

### Atmel AVR32931: UC3-L0 Xplained Getting Started Guide

#### Atmel AVR 32-bit Microcontroller

#### Description

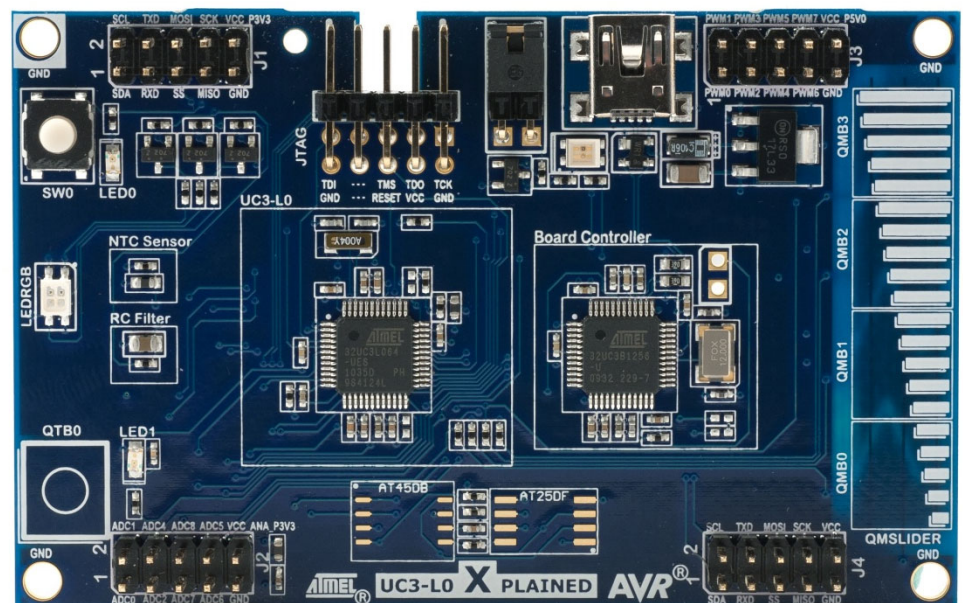
The [Atmel® UC3-L0 Xplained](#) evaluation kit is a hardware platform for evaluating the [Atmel AT32UC3L064](#) microcontroller.

The kit offers a range of features that enable the Atmel AVR® UC3 user to get started using UC3 peripherals right away and understand how to integrate the UC3 device in their own design. This getting started guide describes the basic steps needed to start using the UC3-L0 [Xplained evaluation kit](#).

#### Features

- Easy to reprogram with just a USB cable and a preprogrammed bootloader
- Easy to debug code with JTAG-based debugger and emulator
- Can be used with Atmel AVR JTAGICE mkII, JTAGICE3, Atmel AVR Dragon™, and Atmel AVR ONE!

**Figure 1. UC3-L0 Xplained Evaluation Kit.**



## Table of Contents

1. What is needed to get started? .....	3
1.1 Documentation for UC3-L0 Xplained.....	3
1.2 Atmel Software Framework.....	3
1.3 Integrated Development Environment (IDE) for Atmel AVR 32-bit Microcontrollers.....	3
1.3.1 Atmel Studio .....	3
1.3.2 IAR Embedded Workbench for Atmel AVR 32-bit Devices .....	3
1.4 FLIP (BatchISP) .....	4
1.5 Toolchain .....	4
1.5.1 Atmel AVR32 GNU Toolchain.....	4
1.5.2 IAR AVR Toolchain.....	4
1.6 Programmers and Debuggers.....	4
1.6.1 Atmel AVR Dragon .....	4
1.6.2 Atmel AVR JTAGICE mkII .....	4
1.6.3 Atmel AVR JTAGICE3 .....	5
1.6.4 Atmel AVR ONE! .....	5
2. Board Overview .....	6
3. Connecting the Board .....	7
3.1 Connecting the Atmel AVR Dragon Kit.....	7
3.2 Connecting the Atmel AVR JTAGICE mkII.....	7
3.3 Connecting the Atmel AVR JTAGICE3 .....	8
3.4 Connecting the Atmel AVR ONE!.....	9
4. Programming the Kit .....	10
4.1 Programming via the Bootloader.....	10
4.1.1 Prerequisites.....	10
4.1.2 Step-by-step Guide.....	10
4.2 Restoring the Bootloader .....	11
4.3 Programming via Atmel Studio.....	11
5. Example Applications.....	13
5.1 Preprogrammed Firmware .....	13
5.1.1 Functionality of the Preprogrammed Firmware .....	13
5.1.1.1 Capacitive Touch Module.....	13
5.1.1.2 Pulse Width Modulation Controller.....	13
5.1.1.3 UART .....	13
5.2 More example projects.....	13
6. How to Compile Example Projects.....	14
6.1 AVR GNU Toolchain .....	14
6.2 IAR Embedded Workbench.....	14
7. References.....	15
7.1 IAR Compiler.....	15
7.2 Atmel Software Framework.....	15
8. EVALUATION BOARD/KIT IMPORTANT NOTICE .....	16
9. Revision History .....	17

