Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

XMC1200

Microcontroller Series
for Industrial Applications

RGB LED Lighting Shield with XMC1202 for Arduino

- ✓ Introduction
- ✓ Board Description
- ✓ Getting Started
- ✓ I²C Master-Slave Communication
- ✓ Programming a master Arduino board to control the RGB LED Lighting Shield
- ✓ Setting the Parameters for YOUR LED Lamp

Board Manual

V1.0 2014-11

Microcontrollers

Edition 2014-11

Published by
Infineon Technologies AG
81726 Munich, Germany

© 2014 Infineon Technologies AG
All Rights Reserved.

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision History

Page or Item	Subjects (major changes since previous revision)
V1.0, 2014-11	

Trademarks of Infineon Technologies AG

AURIX™, C166™, CanPAK™, CIPOS™, CIPURSE™, CoolMOS™, CoolSET™, CORECONTROL™, CROSSAVE™, DAVE™, DI-POL™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPIM™, EconoPACK™, EiceDRIVER™, eupec™, FCOS™, HITFET™, HybridPACK™, I²RF™, ISOFACE™, IsoPACK™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OptiMOS™, ORIGA™, POWERCODE™; PRIMARION™, PrimePACK™, PrimeSTACK™, PRO-SIL™, PROFET™, RASIC™, ReverSave™, SatRIC™, SIEGET™, SINDRION™, SIPMOS™, SmartLEWIS™, SOLID FLASH™, TEMPFET™, thinQ!™, TRENCHSTOP™, TriCore™.

Other Trademarks

Advance Design System™ (ADS) of Agilent Technologies, AMBA™, ARM™, MULTI-ICE™, KEIL™, PRIMECELL™, REALVIEW™, THUMB™, µVision™ of ARM Limited, UK. AUTOSAR™ is licensed by AUTOSAR development partnership. Bluetooth™ of Bluetooth SIG Inc. CAT-iq™ of DECT Forum. COLOSSUS™, FirstGPS™ of Trimble Navigation Ltd. EMV™ of EMVCo, LLC (Visa Holdings Inc.). EPCOS™ of Epcos AG. FLEXGO™ of Microsoft Corporation. FlexRay™ is licensed by FlexRay Consortium. HYPERTERMINAL™ of Hilgraeve Incorporated. IEC™ of Commission Electrotechnique Internationale. IrDA™ of Infrared Data Association Corporation. ISO™ of INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. MATLAB™ of MathWorks, Inc. MAXIM™ of Maxim Integrated Products, Inc. MICROTEC™, NUCLEUS™ of Mentor Graphics Corporation. MIPI™ of MIPI Alliance, Inc. MIPS™ of MIPS Technologies, Inc., USA. muRata™ of MURATA MANUFACTURING CO., MICROWAVE OFFICE™ (MWO) of Applied Wave Research Inc., OmniVision™ of OmniVision Technologies, Inc. Openwave™ Openwave Systems Inc. RED HAT™ Red Hat, Inc. RFMD™ RF Micro Devices, Inc. SIRIUS™ of Sirius Satellite Radio Inc. SOLARIS™ of Sun Microsystems, Inc. SPANSION™ of Spansion LLC Ltd. Symbian™ of Symbian Software Limited. TAIYO YUDEN™ of Taiyo Yuden Co. TEAKLITE™ of CEVA, Inc. TEKTRONIX™ of Tektronix Inc. TOKO™ of TOKO KABUSHIKI KAISHA TA. UNIX™ of X/Open Company Limited. VERILOG™, PALLADIUM™ of Cadence Design Systems, Inc. VLYNQ™ of Texas Instruments Incorporated. VXWORKS™, WIND RIVER™ of WIND RIVER SYSTEMS, INC. ZETEX™ of Diodes Zetex Limited.

Last Trademarks Update 2011-11-11

Table of Contents

About this document

Scope and purpose

This document describes how to use the RGB LED Shield with XMC1202 for Arduino.

Intended audience

Engineers, hobbyists and students who want to add flicker-free LED control to Arduino projects.

Related information

Table 1 Supplementary links and document references

Reference	Description
XMC Microcontrollers	32-bit Industrial Microcontroller based on ARM® Cortex™-M from Infineon
XMC1000 Reference Manuals	Documents section contains reference information for XMC1000 microcontrollers
XMC Development Support	XMC Development Tools
Arduino Home Page	All information on Arduino
Arduino Uno Product Page	Arduino Uno R3 description
Infineon Arduino Page	Boards offered by Infineon for Arduino
DAVE™ Development Platform	All details on DAVE™ IDE
J-Link Debug Probes Product Page	Contains information on J-Link Debug Probes

RGB LED Lighting Shield Introduction

1 Introduction

The RGB LED Lighting Shield adds brilliant flicker-free light control to Arduino projects. The Shield communicates with a master board via the I^2C protocol as a slave. Either an Arduino Uno R3 or the XMC1100 Boot Kit from Infineon can be used as the master.

On board the RGB LED Shield is an XMC1202 microcontroller, featuring a dimming control peripheral for LED lighting applications, known as the Brightness and Colour Control Unit (BCCU). It contains 3 independent dimming engines and 9 independent Pulse Density Modulated (PDM) channels. 1 dimming engine and 6 channels are used in this shield.

There are 10 basic sets of I^2C commands to control the shield from the master board, and so control the connected LED Lamp with various lighting effects. There are 22 user configurable parameters and the freedom to connect different LED Lamps.

The RGB LED Lighting Shield can be easily connected to any Arduino board or the XMC1100 boot Kit via headers and DMX512 control is enabled as a mounting option using an interface chip.

