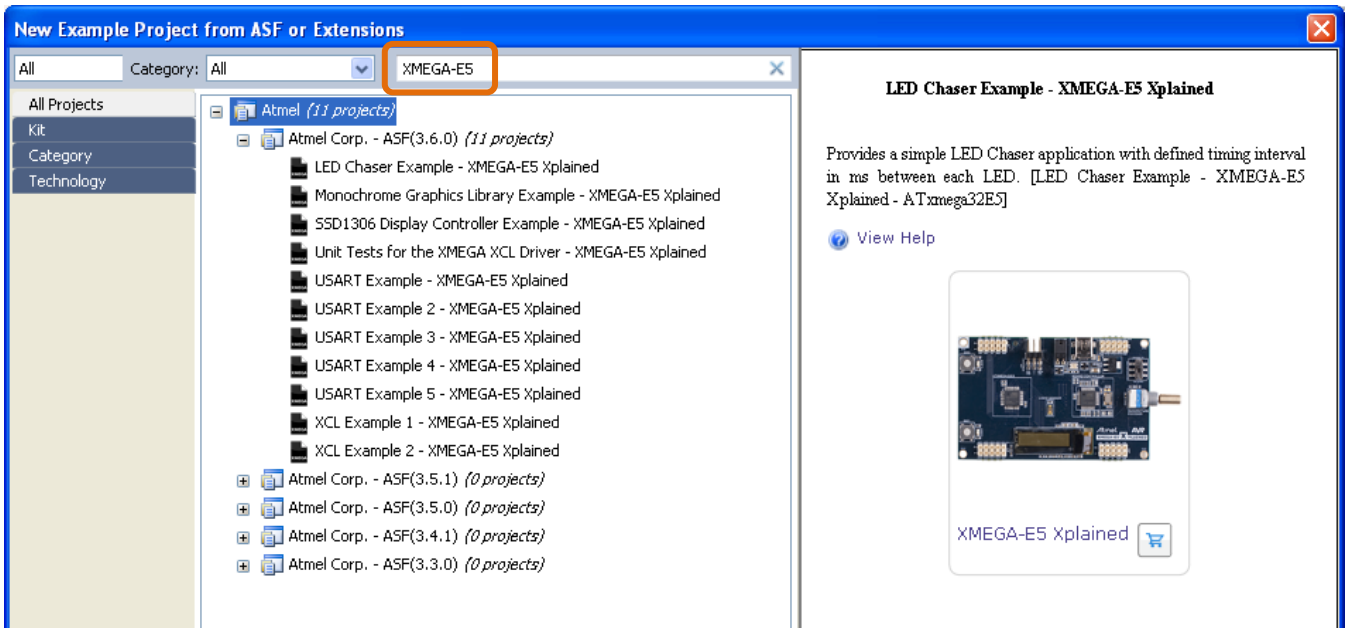
3.1 New project on XMEGA-E5 Xplained

When “New Project” is selected in Atmel Studio 6, the XMEGA-E5 Xplained kit is available in the Atmel board list. Thus, an empty project with the XMEGA-E5 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-E5 Xplained

Several XMEGA-E5 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-E5 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-E5 Xplained_Hardware-References.zip available on the Atmel web page dedicated to this kit: www.atmel.com/.

The Atmel XMEGA E family of devices is specified in the XMEGA E 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 E 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
42085A	04/2013	Initial document release

Atmel | Enabling Unlimited Possibilities®

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
Parkring 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