
**AT07926: Connecting Wireless Networks to the Internet
using Xively Technology**

Atmel ATmega256RFR2**Description**

This application note exhibits techniques to connect IEEE® 802.15.4 wireless networks to the Internet. This example uses ATmega256RFR2 Xplained Pro demo boards as edge devices on a wireless sub-network. The Atmel® Low-Cost Ethernet to Wireless Gateway [1] serves as a bridge between the wireless sub-network and conventional Ethernet LAN infrastructure. Xively's cloud services tie the system together with remote control applications on the World Wide Web. This document describes how to start quickly with the provided sample applications and how to modify them for a specific need.

Features

- Remote control of XPRO demos boards using the internet
- Sample firmware for ATmega256RFR2 XPRO Remote Nodes
- Sample firmware for Atmel Low-Cost Gateway Reference Board
- Sample Xively web application for control
- ATmega256RFR2-XPRO Edge Devices

Table of Contents

1. Introduction	3
2. Xively Setup	4
2.1 Xively Introduction	4
2.2 New Xively Account	4
2.3 Xively Device Creation	4
2.4 Gateway Firmware Setup	5
2.5 Downloading Atmel Studio and Xively LCGW code	6
2.6 Xively Parameters Configuration	7
2.7 Building and Programming the Gateway Application	7
3. Device Firmware Setup	7
3.1 Overview	7
3.2 Programming Device Application	7
4. Physical Connection	7
5. Web Application	8
5.1 Web Application Setup	8
5.2 Making Modifications	10
6. References	11
7. Revision History	12

1. Introduction

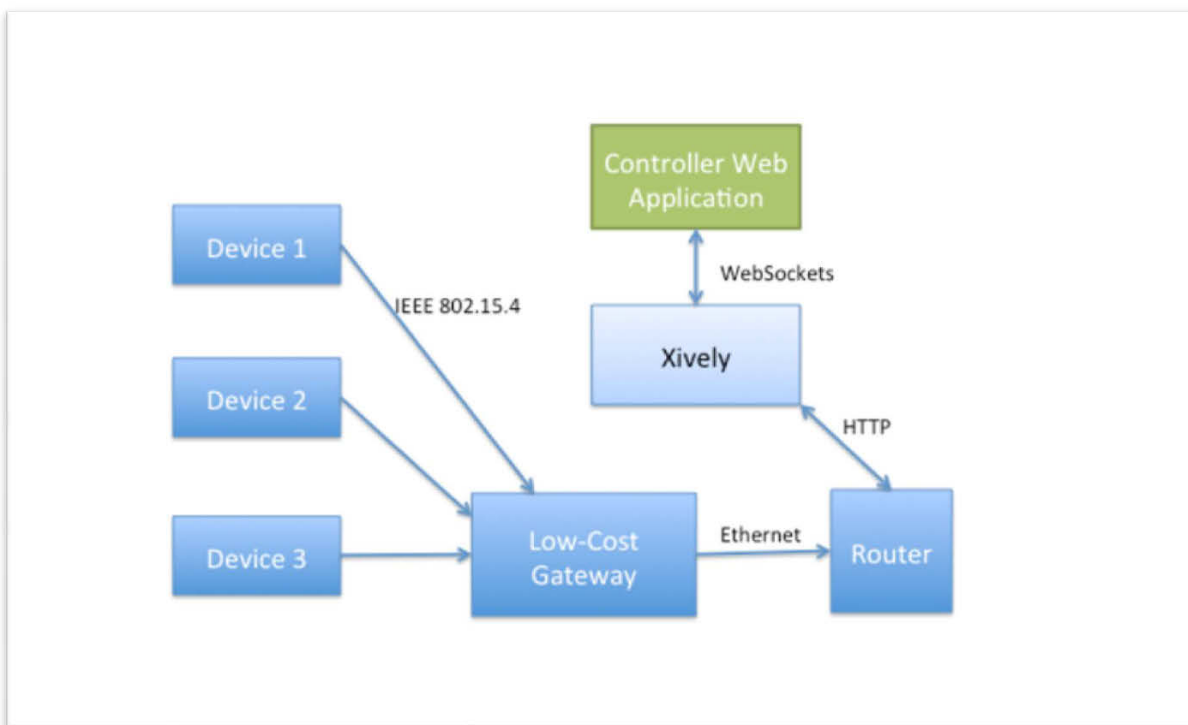
The provided example shows Xively's embedded C library running on the ATmega256RFR2, passing messages from the edge devices to Xively's cloud service. The Low-Cost gateway reference design [1] is the bridge between the wireless sub-network and Xively.