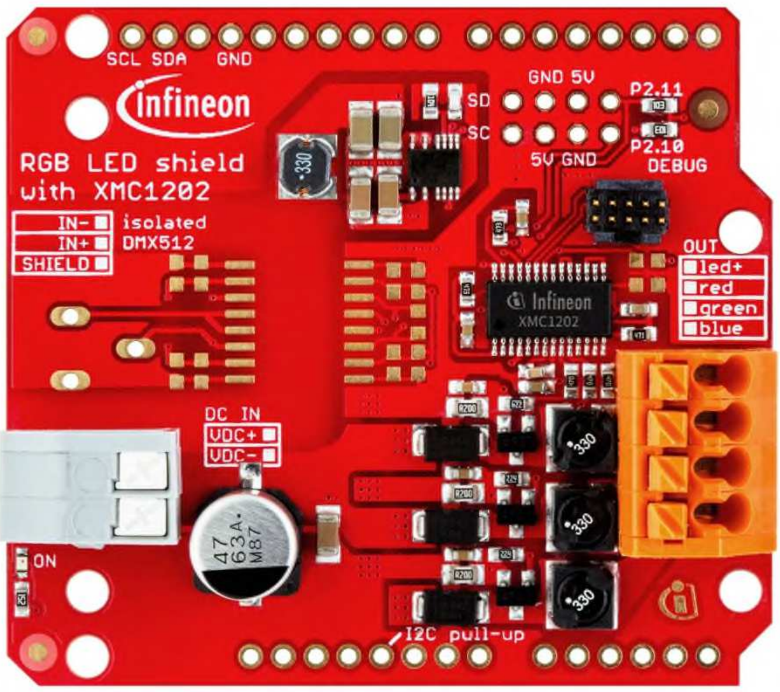

Figure 1 RGB LED Shield photo

1.1 Key Features

The RGB LED Shield has the following features:

- Behaves as an I^2C slave.
  - An Arduino Uno R3, XMC1100 Boot Kit, or similar board connected to the shield can communicate via the SDA and SCL pins as the master.
- Drives and dims up to 3 LED strings with constant current.
- Able to change the colour of a connected LED lamp(if the strings are of different colours; for example red, green, blue).
- High speed flicker-free modulation dimming on each string with Pulse-Density Modulation (PDM).
- Very high power density due to high switching frequency, leading to a small area.

- Up to 48V_{DC} input.
  - The RGB LED Shield is a DC-DC buck LED driver so the input voltage must be higher than the forward voltage of the LED strings.
- Configurable current amplitude.
- Up to 700mA average current on each string.
- Configurable current ripple.
- I²C interface with configurable 10-bit slave address (with a default value of 0x15E) to increases the range of devices that can be connected to the bus line.

RGB LED Shield driving an LED wall washer

1.2 Key Features of the XMC1200 MCU series

- 32-bit ARM® Cortex™-M0, 32MHz.
- Hardware Interconnect Matrix.
- 16kB ~ 200kB Flash with ECC and 16kB RAM.
- Peripherals running up to 64MHz.
- Timer/PWM: CCU4, CCU8, POSIF.
- Analog-mixed Signal: 12-bit ADCs, 12-bit DACs, ACMPs.
- Communication: I^2C, SPI, Dual-/Quad-SPI, SCI, I2S, LIN.
- Application specific: LED Color Control Engine, Touch.
- AES 128-bit secure loader for SW IP protection.
- Operating: 1.8 ~ 5.5Volt and -40° ~ 105°C.
- Free DAVE™ IDP and DAVE Apps (code library) open to 3rd party tools and the wide ARM® ecosystem.

1.3 Getting started

The RGB LED Shield uses high frequency peak-current control with fixed off-times to generate DC LED currents. Although this is highly efficient, low cost, and is suitable for high-speed dimming, it results in the output current being dependent on the input and output voltage ratio. The output current can be adjusted by configuring the peak-current reference and off-time parameters.

A virgin RGB LED Shield is pre-configured with safe peak-current reference and off-time parameters. With the safe parameters, the LED current will not be 'too high' at high input voltages.

The safe parameter values have been tested with LED loads that have a forward voltage of 6V at input voltages up to 48V. At this input, the pre-configured average LED current is measured up to 300mA.

Note: *LED strings that have a forward voltage lower than 6V and current capability lower than 300mA should not be connected without re-configuring the shield first.*

The safe parameters will however result in a discontinuous current with most LED strings and input voltages. For low-ripple continuous current, the off-time and peak-current reference parameters must be configured by the user once the LED lamp and input voltages have been selected.

Generally, the current in any of the strings should never exceed 1A, the peak-current reference parameters should be kept below 0x80, and off-time parameters should be kept above 0x10.

Attention: Improper configuration may result in permanent damage.

RGB LED Lighting Shield Board Description

2 Board Description

The RGB LED Shield can be controlled by programming a master Arduino board, such as the Arduino Uno R3 or the XMC1100 Boot Kit.

