SMART ARM-Based Microcontroller

AT13519: ATSAMR21 LED Driver with ZigBee Light Link – Firmware User Guide

USER GUIDE

Introduction

The Atmel® SAM R21 LED Driver is a reference platform for a Dimmable Light with a LED Driver implemented in firmware, which can be controlled wirelessly over ZigBee® using a ZigBee-based remote control device.

This user guide explains the firmware components of the system.

Features

- LED Driver module implemented in firmware (no external driver IC needed)
- Dimmable Light application as per ZigBee Light Link Specification – implementation uses the Atmel BitCloud® SDK
- 0.5% to 100% dimming using combination of analog and PWM dimming
- Up to 95% efficiency
- Light control from the AVR477 Touch Remote Control

Table of Contents

1. Overview

This document is a firmware user guide for the ATSAMR21-LED-DRIVER reference platform.

The ATSAMR21-LED-DRIVER is a reference design for a wireless dimmable light that uses the SAM R21 MCU to implement a firmware LED driver controlled from a remote control using ZigBee.

1.1. System Overview

The ATSAMR21-LED-Driver is built on a SAM R21 Xplained Pro (ATSAMR21-XPRO) board with modified connectors, an add-on power component board, and an LED load board with diffuser.

The Atmel AVR477 remote control is used to control the ATSAMR21-LED-Driver board (Dimmable Light) over ZigBee Light Link.

Figure 1-1 ATSAMR21-LED-Driver Board (Light) with AVR477 Remote Control

1.1.1. ATSAMR21-LED-Driver (Dimmable) Light

The ATSAMR21-LED-Driver board uses the SAM R21 MCU which is an IEEE® 802.15.4 compliant SiP incorporating an ARM® Cortex® M0+ based 32-bit microcontroller and a 2.4GHz RF transceiver.

The LED driver is implemented in firmware without using any external DC-DC LED driver IC.

This driver is integrated in the BitCloud SDK – The Atmel ZigBee PRO stack that supports the ZigBee Light Link (ZLL) and ZigBee Home Automation (ZHA) Profiles. This solution uses the ZigBee Light Link (ZLL) profile reference application from BitCloud and this can be easily updated to be part of ZigBee Home Automation (ZHA) networks.

The LED driver firmware implements a DC-DC buck converter, operating in boundary conduction mode to provide constant current to the connected LEDs.

The wireless functionality is as defined in the ZLL specification for a dimmable light.

1.1.2. Remote Control

The AVR477 Remote control board uses the ATmega256RFR2, which is an IEEE 802.15.4 compliant SoC incorporating an AVR® 8-bit microcontroller and a 2.4GHz RF transceiver.

This board has nine capacitive touch keys and one touch wheel and is used as the ZigBee Remote control device for controlling the dimmable light. The wireless functionality is as defined in the ZLL specification for a remote controller.

2. Dimmable Light – The ATSAMR21-LED-Driver Board

The firmware for the ATSAMR21-LED-Driver Board is an integration of two major components:

1. The LED driver for driving the LED load.
2. The ZigBee BitCloud SDK for wireless control of the light.

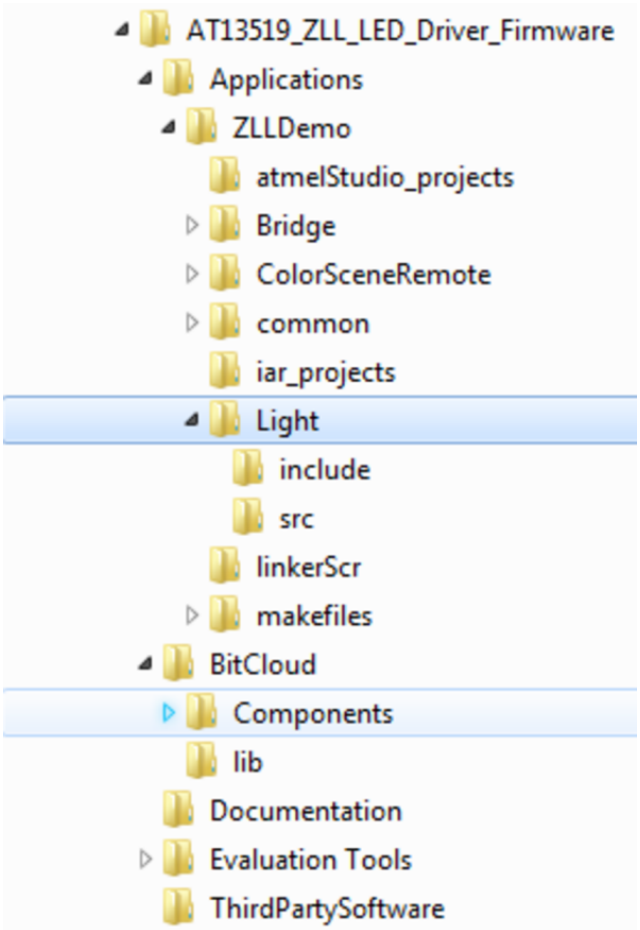
2.1. Source Code Organization

The Dimmable Light application is part of the ZigBee Light Link (ZLL) reference application in the BitCloud SDK.

The Figure 2-1 Folder Structure on page 5 shows the BitCloud SDK package. The Dimmable Light application files are present in the `\Light` inside the `\Applications\ZLLDemo` folder.

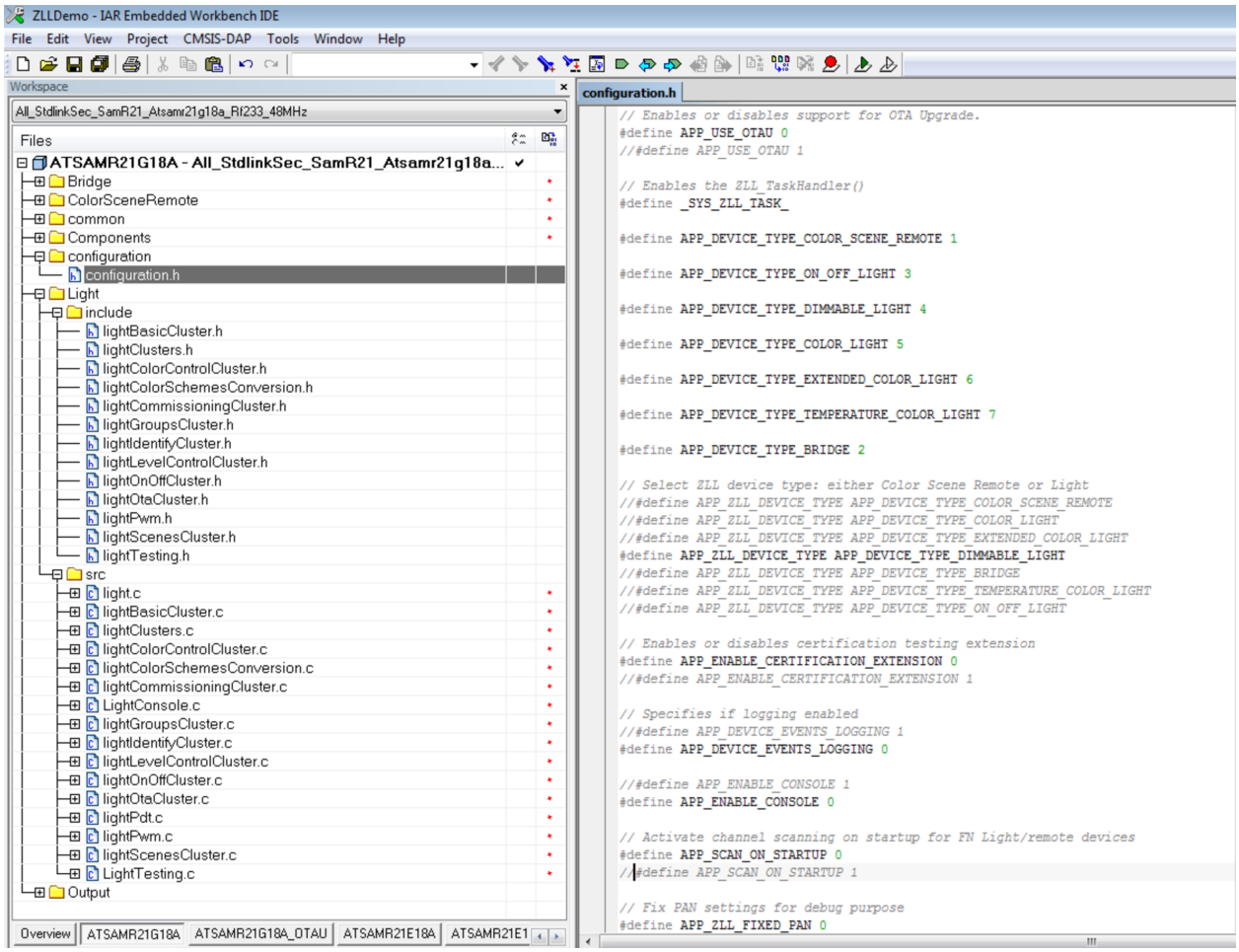
The folder `\BitCloud\lib` contains the BitCloud core stack library and hardware abstraction layer(HAL) libraries to be built along with the application.