This particular implementation uses the Atmel Light Weight Mesh as the wireless sub-network. The ATmega256RFR2 supports several IEEE 802.15.4 network stacks including ZigBee® Pro, ZigBee Light Link, ZigBee HA, IEEE 802.15.4 MAC, and 6LoWPAN. This example exists in the user application layer and is not dependent on the lower layers of the wireless sub-network. In other words this solution can be ported to other IEEE 802.15.4 networks as needed.

The sub-network and basic application are derived from Atmel application note AT03755 Android demo application to control IEEE 802.15.4 devices [2]. For demonstration purposes, AT03755 implements simple indicator LEDs and GPIO button detection. Wireless Edge Devices can take various forms such as battery powered sensors, actuators, indicators, lights, thermostats, control panels and so forth. The firmware used on the ATmega256RFR2-XPRO Edge Device was not modified and is documented in AT03755 [2]. Details on modification to the Gateway Application firmware are detailed in this document.

Data (LED ON / LED OFF states), flows from the Xively service through to the edge devices via the lightweight mesh network. The LED states are sent to Xively by a front-facing web application. The basic architecture is explained in Figure 1-1.

Figure 1-1. Block Diagram



The sample web application will allow you to control the LED on the Device. Other commands can easily be setup to allow added functionality, but this is the default capability of the sample application. While both the application and Xively support multiple devices, for simplicity only one edge device will be used in this demo.

2. Xively Setup

2.1 Xively Introduction

This example shows how Xively technology can be used to control edge devices on wireless sub-networks and how to connect the wireless sub-network to conventional LAN infrastructure. Xively is a platform as a service for building connected products. Through Xively’s messaging platform, you can control local lightweight mesh networks with messages from anywhere in the world. Xively makes it easy to get started building connected experiences and you can use it for this demo for free.

2.2 New Xively Account

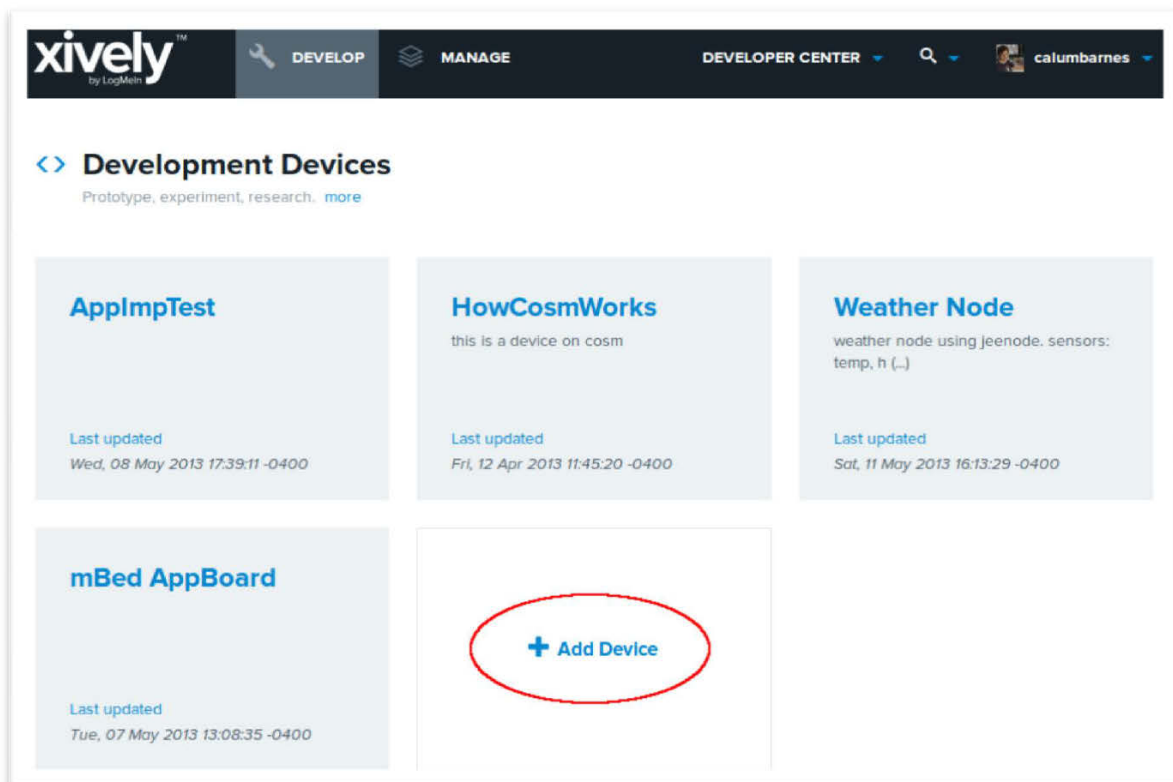
The demo requires a Xively account. If you haven’t made a Xively account already, go to (<https://xively.com/signup/>) to get one.