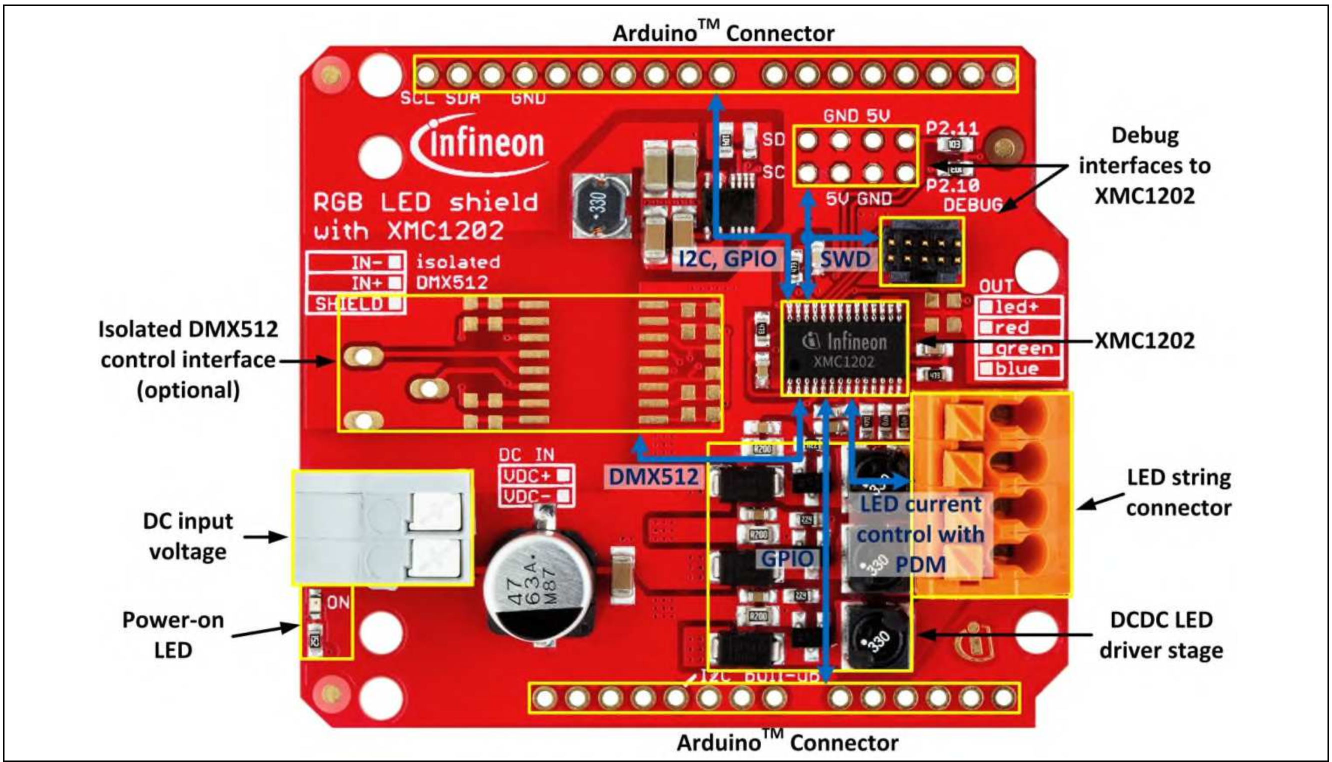

Figure 2 RGB LED Shield Interfaces

2.1 Specifications

Dimensions	2.7 x 2.1 inches (standard Arduino footprint)
Input voltage	up to 48V
Output Current per string	up to 1A peak and 700mA average
Order Number	KIT_LED_XMC1202_AS_01

2.2 Programming Access

The on-board XMC1202 microcontroller can be programmed over SWD via the debug interfaces using a J-Link debug probe from Segger that supports ARM® Cortex™-M0 (Figure 3).

Flash content can be updated over SWD using the TASKING debugger integrated in DAVE™.