Figure 2-1 Folder Structure



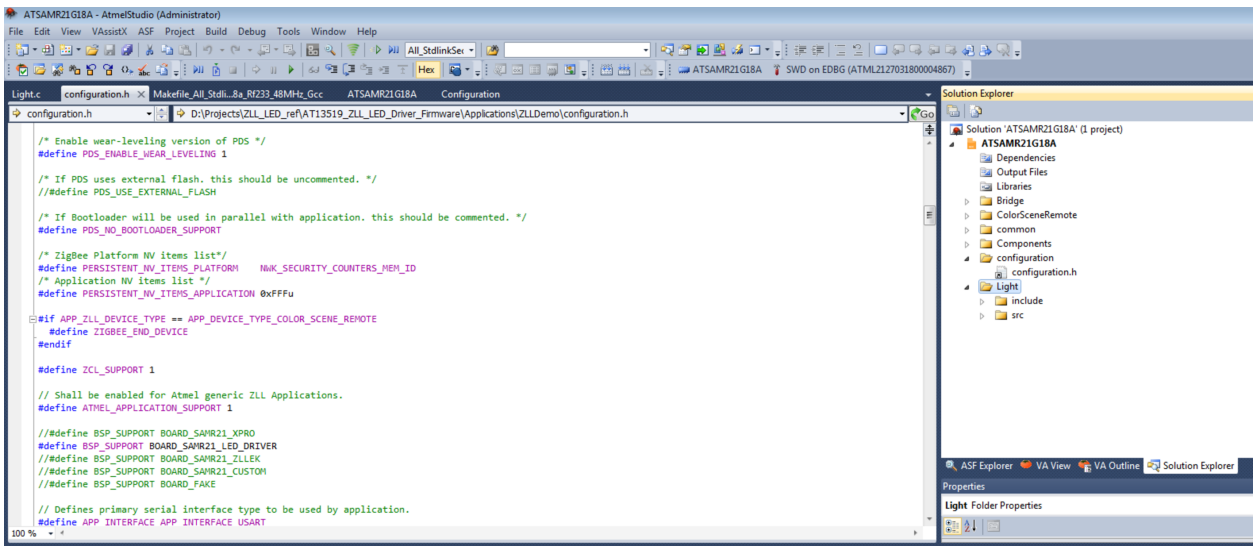
The ATSAMR21-LED-Driver project uses the 48MHz configuration which is based on the 48MHz CPU clock configuration in the SAMR21G18A. Figure 2-2 IAR Embedded Workspace for the ATSAMR21-LED-Driver Board on page 6 captures the project workspace of the light in IAR Embedded Workbench®.

Figure 2-2 IAR Embedded Workspace for the ATSAMR21-LED-Driver Board



The Figure 2-3 Atmel Studio Project for the ATSAMR21-LED-Driver Board on page 6 is also provided in the firmware package available with the application note.

Figure 2-3 Atmel Studio Project for the ATSAMR21-LED-Driver Board



Note: The firmware is based on BitCloud SDK version 3.2.0 and the IAR™ and Atmel Studio versions are as per the BitCloud Quick Start Guide.

2.2. Project Configuration

This firmware is based on the following project configuration.

IAR Embedded Workbench:

1. Open ZLLDemo.eww (`\Applications\iar_projects`)-> ATSAMR21G18A.ewd
2. Select project configuration: `All_StdlinkSec_SamR21_Atsamr21g18a_Rf233_48MHz_Iar`

Atmel Studio:

1. Open ATSAMR21G18A.atsln (`\Applications\atmelStudio_projects`)
2. Select project configuration: `All_StdlinkSec_SamR21_Atsamr21g18a_Rf233_48MHz_Gcc`

The BitCloud core libraries from the standard BitCloud SDK release are used.

The HAL libraries are generated with specific changes for ATSAMR21-LED-Driver board (described in Configuration of Parameters on page 7) are:

- **IAR Embedded Workbench:** `libHAL_Samr21_LedDrv_Atsamr21g18a_48Mhz_Iar.a`
- **GCC/Atmel Studio:** `libHAL_Samr21_LedDrv_Atsamr21g18a_48Mhz_Gcc.a`

2.3. Configuration of Parameters

The ZLLDemo reference application has been modified to suit the requirements of the ATSAMR21-LED-Driver board.