Now we will go through the process of adding a new device to the Xively Workbench, configuring it for our needs, and finding the necessary information to connect Xively to our Low-Cost gateway.

2.3 Xively Device Creation

- Start off by going to the Develop tab
- Once in the Develop tab, press the ‘Add Device’ button

Figure 2-1. Adding Device



- This will bring you to the “add new device” page; fill in a descriptive name for your device. “Atmel Gateway” is a good one.
- Select whether you want your feed to be public or private

- You can also add an optional description at this stage
- Finally click the “Add Device” button at the bottom to create your device
- You will be taken to the newly created Xively Workbench for your device
- The Xively Workbench is the center for all device development on Xively. It shows all your devices information (such as the feed ID which we will need later), recent requests to and from that device, current channel values, API keys, triggers, and more.
- Now that your device is setup, we are ready to move on. You will need two very important pieces of data from this page, the Feed ID and the API key (see [Figure 2-2](#)), so make sure you keep the page open or copy these down somewhere.

Figure 2-2. Feed ID and API Key

The screenshot shows the Xively 'Atmel Gateway' configuration page. The 'Private Device' section lists details like Product ID, Secret, Serial Number, and Activation Code. The 'Request Log' section shows a 'Waiting for requests' status. The 'API Keys' section displays an auto-generated device key for the feed ID 106526775. Red circles highlight the Feed ID and the device key.

2.4 Gateway Firmware Setup

The hardware gateway creates a bridge between the lightweight mesh network and the Internet, and from there to Xively. The gateway application is designed to be run on the ATmega256RFR2 Low-Cost Gateway reference design [1]. We will pass data from the lightweight mesh devices directly to Xively by using the translation layer provided in this gateway app. The gateway firmware queries data from Xively and passes relevant binary information to the devices in the mesh.

The gateway must be configured to talk to a specific Xively feed. By default, it is designed to be setup with only a single Xively feed. However, this feed can pass data to multiple lightweight mesh devices.