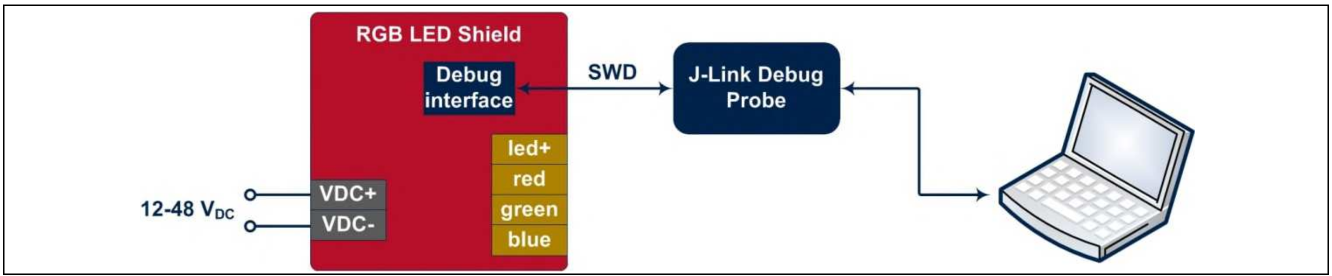

Figure 3 Segger J-Link debug probe connected to the RGB LED Shield

2.3 Schematics and Layout

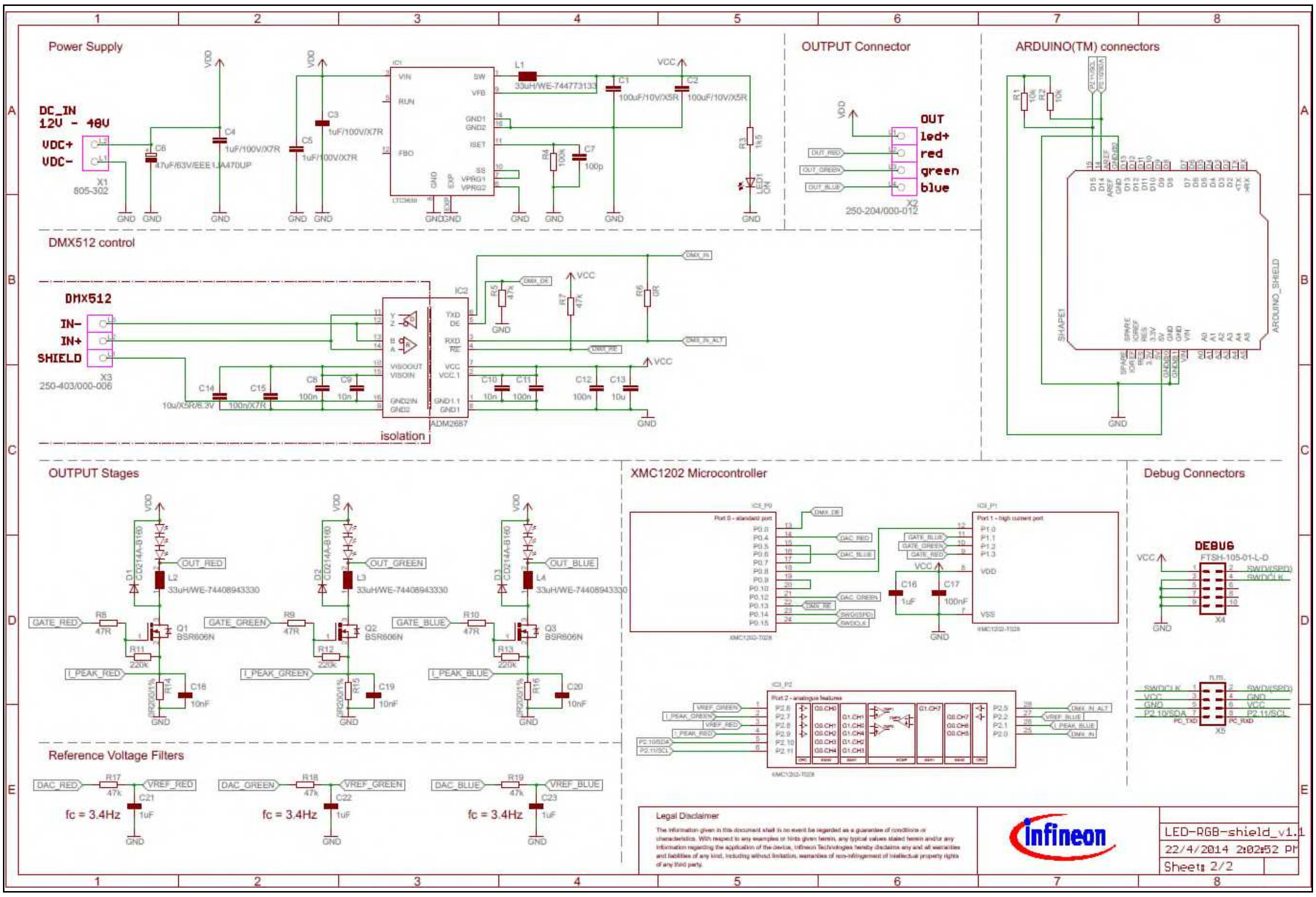

Figure 4 RGB LED Shield – Schematics

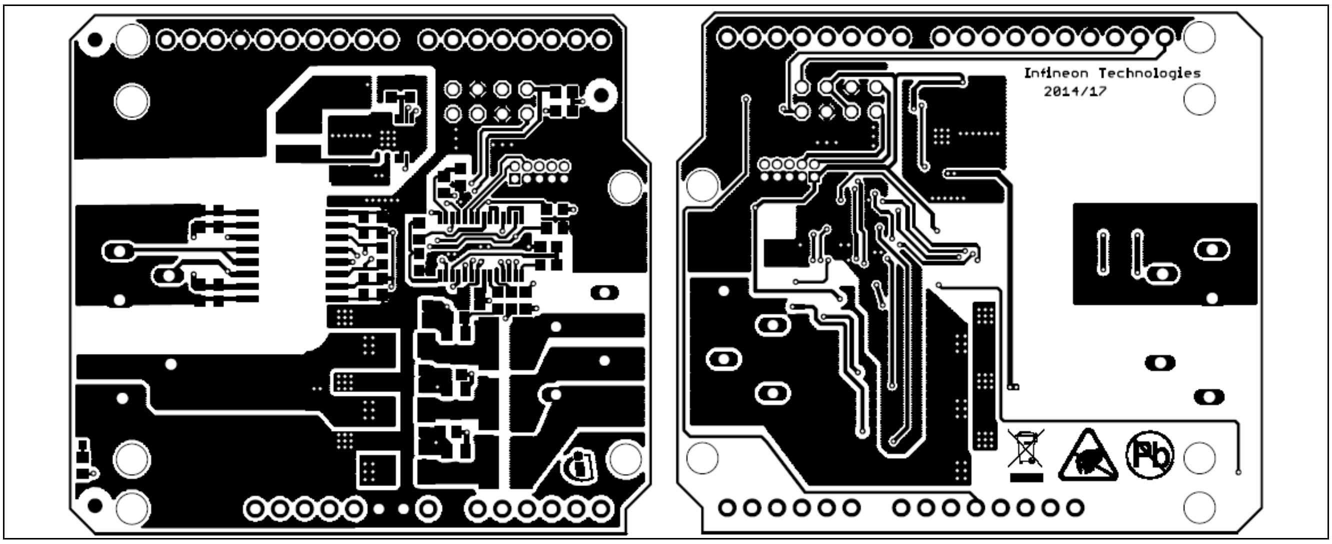

Figure 5 RGB LED Shield – Top and Bottom Layers

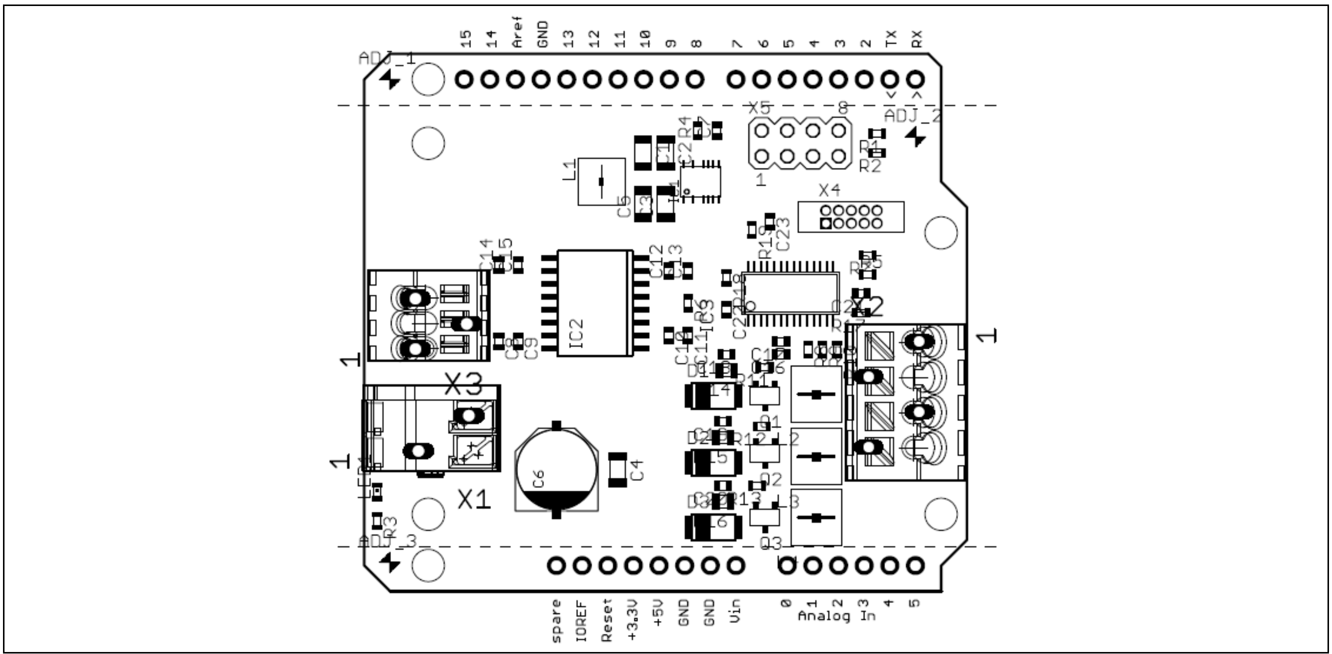

Figure 6 RGB LED Shield – Components

Partlist

Exported from LED-RGB-shield_v1.1.brd at 22/4/2014 2:05:49 PM

Updated on 23/5/2014

EAGLE Version 5.7.0 Copyright (c) 1988-2010 CadSoft

Part	Value	Package	Library	Position (mil)	Orientation	Comment	MPN
C1	100uF/10V/X5R	C1206	rcl	(1087.5 1712.5)	R270	-	TDK C3216X5R1A107M160AC
C2	100uF/10V/X5R	C1206	rcl	(1175 1712.5)	R270	-	TDK C3216X5R1A107M160AC
C3	1uF/100V/X7R	C1206	rcl	(1175 1512.5)	R90	-	AVX 12061C105K4Z2A
C4	1uF/100V/X7R	C1206	rcl	(987.5 475)	R270	-	AVX 12061C105K4Z2A
C5	1uF/100V/X7R	C1206	rcl	(1087.5 1512.5)	R90	-	AVX 12061C105K4Z2A
C6	47uF/63V/EEE1JA470UP	PANASONIC_F	rcl	(750 475)	R90	-	Panasonic EEE1JA470UP
C7	100p	C0603	rcl	(1375 1800)	R90	-	Multicomp MC0603N101J500CT
C8	100n	C0603	rcl	(525 975)	R270	DMX512	AVX 06033C104JAT2A
C9	10n	C0603	rcl	(600 975)	R270	DMX512	Multicomp MC0603B103K160CT
C10	10n	C0603	rcl	(1187.5 1000)	R270	DMX512	Multicomp MC0603B103K160CT
C11	100n	C0603	rcl	(1262.5 1000)	R270	DMX512	AVX 06033C104JAT2A
C12	100n	C0603	rcl	(1187.5 1250)	R90	DMX512	AVX 06033C104JAT2A
C13	10u	C0603	rcl	(1262.5 1250)	R90	DMX512	Murata GRM188R60J106ME47D
C14	10u/X5R/6.3V	C0603	rcl	(525 1275)	R90	DMX512	Murata GRM188R60J106ME47D
C15	100n/X7R	C0603	rcl	(600 1275)	R90	DMX512	AVX 06033C104JAT2A
C16	1uF	C0603	rcl	(1625 925)	R180	-	Murata GRM188R60J105KA01D
C17	100nF	C0603	rcl	(1625 975)	R180	-	AVX 06033C104JAT2A