Changes with respect to MCU operating frequency, peripheral configuration, and usage are performed in the hardware abstraction layer(HAL).

Optionally, to further modify HAL, the required code can be edited. The HAL can be re-built from command-line based on the instructions available in BitCloud Quick Start Guide.

The HAL Configuration (`\BitCloud\Components\HAL\Configuration`) parameters relevant to the ATSAMR21-LED-Driver board are covered under build switch `PLATFORM_SAMR21_LED_DRIVER`.

`SAMR21_LED_DRIVER_PWM_CONFIG` defines the clock and timer configuration specific to the ATSAMR21-LED-Driver board and the value shall be set to `True`.

`PWM` shall be set to `False` as this is the default BitCloud timer configuration and is not used for this board.

The unique 64-bit MAC address for the ATSAMR21-LED-Driver board is copied from EDBG chip and stored in the auxillary page of the internal flash in SAMR21G18A. `FLASH_AUX_PAGE_ACCESS` shall be set to `True` to retain this implementation.

The Dimmable Light has several configuration parameters relevant to ZigBee functionalities such as joining a network, addressing, networking tables as well parameters relevant to board support. Such configurable parameters are available in the `\Application\configuration.h` file, as shown in Figure 2-2 IAR Embedded Workspace for the ATSAMR21-LED-Driver Board on page 6.

Parameters relevant to the ATSAMR21-LED-Driver board support are:

`#define APP_ZLL_DEVICE_TYPE APP_DEVICE_TYPE_DIMMABLE_LIGHT` - sets the ZLL Device type as Dimmable Light.

`#define APP_SCAN_ON_STARTUP 0` - disables active scanning for ZigBee networks when the light is reset.

`#define BSP_SUPPORT BOARD_SAMR21_LED_DRIVER` - includes all board related implementation for the ATSAMR21-LED-Driver board.

`#define APP_USE_PWM` - shall not be defined as this is the default PWM implementation in BitCloud and shall not be used for this board.

Default values can be retained for the other parameters in this file.

2.4. Memory Footprint

The memory footprint of the firmware used on the light and the remote are as follows:

- Light (ATSAMR21) :
  - 150.742KBytes of read-only code memory
  - 3.626KBytes of read-only data memory
  - 43.657KBytes of read-write data memory
- Remote (ATmega256RFR2) :
  - 187.733KBytes of CODE memory
  - 22.477KBytes of DATA memory

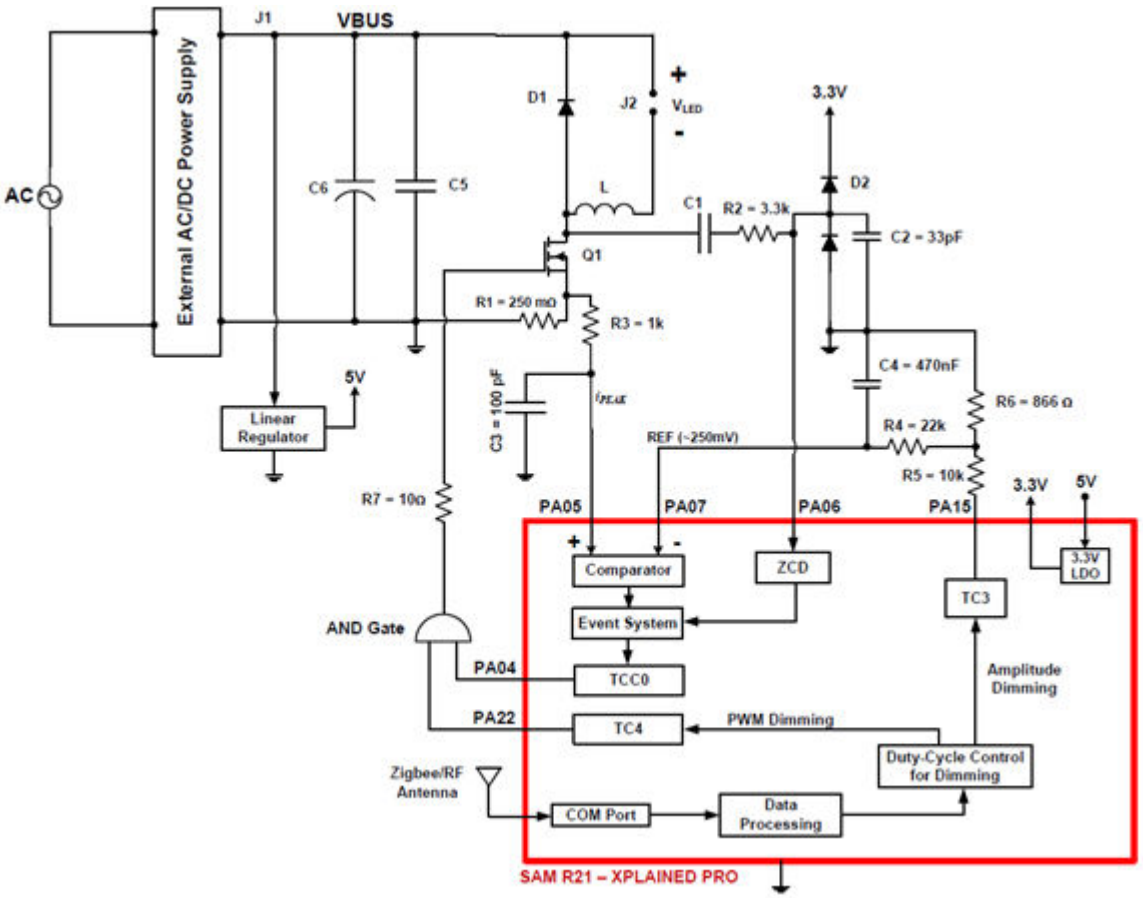
This memory footprint is obtained when compiling the projects with IAR Embedded workbench for AVR(Remote Control) and ARM (light).