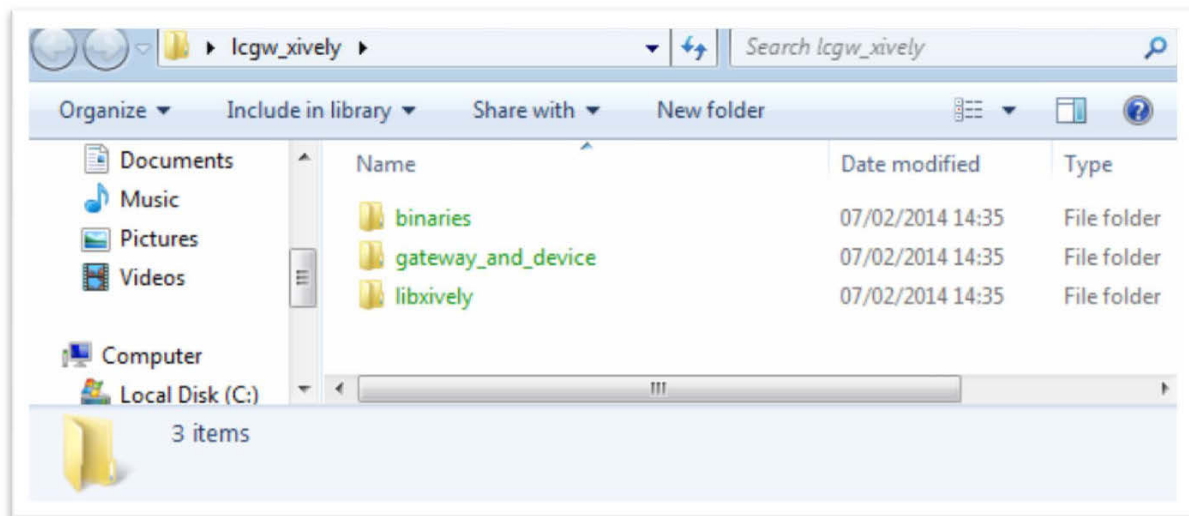
For the purposes of this tutorial all the building and configuration of the code will use Atmel Studio 6.1.

2.5 Downloading Atmel Studio and Xively LCGW code

In addition to the core gateway functionality, we must also configure the Xively parameters before we can start building the application and running it on the Low-Cost Gateway.

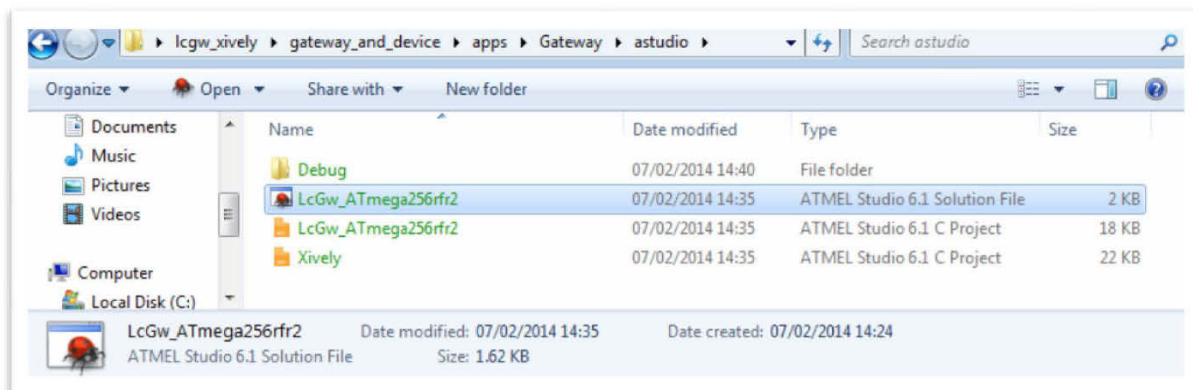
If you don't have it already, download and install Atmel Studio 6.1 from Atmel.com/Studio. Once installed, ensure that you do the "kits update" in order to ensure Xplained Pro support is up to date. In addition to the software, you will need a JTAGICE dongle to program the gateway using the J1 5x2 100 mil header.

Figure 2-3. Unzipped Folder View



The file lcgw_xively_v0.1.zip contains all the code needed to get started building a Low-Cost Gateway solution. Browse to where the project was unzipped as shown in the figure, and open the 'LcGw_ATmega256rfr2' solution file.

Figure 2-4. Opening Solution File



2.6 Xively Parameters Configuration

Now the project is ready to be configured with a specific Xively Feed and API Key. Open the 'appXivelyClient.h' file, and find the lines for 'XI_API_KEY' and 'XI_FEED_ID'. Replace these with the API key and Feed ID from the Xively setup step. This will connect the gateway with your Xively account.