C18	10nF	C0603	rcl	(1412.5 925)	R180	-	Multicomp MC0603B103K160CT
C19	10nF	C0603	rcl	(1400 662.5)	R180	-	Multicomp MC0603B103K160CT
C20	10nF	C0603	rcl	(1400 412.5)	R180	-	Multicomp MC0603B103K160CT
C21	1uF	C0603	rcl	(1937.5 1162.5)	R180	-	Murata GRM188R60J105KA01D
C22	1uF	C0603	rcl	(1412.5 1100)	R270	-	Murata GRM188R60J105KA01D
C23	1uF	C0603	rcl	(1581.252 1443.752)	R270	-	Murata GRM188R60J105KA01D
D1	CD214A-B160	DO-214AC	diode	(1362.5 762.5)	R0	-	Bourns CD214A-B160LF
D2	CD214A-B160	DO-214AC	diode	(1362.5 500)	R0	-	Bourns CD214A-B160LF
D3	CD214A-B160	DO-214AC	diode	(1362.5 250)	R0	-	Bourns CD214A-B160LF
IC1	LTC3630	MSE16(12)	LinearTechnology	(1312.5 1600)	R0	-	Linear Technology LTC3630EMSE#PBF
IC2	ADM2687	SO-16W	AnalogDevice	(900 1125)	R90	DMX512	Analog Devices ADM2687EBRIZ
IC3	XMC1202-T028	P-TSSOP28-16	XMC1000	(1662.5 1162.5)	R90	-	Infineon Technologies XMC1202-T028X0016 AA
L1	33uH/WE-744773133	WE-PD2_S	wuerth-elektronik	(925 1600)	R270	-	Wuerth Elektronik WE-744773133
L2	33uH/WE-74408943330	WE-SPC_48XX	PowerMagnetics	(1762.5 768.752)	R0	-	Wuerth Elektronik WE-74408943330
L3	33uH/WE-74408943330	WE-SPC_48XX	PowerMagnetics	(1762.5 525)	R0	-	Wuerth Elektronik WE-74408943330
L4	33uH/WE-74408943330	WE-SPC_48XX	PowerMagnetics	(1762.5 287.5)	R0	-	Wuerth Elektronik WE-74408943330
LED1	ON	CHIP-LED0603	led	(50 387.5)	R0	-	Wuerth Elektronik 150060RS75000
Q1	BSR606N	SC59	infineon_discrete	(1562.5 762.5)	R180	-	Infineon Technologies BSR606N
Q2	BSR606N	SC59	infineon_discrete	(1562.5 537.5)	R180	-	Infineon Technologies BSR606N
Q3	BSR606N	SC59	infineon_discrete	(1562.5 287.496)	R180	-	Infineon Technologies BSR606N
R1	10k	R0603	rcl	(2000 1787.5)	R180	-	Bourns CR0603-JW-103ELF
R2	10k	R0603	rcl	(2000 1712.5)	R180	-	Bourns CR0603-JW-103ELF