2.5. Operating Principles of the LED Driver

The LED Driver implementation is fully firmware-based and uses various peripherals of the SAM R21 MCU to drive the LEDs.

Figure 2-4 Circuit Block Diagram on page 9 shows the block diagram of the scheme used in the implementation of buck LED driver.

Figure 2-4 Circuit Block Diagram



The SAMR21 controls the LED peak current using boundary-conduction-mode. The peripherals used for the implementation are:

1. One Analog Comparator to monitor inductor or MOSFET peak current
2. One Timer (TC3) for reference voltage generation for the inductor peak current for Amplitude Dimming
3. One Timer (TC4) for PWM Dimming
4. One PWM Timer (TCC0) for MOSFET control
5. One external interrupt input channel for inductor Zero-Current-Detection (ZCD)
6. Event system

The event system connects the output of the analog comparator and external interrupt (ZCD). It sends a signal to the timer TCC0 to start or stop the PWM signal for the buck MOSFET depending on the mode-of-transition (high-to-low or low-to-high) of analog comparator output and ZCD signals.

The reference for the peak current is generated using timer TC3. Timer TC3 generates a PWM signal at 10kHz with a desired duty-cycle and the PWM signal is passsed through a low-pass filter to give a DC reference for the analog comparator. The duty-cycle of TC3 is varied to adjust the reference voltage to achieve amplitude dimming.

The LED Driver implementation is available in `\Light\src\LightPwm.c` and corresponding header file.

2.6. ZigBee Functionalities of the Dimmable Light

The ATSAMR21-LED-Driver board implements a Dimmable Light as per ZLL specification. This section outlines the major wireless functionalities of the ATSAMR21-LED-Driver board.

2.6.1. Joining a ZigBee Light Link Network

As per the ZLL specification, the dimmable light is configured as a ZigBee Router. The Dimmable light is brought into a ZigBee network through a commissioning process known as Touchlinking, which is initiated from a remote control. Hence it is expected that, on startup, the light does not explicitly scan for ZigBee networks in the vicinity. This is an application controlled behavior and is disabled by setting the macro `APP_SCAN_ON_STARTUP` to zero as mentioned in section Configuration of parameters.

The AVR477 Remote Control is used to touchlink the dimmable light. Touchlinking is the process of sending network information to a light to make it join a ZigBee network. The functionality is described in section Remote Control – The AVR477 Touch Remote Control Board on page 13.

2.6.2. Unique MAC Address Storage

The SAMR21-XPRO has an EDBG chip with a unique 64-bit MAC Address stored on each light. The SAMR21G18A MCU reads the MAC address on start-up from the EDBG chip and stores it in the internal flash (auxiliary page) when the light application is initialized for the first time. Subsequently, in case the light is reset and initializes, the MAC address is read from the internal flash.