## 1. What is needed to get started?

To be able to run the preprogrammed code, the [Atmel UC3-L0 Xplained](#) evaluation kit and a USB cable (Standard-A to Mini-B, or Mini-AB) are needed to connect the board to a PC.

When connecting the [UC3-L0 Xplained](#) to a PC, the operating system will request a driver file for installing the serial communication driver. This driver file is available in the zip file associated with this document. The driver file supports both 32- and 64-bit versions of the Windows® XP and Windows 7 operating systems. Driver installs are not necessary on Linux® operating systems.

To modify the example code or write new code and compile it, a toolchain for Atmel AVR microcontrollers and an IDE are needed to edit and debug code. Atmel provides [Atmel Studio](#) for this purpose. Other IDEs and tool chains are also available, such as the [IAR Embedded Workbench®](#).

To debug the code at runtime, a debugger, such as [Atmel JTAGICE3](#) or [Atmel AVR ONE!](#), will also be needed. More information on these tools is available from the [Atmel website](#). See Chapter 7 [References](#) in this document for more details.

### 1.1 Documentation for UC3-L0 Xplained

The AVR32924: UC3-L0 Xplained Hardware User's Guide package includes in-depth information about the kit, such as schematics, BOM, assembly drawings, 3D plots, and more. The package is available from the Atmel website:

<http://www.atmel.com/tools/UC3-L0XPLAINED.aspx>

### 1.2 Atmel Software Framework

The [Atmel Software Framework \(ASF\)](#) provides software drivers and libraries to build applications for Atmel AVR devices. It has been designed to help develop and merge the different components of a software design. It can easily be integrated into an operating system (OS) or run as a standalone product.

The ASF (latest version 3.10) that integrates with [Atmel Studio](#) can be found here:

<http://www.atmel.com/tools/AVRSOFTWAREFRAMEWORK.aspx>

### 1.3 Integrated Development Environment (IDE) for Atmel AVR 32-bit Microcontrollers

#### 1.3.1 Atmel Studio

The [Atmel Studio](#) is a free IDE for developing applications for Atmel AVR 8- and 32-bit Microcontrollers. It offers an advanced editor as well as programming and debugging support.

**Atmel Studio** (free Atmel 8- and 32-bit IDE):

<http://www.atmel.com/tools/ATMELSTUDIO.aspx>

To learn how to use this tool, take a look at the Release Notes section.

#### 1.3.2 IAR Embedded Workbench for Atmel AVR 32-bit Devices

[IAR Embedded Workbench](#) is an integrated development environment with project management tools and editor.

**IAR Embedded Workbench for Atmel AVR 32-bit Microcontrollers:**

<http://www.iar.com/en/Products/IAR-Embedded-Workbench/AVR32/>

## 1.4 FLIP (BatchISP)

BatchISP is a command-line tool for programming the flash and EEPROM memories of the board controller, and is part of the [FLIP](#) installation. It can be used to communicate with the preprogrammed bootloader on the board controller.

**FLIP** (FLexible In-system Programmer) for programming:

<http://www.atmel.com/tools/FLIP.aspx>

## 1.5 Toolchain

In order to build an application from the source code, a toolchain is needed, which includes required tools such as an assembler, compiler, and linker. The code for this kit is compatible with the [Atmel AVR32 GNU](#) and [IAR™ AVR](#) toolchains.

### 1.5.1 Atmel AVR32 GNU Toolchain

The [AVR32 GNU Toolchain](#) is a free toolchain that is integrated with [Atmel Studio](#) from the [Atmel website](#). It is also available as a separate installer. It includes all the necessary tools, such as a C/C++ compiler, an assembler, and a linker, for AVR application development. The toolchain is also available for the Linux OS. For more information about the supported distributions, see the release notes on the Atmel website.