R3	1k5	R0603	rcl	(50 275)	R270	-	Bourns CR0603-JW-152GLF
R4	100k	R0603	rcl	(1300 1800)	R90	-	Bourns CR0603-JW-104ELF
R5	47k	R0603	rcl	(1962.5 1237.5)	R0	DMX512	Bourns CR0603-JW-473GLF
R6	0R	R0603	rcl	(1262.5 1125)	R270	DMX512	Bourns CR0603-J/-000ELF
R7	47k	R0603	rcl	(1962.5 1312.5)	R180	DMX512	Bourns CR0603-JW-473GLF
R8	47R	R0603	rcl	(1731.252 943.752)	R270	-	Bourns CR0603-JW-470GLF
R9	47R	R0603	rcl	(1787.5 943.752)	R270	-	Bourns CR0603-JW-470GLF
R10	47R	R0603	rcl	(1843.752 943.748)	R270	-	Bourns CR0603-JW-470GLF
R11	220k	R0603	rcl	(1562.5 875)	R180	-	Bourns CR0603-FX-2203ELF
R12	220k	R0603	rcl	(1550 643.752)	R180	-	Bourns CR0603-FX-2203ELF
R13	220k	R0603	rcl	(1531.252 412.5)	R180	-	Bourns CR0603-FX-2203ELF
R14	0R200/1%	R0805	rcl	(1412.5 862.5)	R180	-	Bourns CRL0805-FW-R200ELF
R15	0R200/1%	R0805	rcl	(1400 600)	R180	-	Bourns CRL0805-FW-R200ELF
R16	0R200/1%	R0805	rcl	(1400 350)	R180	-	Bourns CRL0805-FW-R200ELF
R17	47k	R0603	rcl	(1937.5 1087.5)	R180	-	Bourns CR0603-JW-473GLF
R18	47k	R0603	rcl	(1412.5 1225)	R270	-	Bourns CR0603-JW-473GLF
R19	47k	R0603	rcl	(1512.5 1412.5)	R270	-	Bourns CR0603-JW-473GLF
SHAPE1	ARDUINO_SHIELD	XMC1100_FOR_ARDUINO	arduino	(0 0)	R0	-	
X1	805-302	P-805-302	con-wago_805	(262.5 625)	R180	-	WAGO 805-302
X2	250-204/000-012	P-250-204	con-wago_250	(2112.5 737.5)	R0	-	WAGO 250-204/000-012
X3	250-403/000-006	P-250-403	con-wago_250	(250 1075)	R180	DMX512	WAGO 250-403/000-006
X4	FTSH-105-01-L-D	FTSH-105-01-L-DV-THROUGHHOLE	TIMM1	(1800 1437.5)	R0	-	Samtec FTSH-105-01-L-DV
X5	n.m.	MA04-2	con-lstb	(1700 1750)	R0	-	Multicomp 10-89-7082

Figure 7 RGB LED Shield – BOM

Getting Started

3 Getting Started

You can bring YOUR LED lamp to life in seven simple steps.