2.6.3. Information Sets/Clusters Supported by the Dimmable Light

The ZLL specification defines a set of functionalities to be supported in a dimmable light. This is organized as a set(cluster) of variables(attributes) and control commands. Typical attributes include on/off state of the light and brightness level. These attributes can be controlled from the remote using commands to adjust the dimming, on/off state of the light.

The table below lists the clusters supported on the light. Each cluster has a set of attributes and commands.

Example: Level Control cluster has an attribute named `CurrentLevel` which can be read from the remote control. The remote control can use commands such as `step` to change the brightness level of the light. The step command modifies the `CurrentLevel` value on the light.

Table 2-1 Dimmable Light - Supported Clusters

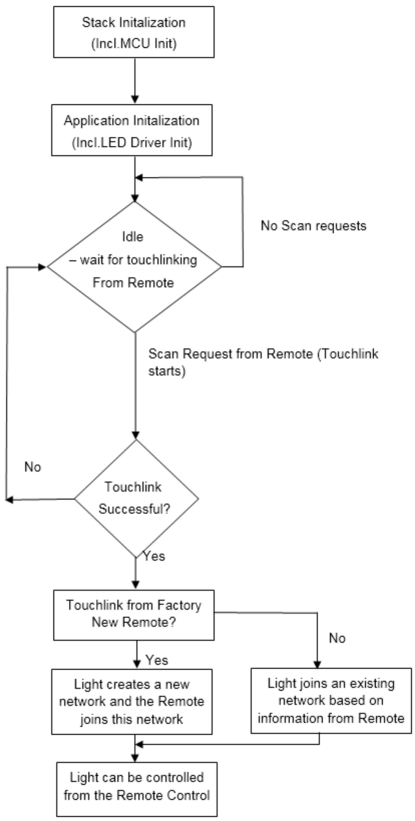
Dimmable Light Device (ID: 0x0100)	
Server Clusters	**Client Clusters**
Basic	Basic
Commissioning	NA
Identify	NA
Groups	NA
Scenes	NA
Level Control	NA
On/Off	NA

Note: Over-the-Air Upgrade (OTAU) Cluster is also supported in BitCloud, however the hardware, ATSAMR21-LED-Driver, does not have an external flash on board to save the updated firmware.

2.6.4. Application Code Flow

The following flow diagram shows the overall application code flow on the light.

Figure 2-5 Application Flow Diagram



2.6.5. Light Control from the Remote

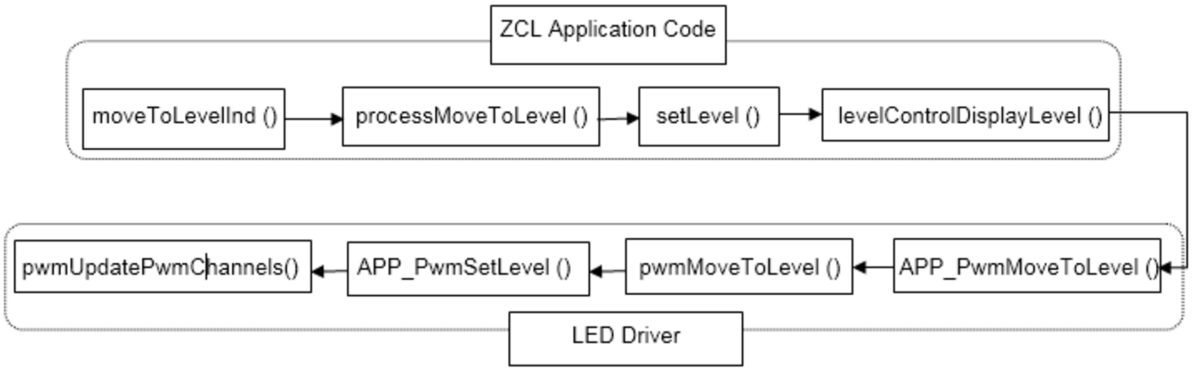
The control of the Light from the Remote happens in a client-server model. This is as per the Zigbee Cluster Library (ZCL) specification. The light holds the relevant data (attributes) and is the server. The Remote is the client and manipulates the attributes on the light through commands.

Example: Change in brightness level of the Light from the AVR477 Remote Control.

The `Move to Level` command is part of the Level Control Cluster. This command is sent from the remote to the light with the parameters: `level` and `transition time`. When the light receives this command, it should move to the new level over the specified transition time.

The above example is the over-the-air command exchange from the remote to the light. The light needs to convert the received level into a PWM value that will be provided to the LED driver to cause the LED to change brightness accordingly.The API flow to perform this is shown in Figure 2-6 API Usage Example on page 12 .