### 1.5.2 IAR AVR Toolchain

The [IAR AVR Toolchain](#) is a commercial toolchain that includes all the necessary tools, such as a C/C++ compiler, an assembler, and a linker, for AVR application development. A 30-day evaluation version as well as a 32k (code size limited) kick-start version is available from the [IAR website](#).

IAR Embedded Workbench for Atmel AVR 32-bit Microcontrollers:

<http://www.iar.com/en/Products/IAR-Embedded-Workbench/AVR32/>

## 1.6 Programmers and Debuggers

Programmers and debuggers are needed when either application programming occurs without using the bootloader or debugging support is desired. Several different tools are available for use directly with the Xplained kit, and the most common of these are described in the following sections.

### 1.6.1 Atmel AVR Dragon

The low-cost [Atmel AVR Dragon](#) debug/programming tool can be used for code development with the [UC3-L0 Xplained](#) kit.

Atmel AVR Dragon (on-chip programming and debugging tool)

<http://www.atmel.com/tools/avrdragon.aspx>

### 1.6.2 Atmel AVR JTAGICE mkII

The [Atmel AVR JTAGICE mkII](#) is a mid-range debug/programming tool that can be used for code development with the [UC3-L0 Xplained](#) kit.

**Atmel AVR JTAGICE mkII** (on-chip programming and debugging tool)

<http://www.atmel.com/tools/avrjtagicemkii.aspx>

### 1.6.3 Atmel AVR JTAGICE3

The [Atmel AVR JTAGICE3](#) is a mid-range debug/programming tool that can be used for code development with the [UC3-L0 Xplained](#) kit.

**Atmel AVR JTAGICE3** (on-chip programming and debugging tool)

<http://www.atmel.com/tools/JTAGICE3.aspx>

### 1.6.4 Atmel AVR ONE!

The high-end [Atmel AVR ONE!](#) debug/programming tool can be used for code development with the [Atmel UC3-L0 Xplained](#) kit.

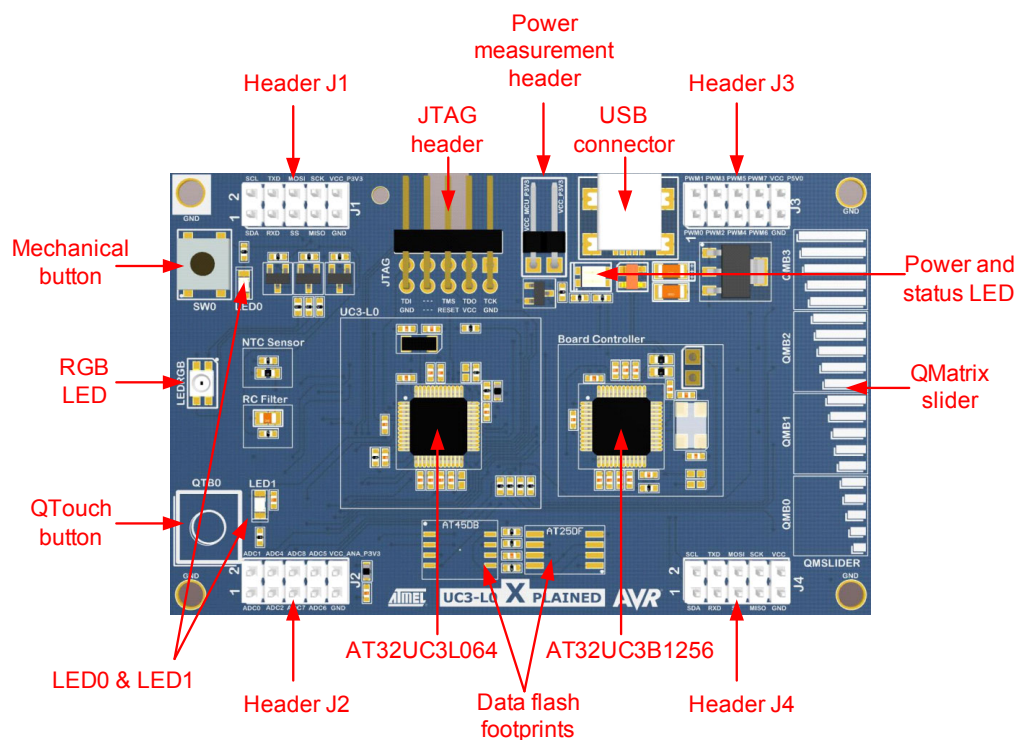
**Atmel AVR ONE!** (on-chip programming and debugging tool)

[http://www.atmel.com/tools/AVRONE\\_.aspx](http://www.atmel.com/tools/AVRONE_.aspx)

## 2. Board Overview

Figure 2-1 shows most of the available features on the UC3-L0 Xplained board. For more detailed information on the Atmel UC3-L0 Xplained hardware, take a look at the Atmel application note, [AVR32924: UC3-L0 XPLAINED Hardware User's Guide](#).