STEP 1. Choose a high-power light engine

a. Maximum three channels (e.g. RGB)

b. Minimum 300mA LED current rating

NOTE: If the current rating is <300mA you can easily configure your RGB LED Lighting Shield using the instructions in chapter 6 (Parameter Setup for YOUR LED Lamp).

c. Maximum 48V forward voltage per LED channel

STEP 2. Choose a DC adapter

a. Input voltage to the RGB LED Lighting Shield: 12V ~ 48V DC

b. Maximum 48V forward voltage per LED channel

NOTE: DC input voltage to the RGB LED Lighting Shield should be higher than the forward voltage of the LED channels.

STEP 3. Solder pin headers on the RGB LED Lighting Shield

STEP 4. Connect the RGB LED Lighting Shield to

a. Arduino Uno R3

b. XMC1100 Boot Kit

STEP 5. Program Arduino Uno R3 or XMC1100 Boot Kit

a. Example Sketches and projects: www.infineon.com/arduino

i. Upload *RGBLED_2_SAFE.ino* to Arduino Uno R3

ii. Upload *RGBLED_2_Safe_XMC11.zip* to XMC1100 Boot Kit

STEP 6. Connect the DC adapter to the RGB LED Lighting Shield

STEP 7. Turn on the power

I^2C Master-Slave Communication Protocol

4 I^{2}C Master-Slave Communication Protocol

Command words have been defined in software. Parameters can be changed by sending these commands from the master to the RGB LED Lighting Shield. These commands can be sent to the shield from the master board using pre-defined functions.

4.1 Brief Description of I^{2}C Functions

The I^{2}C commands together with the required data can be sent to the RGB LED Lighting Shield from the master board using the functions provided. These functions encapsulate the data in the necessary format for transfer via the I^{2}C communication protocol.

The functions are provided for the Arduino Uno R3 and the XMC1100 Boot Kit.

The RGB LED Shield's I^{2}C address is a 10-bit address and is pre-configured to be 0x15E. To address it, the master will send 2 bytes of address:

- The first 7 bits of the first byte are 11110XX, of which XX are the two most significant bytes of the 10-bit address. The 8th bit determines the read or write direction of the data transfer.
- The second byte is the lower 8-bits of the address.

Write functions

I2CWRITE2BYTES, I2CWRITE6BYTES, I2CWRITE9BYTES, I2CWRITE_DIRECTACCESS, I2CCHANGEADDRESS, I2CDMX and I2CSAVEPARAM

- The I^{2}C START condition is sent, followed by the 1st byte of the RGB LED Shield address byte, a 'zero' bit to indicate a transmission request and the 2nd address byte.
- The appropriate command word is then sent, followed by the data and a STOP condition to terminate the transfer. Data is always put on the SDA line as a byte that is 8-bits long. 16-bit data is sent as 2 bytes and 32-bit data as 4 bytes.

Read functions

I2CREAD, I2CREAD_DIRECTACCESS

- The I^{2}C START condition is sent, followed by the 1st byte of the RGB LED Shield address byte, a 'zero' bit to indicate a transmission request and the 2nd address byte.
- The appropriate command word is then sent.
- A repeated START condition is then sent followed by the 1st byte of the RGB LED Shield address byte, a 'zero' bit, the 2nd address byte and the 1st address byte with a 'one' bit to request for data.
- Acknowledge pulses are subsequently sent.
- A STOP condition is sent to terminate the transfer.

Note: *A detailed description of each function can be found in the Appendix.*

2.1.1 Command Overview Table

The following tables provides a short description of the commands that can be sent with the functions.

Table 2 Commands and Functions

I^2C Commands	Description	I^2C Function used
INTENSITY_RED	Change relative colour intensity of red channel	I2CWRITE2BYTES
INTENSITY_GREEN	Change relative colour intensity of green channel	I2CWRITE2BYTES
INTENSITY_BLUE	Change relative colour intensity of blue channel	I2CWRITE2BYTES
INTENSITY_RGB	Change relative colour intensity of red, green and blue channels	I2CWRITE6BYTES
CURRENT_RED	Change peak-current reference of red channel	I2CWRITE2BYTES
CURRENT_GREEN	Change peak-current reference of green channel	I2CWRITE2BYTES
CURRENT_BLUE	Change peak-current reference of blue channel	I2CWRITE2BYTES
OFFTIME_RED	Change off-time of red channel	I2CWRITE2BYTES
OFFTIME_GREEN	Change off-time of green channel	I2CWRITE2BYTES
OFFTIME_BLUE	Change off-time of blue channel	I2CWRITE2BYTES
WALKTIME	Change walktime of red, green and blue channels	I2CWRITE2BYTES
DIMMINGLEVEL	Change brightness level	I2CWRITE2BYTES
FADERATE	Change time taken to dim to 0%	I2CWRITE2BYTES
CHANGEADDRESS	Change address of RGB LED Shield	I2CWRITE2BYTES
DMXOFF	Disable DMX512 control	I2CDMX
DMXON	Enable DMX512 control	I2CDMX
DMXSLOT	Change first relevant slot of DMX512 control	I2CWRITE2BYTES
DMX8BIT	Read 8-bits of colour information from each DMX512 slot	I2CWRITE6BYTES
DMX16BIT	Read 16-bits of colour information from each DMX512 slot	I2CWRITE12BYTES
READ_CONFIG	Query if RGB LED Shield has been configured	I2CREAD
READ_INTENSITY_RED	Request for relative colour intensity of red channel	I2CREAD
READ_INTENSITY_GREEN	Request for relative colour intensity of green channel	I2CREAD
READ_INTENSITY_BLUE	Request for relative colour intensity of blue channel	I2CREAD
READ_CURRENT_RED	Request for peak current reference of red channel	I2CREAD
READ_CURRENT_GREEN	Request for peak current reference of green channel	I2CREAD
READ_CURRENT_BLUE	Request for peak current reference of blue channel	I2CREAD
READ_OFFTIME_RED	Request for off-time of red channel	I2CREAD
READ_OFFTIME_GREEN	Request for off-time of green channel	I2CREAD
READ_OFFTIME_BLUE	Request for off-time of blue channel	I2CREAD
READ_WALKTIME	Request for linear walk time	I2CREAD
READ_DIMMINGLEVEL	Request for dimming level	I2CREAD
READ_FADERATE	Request for rate of dimming	I2CREAD
READ_DMX	Query if DMX512 control is enabled	I2CREAD
READ_DMXSLOT	Request for first relevant slot in DMX512 control	I2CREAD
READ_DMXBIT	Request for number of bits of colour information expected from DMX512 control	I2CREAD
READ_DMXREDH	Request for slot which stores upper 8-bits of red colour information	I2CREAD
READ_DMXREDL	Request for slot which stores lower 8-bits of red colour information	I2CREAD