Figure 2-6 API Usage Example



When the wireless command to move to level reaches the ATSAMR21 MCU, it is propagated to the application from the lower layers in the stack to the application. The `moveToLevelInd()` function (in `lightLevelControlCluster.c`) is an indication function to inform the application that a new `moveToLevel` command has been received.

This command is processed by the application and provided to the LED Driver via function `APP_PWMMoveToLevel()` (in `lightPwm.c`). This function scales the level value to the PWM duty cycle value and starts a timer to periodically move level. This is done so that the new level is achieved at the end of the provided transition time.

3. Remote Control – The AVR477 Touch Remote Control Board

The firmware for the AVR477 Remote Control is an integration of two major components – the QTouch® library for capacitive touch sensing and the ZigBee BitCloud SDK for wireless control of the light. The ATmega256RFR2 MCU is used on the AVR477 to allow for the memory requirements by the BitCloud ZLL reference application.

The procedure in the application note - Integration of QTouch Library with BitCloud Zigbee Light Link, describes a method to integrate the QTouch Library into BitCloud SDK and this firmware is used to control the light. The application used is the reference ZLL application where the device type is selected as color scene remote controller.

As per ZLL specification, the remote acts as client device and controls the light through several commands and manipulates the light's attributes. However, since the AVR477 hardware has nine touch keys and one rotor, the control functionality has been chosen as per Table 3-1 Touch Keys for executing ZLL Remote Controller Commands on page 14 below. This is also provided in AT13520: ATSAMR21 LED driver with ZigBee Light Link – Quick Start Guide.