Figure 2-1. Overview of the UC3-Lo Xplained Kit



### 3. Connecting the Board

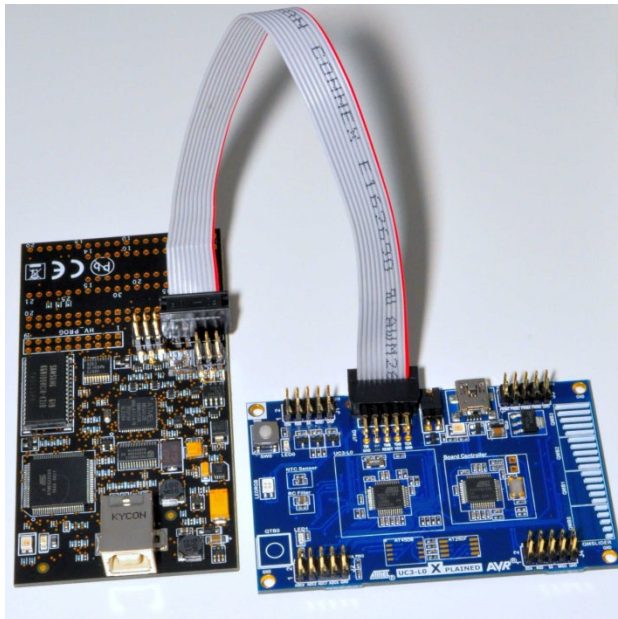
Connect a USB cable (Standard-A to Mini-B, or Mini-AB) between the board and a PC or a USB power supply to power it up. That is all that is needed. When power is applied, the power/status LED will light up green, and the RGB LED will light up blue.

**Warning:** Do not power the board without having a jumper installed or an ammeter mounted across the pins of the power measurement header located next to the USB connector. Otherwise, the device may be damaged by power sourcing through the I/O pins. The power measurement header location is shown in [Figure 2-1](#).

#### 3.1 Connecting the Atmel AVR Dragon Kit

A 10-pin header is needed to connect the [AVR Dragon](#) kit to the [UC3-L0 Xplained](#) board. Connect the cable between the JTAG connector on the [AVR Dragon](#) kit and the JTAG connector on the [UC3-L0 Xplained](#) board. [Figure 3-1](#) shows how the connection should be made.

**Figure 3-1. Connecting the AVR Dragon Kit to the UC3-L0 Xplained Board**



#### 3.2 Connecting the Atmel AVR JTAGICE mkII

The grey connector on the [AVR JTAGICE mkII](#) probe has to be used when connecting to the [UC3-L0 Xplained](#) board. [Figure 3-2](#) shows how to make the connection. The notch of the grey connector must be placed into the cutout of the board.



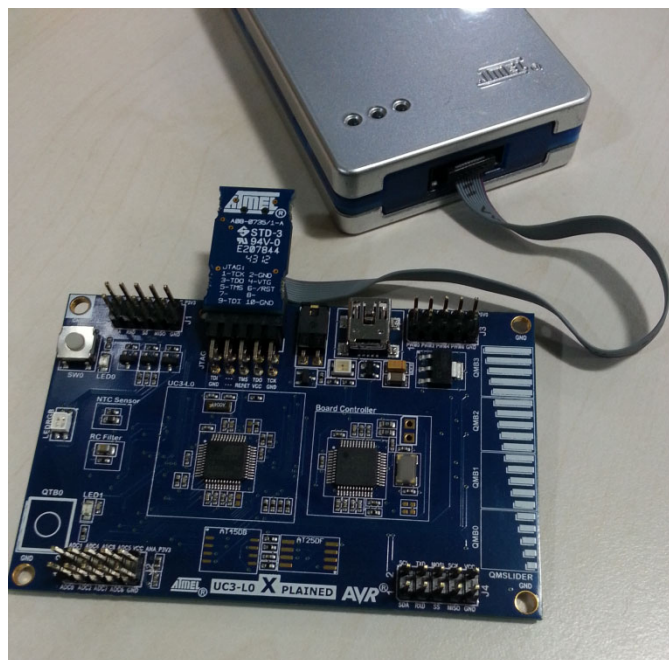
Figure 3-2. Connecting the Atmel AVR JTAGICE mkII to the Atmel UC3-L0 Xplained Board



### 3.3 Connecting the Atmel AVR JTAGICE3

The grey connector on the [AVR JTAGICE3](#) probe has to be used when connecting to the [UC3-L0 Xplained](#) board. [Figure 3-3](#) shows how the connections should be made. It requires a 50mil to 100mil adapter to connect the JTAGICE3 to the UC3L0 Xplained Board.