2.7 Building and Programming the Gateway Application

Now the gateway application is ready to be built. To build it go to 'Build > Build Solution'. The output screen will show the build progress. Some Warnings may be thrown, but as long as there are no errors, your build has completed successfully and the code is ready for flashing onto the gateway chip.

3. Device Firmware Setup

3.1 Overview

The device application is designed to run on the ATmega256RFR2 Xplained Pro board. The application allows the LED on the board to be controlled with the web application built on Xively. This functionality can easily be expanded but is the sole focus of this demo.

3.2 Programming Device Application

The Xplained Pro boards have embedded USB debugger and can be programmed with the Device Programming Tool in Atmel Studio 6 and a USB Micro-B cable. The firmware running on the device has been reused from AT03755 [2]. You simply need to program the device with pre-compiled binary, which can be found in 'lcgw_xively\binaries\Device_1.hex'. The recommended fuse settings are 0xC2, 0x9D, 0xFE.

4. Physical Connection

At this point make sure the programmers are disconnected. The gateway is powered and connected to your LAN with an Ethernet cable. You can reset the gateway with SW2 and look for LED status lights on the RJ48 Jack to verify Ethernet activity. Next; power up the remote XPRO board using a battery pack, or USB cable. Reset the XPRO by pressing the RESET button.

5. Web Application

5.1 Web Application Setup

In order to use the dashboard app, you should already have a development device set-up on your Xively account. You will need the API key as well as the feed ID (the same you have used in the gateway firmware configuration header earlier). To access the demo version of the app, go to:

http://xively.github.io/lcgw_ctrl/#dashboard

You will be prompted for the feed ID and the API key, make sure you have entered both correctly. You can now bookmark the URL, so you will not have to enter these credentials again.

Figure 5-1. Xively Workbench with Feed ID to Copy

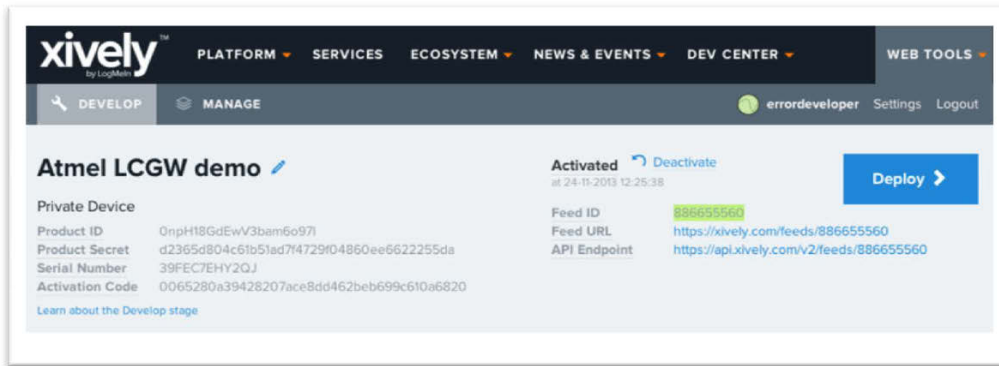
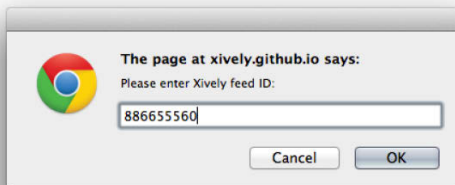
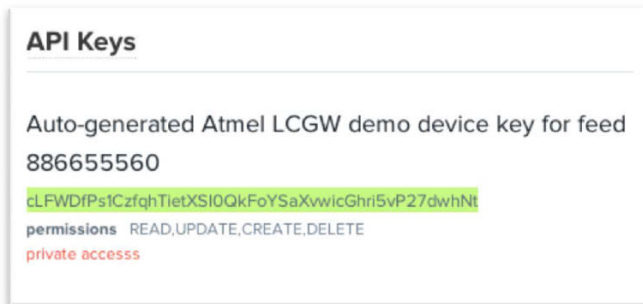
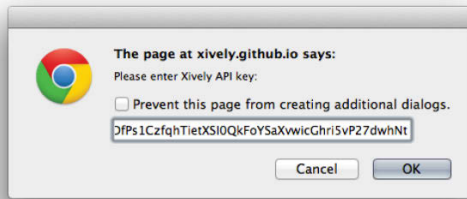