Figure 3-1 AVR477 QTouch Buttons and Rotor Functionality

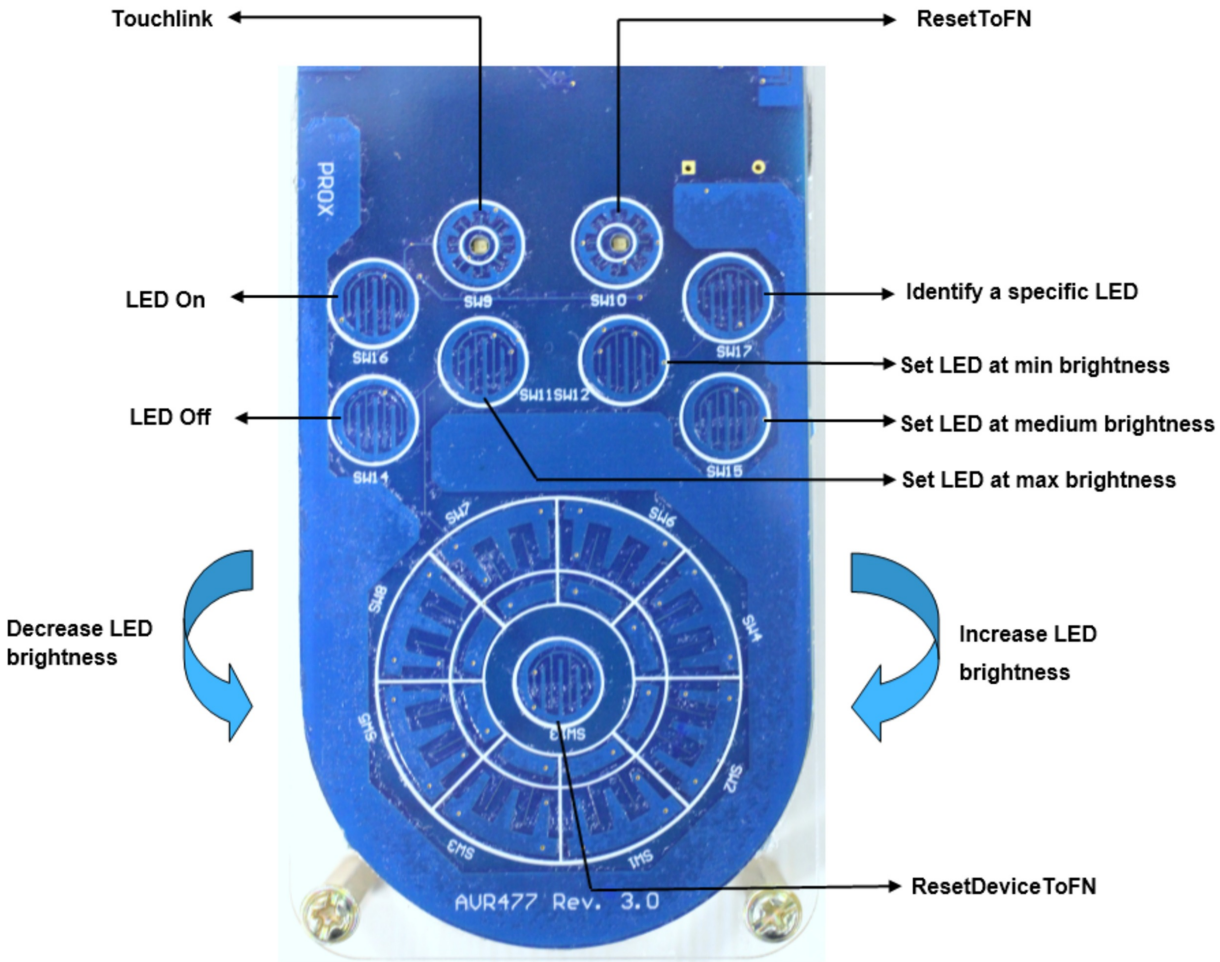


Table 3-1 Touch Keys for executing ZLL Remote Controller Commands

Key	Description
SW9	Initiate touchlinking procedure
SW10	Extended Press of this key for 3 seconds causes Remote to Reset to factory new state
SW16	On Command to light
SW14	Off Command to light
SW17	Select the next bound device and requests it to identify itself. This allows sending unicast commands to a single device. Groupcast mode will be entered automatically after twenty seconds of inactivity.
SW11	Set light level to minimum brightness
SW12	Set light level to medium brightness
SW15	Set light level to maximum brightness
Rotor (SW1 – SW8)	Increase/decrease light level by pre-defined steps. Clockwise rotation causes increase in light brightness and anti-clockwise rotation causes a decrease in light brightness level.
SW13	Reset device to factory new state (This will cause Light to be reset to factory new state)

3.1. Status LED Indications on the AVR477 Remote

There are two LEDs on-board the AVR477 Remote – one under SW9 and another under SW10. The LED notifcation behavior is explained in the following table.

Table 3-2 LED Notifications on the AVR477 Remote

Action	LED Behavior
MCU Active state	Both LEDs (under SW09 and SW10) remain turned ON
MCU Sleep state	Both LEDs are turned OFF
Any touch key press	LED under SW10 goes from ON to OFF state and remains OFF as long as the touch key is pressed. When the finger is removed, it turns ON again.
Wheel rotation	LED under SW10 toggles as long as the rotation is done. Once the finger is removed, the LED remains ON.
resetToFN	When SW10 is long-pressed, the LED under SW09 goes from ON to OFF state and remains OFF for about 2 seconds and then both LEDs blink once and turn OFF.
resetDeviceToFN	When SW13 is long-pressed, the LED under SW09 turns OFF for about 2 seconds and turns ON again. LED under SW10 remains ON.
Successful Touchlink	When SW09 is pressed, after approximately 5 seconds, both LEDs blink three times to indicate successful touchlink. After this, both LEDs are turned ON.

This is also provided in the Quick Start Guide of the ATSAMR21-LED-Driver board.

3.2. Resetting the Light to Factory New State from the AVR477 Remote

It is possible to bring the light to a “Factory New” state by sending a command from the AVR477 Remote after successful touchlinking. This erases all its networking information and returns the light to a state with factory default settings. The light will no longer be part of the network.

3.3. Controlling Multiple Lights from a Single AVR477 Remote Control

When the “Factory New” AVR477 Remote Control is touchlinked for the first time with a light (also factory new), the light will create the network and the remote control joins the network with the light as its parent (see Figure 2-5 Application Flow Diagram on page 11). Both devices are now non-factory new.

To add more number of lights to this ZLL network, the non-factory new remote can be touchlinked with other factory new lights. A successful touchlink with a factory new light brings this light into the network and the light becomes non-factory new.

When there are several lights in the network, the remote control can control the lights as a group (example: turn on/off several lights using a single command) or control lights individually. To control a light individually, the AVR477 Remote must select the light using the Identify touch button, SW17. When the light to be controlled identifies itself by blinking, the commands sent from the remote for the next 20 seconds are unicast to the selected light only.

4. Related Documents

- AT13520: ATSAMR21 LED driver with ZigBee Light Link – Quick Start Guide
- AT13518: ATSAMR21 LED driver with ZigBee Light Link – Hardware User Guide

5. Revision History

Doc Rev.	Date	Comments
42483A	08/2015	Initial document release

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**