Figure 3-3. Connecting the Atmel AVR JTAGICE3 to the Atmel UC3-L0 Xplained Board





### 3.4 Connecting the Atmel AVR ONE!

The AVR ONE! JTAG connector can be connected to the UC3-L0 Xplained board with a standoff adapter, as shown in Figure 3-4.

Figure 3-4. Connecting the AVR ONE! to the UC3-L0 Xplained Board



## 4. Programming the Kit

### 4.1 Programming via the Bootloader

The preprogrammed bootloader on the [Atmel AT32UC3L064](#) makes it possible to program the device via the UART interface. Because the default board controller firmware acts as UART-to-USB bridge, the [AT32UC3L064](#) can be programmed from the PC via the USB connector.

The [AT32UC3L064](#) boot loader is started by pushing the mechanical switch (SW0) during reset. Programming can be performed by using the BatchISP command-line tool from the [FLIP](#) installation.

#### 4.1.1 Prerequisites

To program the device, BatchISP is needed, and it can be found under the [FLIP](#) installation directory. Install [FLIP](#) version 3.4.2 or later.

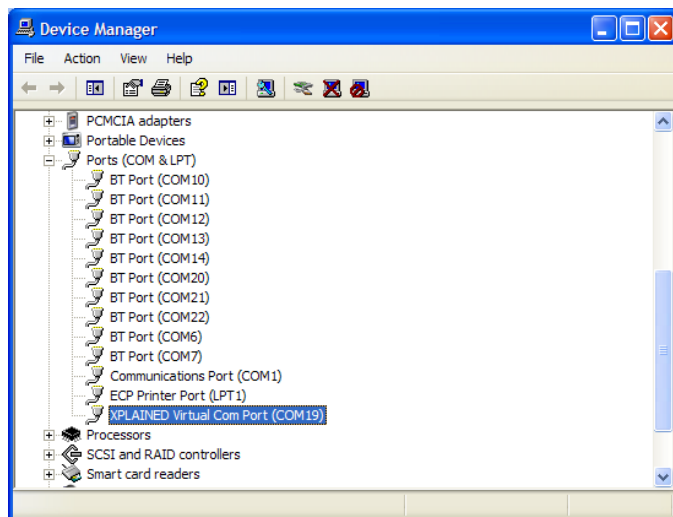
The UART-USB gateway in the board controller must be intact to program the new firmware into the device using BatchISP.

#### 4.1.2 Step-by-step Guide

To program the device, follow these steps:

1. Disconnect the USB cable from the kit.
2. Press and hold the button (SW0) while re-attaching the USB cable.
3. Open the device manager and locate the COM port used by the kit:

**Figure 4-1. Xplained Kit COM Port**



4. Run the command below with the correct COM port:

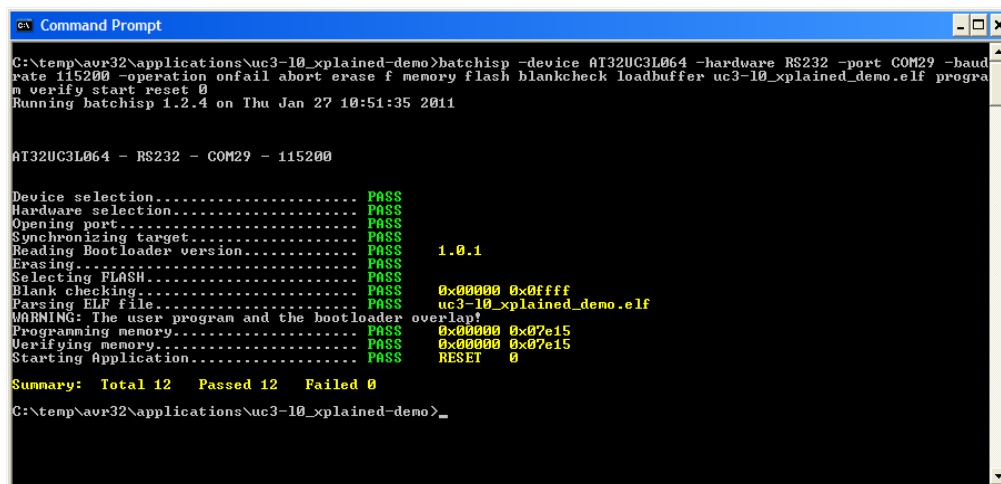
```
Batchisp    -device AT32UC3L064
            -hardware RS232
            -port COM19
            -baudrate 115200
            -operation onfail abort
            memory flash
            erase f
            blankcheck
            loadbuffer your_firmware_file.hex
            program verify
            start reset 0
```