Figure 5-2. Xively Workbench (right) with API to Copy





Current version of the app consists of two views, named Dashboard and Commander. The Dashboard view is primary user interface to the network, while the Commander is more advanced and allows a developer to send arbitrary commands to the devices on the network; it is useful for testing newly added commands.

Figure 5-3. Dashboard View

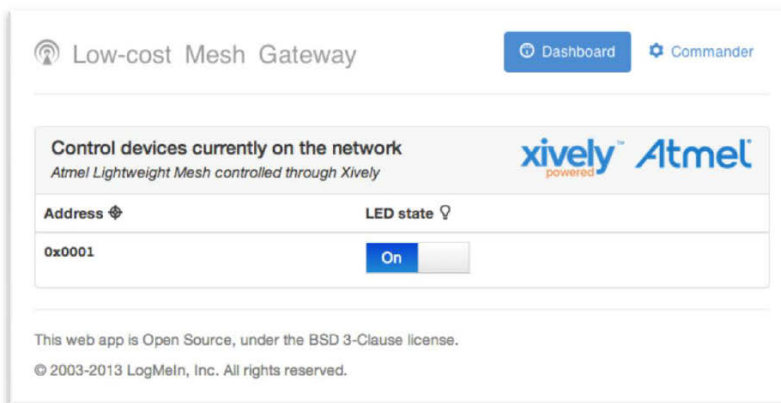
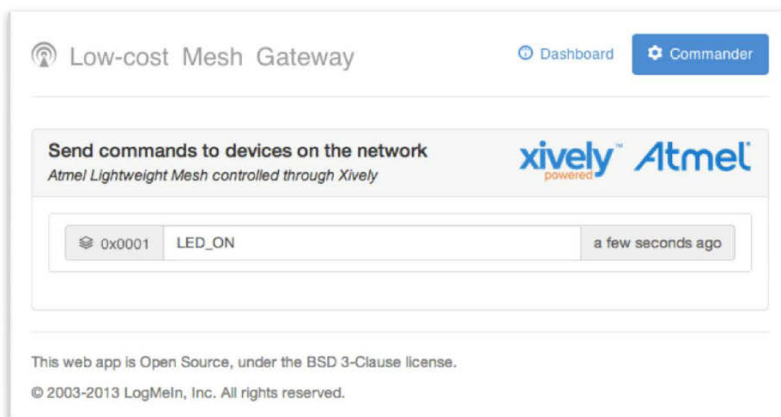


Figure 5-4. Commander View



When you have loaded the Dashboard view in your browser, you should see a toggle switch for each of the devices on the network. If there are no buttons, it means that the gateway is not connected to the Internet. You may have to RESET the gateway to initialize the application.

5.2 Making Modifications

The app has been developed using state of the art web frameworks, the key part being AngularJS and Bootstrap. For additional productivity, our developer chose to use Yeoman, Bower and Grunt, which are based on Node.js and Ruby, so you will need to have Ruby and Node.js installed. To get started, refer to the latest version of README.md document.

If you prefer to start from scratch, you can always use either Xively JavaScript library (XivelyJS) or any of the native libraries.

6. References

- [1] [AT01030: Low-cost Ethernet to Wireless Gateway with ATmega256RFR2.](#)
- [2] [AT03755: Android demo application to control IEEE 802.15.4 devices.](#)

7. Revision History

Doc. Rev.	Date	Comments
42275A	04/2014	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
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 Bldg.
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032

JAPAN

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

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

© 2014 Atmel Corporation. All rights reserved. / Rev.: 42275A-AVR-04/2014

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, AVR®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. 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.