5. The firmware should now be running.

**Note:** The bootloader can only be synchronized once after power up, which means it is possible to connect to the target only once, using BatchISP. If everything was not able to be done on the first try, power cycle the board and run the command again.

An example of programming the device via the bootloader can be seen in [Figure 4-2](#). Note that the output gives a warning about the user program and the bootloader overlap. This happens because the bootloader resides in the start of the flash section, which is the same area as the trampoline area in which the user code is located. As the bootloader is protected, the bootloader will not be overwritten.

**Figure 4-2. Example of Programming with BatchISP**



```
Command Prompt
C:\temp\avr32\applications\uc3-10_xplained-demo>batchisp -device AT32UC3L064 -hardware RS232 -port COM29 -baud
rate 115200 -operation onfail abort erase f memory flash blankcheck loadbuffer uc3-10_xplained_demo.elf progra
m verify start reset 0
Running batchisp 1.2.4 on Thu Jan 27 10:51:35 2011

AT32UC3L064 - RS232 - COM29 - 115200

Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Synchronizing target..... PASS
Reading Bootloader version..... PASS 1.0.1
Erasing..... PASS
Selecting FLASH..... PASS
Blank checking..... PASS
Parsing ELF file..... PASS uc3-10_xplained_demo.elf
WARNING: The user program and the bootloader overlap!
Programming memory..... PASS 0x00000 0x07e15
Verifying memory..... PASS 0x00000 0x07e15
Starting Application..... PASS RESET 0

Summary: Total 12 Passed 12 Failed 0
C:\temp\avr32\applications\uc3-10_xplained-demo>
```

For more information about the BatchISP command line tool, run `batchisp -h` from a command line to get a list of options and parameters. Read the help file for [FLIP](#) for more detailed information.

The bootloader supports reading and writing of the following memories:

- Flash (0x0000 - 0xFFFF, application section)
- User (0x000 - 0x1FF, user signature row)

## 4.2 Restoring the Bootloader

When the bootloader has been erased, it can be restored by using any of the previously mentioned programming tools.

The source for the UART bootloader is part of ASF 2.2.x and later, and can be downloaded and programmed into the device to restore the bootloader functionality, if the bootloader was accidentally erased.

**Note:** The bootloader section is protected, and can only be erased by using an external programmer.

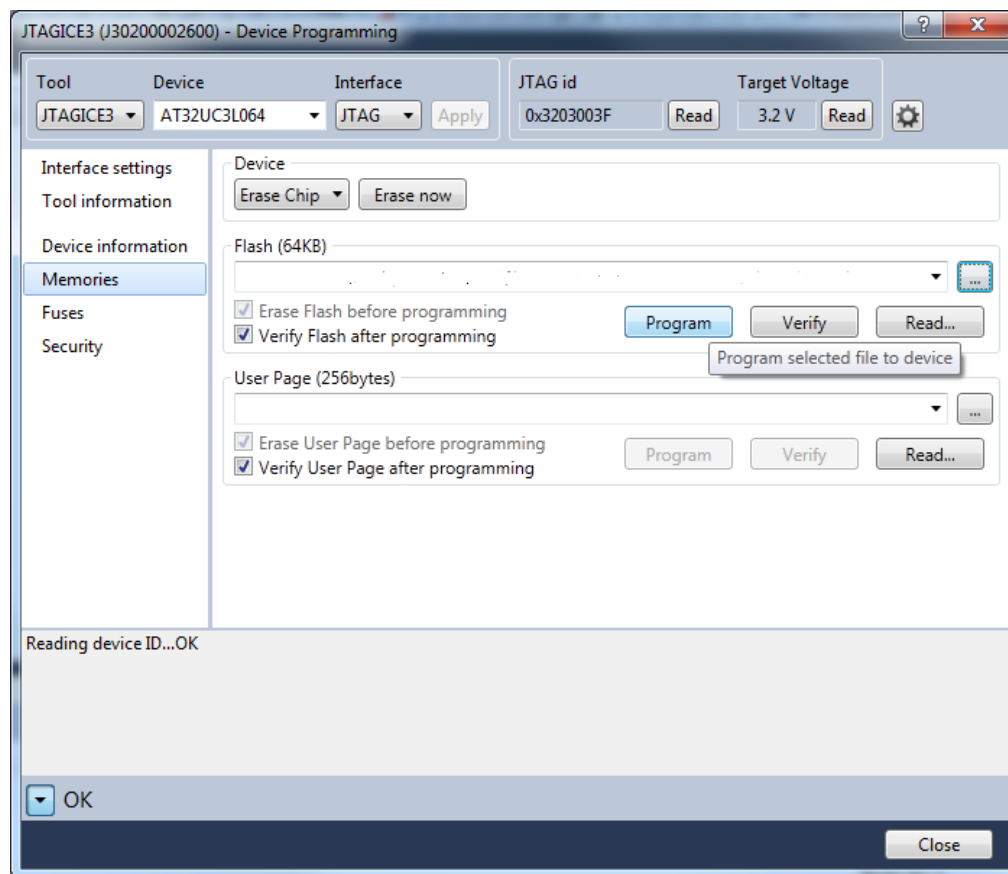
## 4.3 Programming via Atmel Studio

The device [AT32UC3L064](#) can also be programmed via [Atmel Studio](#) by connecting the [UC3-L0 Xplained](#) kit via connecting the programmers / debuggers as mentioned in [Chapter 3](#).

Open the Device Programming option by clicking the icon as shown below.



From the Device Programming window, select the Tool, Device and the programmable .hex file.



Note: The [IAR Embedded Workbench](#) can also be used for programming / Debugging as similar to Atmel Studio.

## 5. Example Applications

Several examples are available in ASF that is integrated with Atmel Studio IDE. Each example has full source code with documentation, and also readymade binaries to get started quickly. The examples can be compiled by using either GCC or IAR compilers.

### 5.1 Preprogrammed Firmware

The [Atmel AT32UC3L064](#) on the [Atmel UC3-L0 Xplained](#) kit is preprogrammed with a UART bootloader and a default example firmware.

#### 5.1.1 Functionality of the Preprogrammed Firmware

The preprogrammed firmware in the [AT32UC3L064](#) is a demonstration of the Atmel QTouch<sup>®</sup> button and Atmel QMatrix slider. By pushing the QTouch button (QTB0), LED1 will light up. The QTouch button will cycle through the selection of the colors red, green, and blue for the RGB LED. When moving a finger on the QMatrix slider (QMSlider), the intensity of the color selected by the QTouch button will change. The RGB colors are controlled with 8-bit resolution PWM, which theoretically gives the user the ability to mix up to 16.7 million colors.

##### 5.1.1.1 Capacitive Touch Module

The preprogrammed firmware was made to demonstrate the built-in hardware functionality of the [AT32UC3L064](#) device. The device has a capacitive touch (CAT) module, which can be used for both QTouch and QMatrix sensors. To demonstrate this, the [UC3-L0 Xplained](#) board has one of each sensor; one QTouch button and one QMatrix slider.

The design of the sensor defines if it is a QTouch or a QMatrix sensor. One of each is included for demonstration. The QMatrix slider can also be used as four separate QMatrix buttons.

The preprogrammed firmware uses the QTouch button to select among the three colors of the RGB LED to control with the QMatrix slider. When touching the QTouch button, LED1 will also light up while a touch is registered.

The QMatrix slider has an 8-bit resolution, which gives 256 different values according to where on the slider the touch is held. This value is passed on to the PWM module, which will set the brightness of the selected color on the RGB LED.

Note: Register and accept the license to get access to the QTouch libraries included in the source zip file provided with this application note.

##### 5.1.1.2 Pulse Width Modulation Controller

The [AT32UC3L064](#) device has a pulse width modulation controller (PWMA), which can output an 8-bit PWM signal on any of the I/O pins of the device. This module is used to control the brightness of each of the colors on the kit's RGB LED with the value received from the QMatrix slider and the color selected by the QTouch button.

##### 5.1.1.3 UART

The preprogrammed firmware has a UART module set up for transferring serial data through the board controller to a PC. This can be used for sending data to the PC to implement any communication in a custom design. This gateway is currently used to send data from the QTouch driver to the PC, making it possible to reprogram the board controller with QTouch HID firmware and then use this to connect the board in QTouch Studio for debugging and logging of QTouch data. The QTouch HID firmware can be downloaded from ASF 2.2.x.

### 5.2 More example projects

For further exploration of the [AT32UC3L064](#) device, see the application notes that are applicable to the device. There are also several example projects and drivers available in the ASF modules for the target device, [AT32UC3L064](#) integrated with Atmel Studio. Readymade projects targeted just for the [Atmel UC3-L0 Xplained](#) kit are also available.

## 6. How to Compile Example Projects

The compilers generate two files when compiling a project. The “hex” file contains the actual data that will be programmed into the device. This hex file can be programmed into the device using one of the programmers listed in Section 1.6, or by using the preprogrammed bootloader.

The other file generated is a debug file, which contains both the data to be programmed into the device and also debug information to make the code readable during debugging. This file can be opened in [Atmel Studio](#) for single stepping and debugging. The file extension of the debug file depends on the compiler, and will be .elf for the Atmel GNU Toolchain and .dbg or .d82 for the IAR toolchain.

### 6.1 AVR GNU Toolchain

To make this project using [Atmel AVR32 GNU Toolchain](#), open the shell/dos prompt and navigate to the gcc subfolder. Type “make,” and press enter. The binary and debug files are created as:

```
avr32_applications_uc3-10_xplained-demo_at32uc31064_uc3_10_xplained.hex  
avr32_applications_uc3-10_xplained-demo_at32uc31064_uc3_10_xplained.elf
```

### 6.2 IAR Embedded Workbench

To open the project in IAR, navigate to the iar subfolder and open the project:

```
avr32_applications_uc3-10_xplained-demo_at32uc31064_uc3_10_xplained.eww
```

The project can be compiled by pressing the F7 key on the keyboard, or select “make” from the project menu. The binary and debug files are created as:

```
avr32_applications_uc3-10_xplained-demo_at32uc31064_uc3_10_xplained.a82  
avr32_applications_uc3-10_xplained-demo_at32uc31064_uc3_10_xplained.d82
```



## 7. References

- [www.atmel.com/Xplained](http://www.atmel.com/Xplained)
- [www.atmel.com/tools/ATMELSTUDIO.aspx](http://www.atmel.com/tools/ATMELSTUDIO.aspx)

### 7.1 IAR Compiler

The IAR Compiler is a commercial C/C++ compiler that is available for Atmel AVR 32-bit devices. A 30-day evaluation version as well as a 32k (code size limited) kick-start version is available from the [IAR website](http://www.iar.com).

#### **IAR Embedded Workbench for Atmel AVR32**

<http://www.iar.com/en/Products/IAR-Embedded-Workbench/AVR32/>

### 7.2 Atmel Software Framework

This is a large library of drivers and code that is available for download and free use from the [Atmel website](http://www.atmel.com). It can be found here:

#### **Atmel Software Framework (ASF)**

<http://www.atmel.com/tools/AVRSOFTWAREFRAMEWORK.aspx>

## 8. EVALUATION BOARD/KIT IMPORTANT NOTICE

This evaluation board/kit is intended for use for **FURTHER ENGINEERING, DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY**. It is not a finished product, and may not (yet) comply with some or any technical or legal requirements that are applicable to finished products, including, without limitation, directives regarding electromagnetic compatibility, recycling (WEEE), FCC, CE, or UL (except as may be otherwise noted on the board/kit). Atmel supplied this board/kit “AS IS,” without any warranties, with all faults, at the buyer’s and further users’ sole risk. The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies Atmel from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user’s responsibility to take any and all appropriate precautions with regard to electrostatic discharge and any other technical or legal concerns.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER USER NOR ATMEL SHALL BE LIABLE TO EACH OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

No license is granted under any patent right or other intellectual property right of Atmel covering or relating to any machine, process, or combination in which such Atmel products or services might be or are used.

Mailing Address: Atmel Corporation, 1600 Technology Drive, San Jose, CA 95110

Copyright © 2013, Atmel Corporation

## 9. Revision History

Doc. Rev.	Date	Comments
32162B	10/2013	AVR Studio and AVR JTAGICE3 is added
32162A	03/2011	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](http://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.: 32162B-AVR-10/2013

Atmel®, Atmel logo and combinations thereof, AVR®, Enabling Unlimited Possibilities®, QTouch®, 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. 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.