I^2C Commands	Description	I^2C Function used
READ_DMXGREENH	Request for slot which stores upper 8-bits of green colour information	I2CREAD
READ_DMXGREENL	Request for slot which stores lower 8-bits of green colour information	I2CREAD
READ_DMXBLUEH	Request for slot which stores upper 8-bits of blue colour information	I2CREAD
READ_DMXBLUEL	Request for slot which stores lower 8-bits of blue colour information	I2CREAD
DIRECTACCESS_READ	Request for value contained in a specific register	I2CREAD_DIRECTACCESS
DIRECTACCESS_MOVE	Move value into a specific register	I2CWRITE_DIRECTACCESS
DIRECTACCESS_AND	Bitwise AND operation on a user specified value and the value in a specific register	I2CWRITE_DIRECTACCESS
DIRECTACCESS_OR	Bitwise OR operation on a user specified value and the value in a specific register	I2CWRITE_DIRECTACCESS
SAVEPARAMETERS	Save current parameters to Flash memory	I2CSAVEPARAM

4.2 Command Description

4.2.1 Colour Intensity (INTENSITY_RED, INTENSITY_GREEN, INTENSITY_BLUE, INTENSITY_RGB)

The colour intensities of the Red, Green and Blue colour channels on the RGB LED Lighting Shield can be changed.

Three of the 9 available BCCU channels on the XMC1202 microcontroller on-board the RGB LED Shield are used to control the colour intensities. A change in the relative colour intensity in any of three channels will change the colour of the lamp attached to the shield. Colour intensities are 12-bit values. The maximum intensity of each channel is 0xFFF.

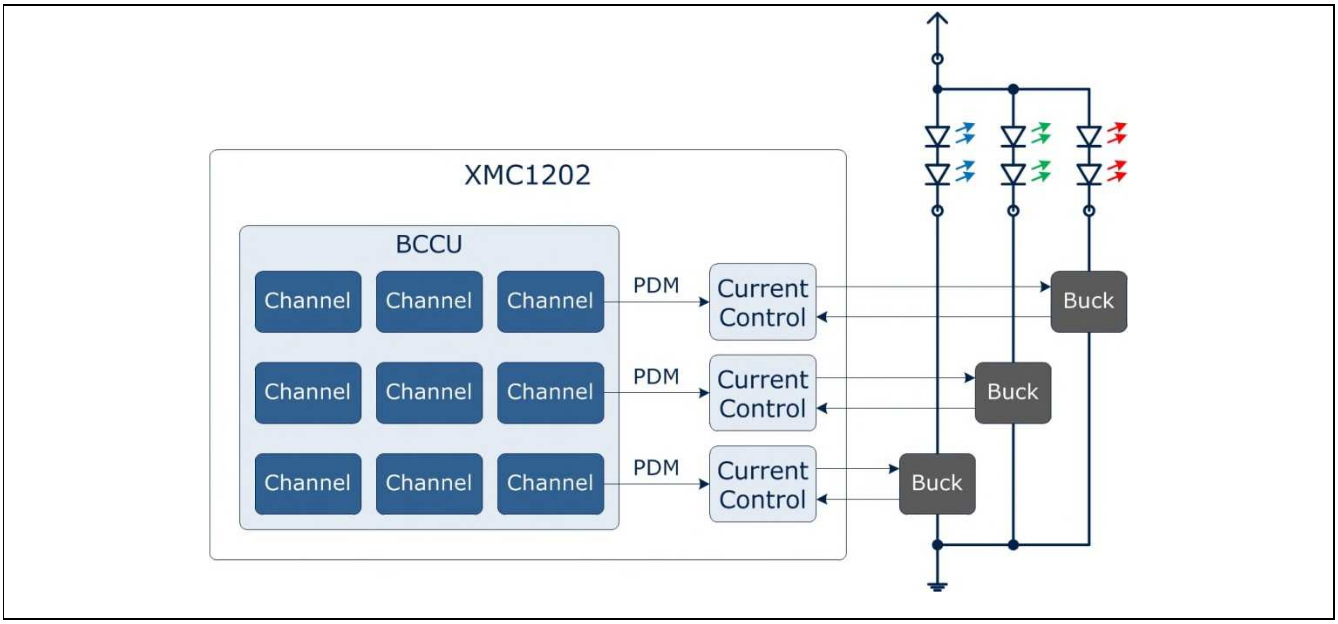

Figure 8 PDM Channels in the microcontroller on board the RGB LED Shield

INTENSITY_RED

Changes the relative colour intensity of the red channel.
To set the red channel to maximum intensity, send the following from the master:

```
I2CWRITE2BYTES(INTENSITY_RED, 0xFFF);
```

INTENSITY_GREEN

Changes the relative colour intensity of the green channel.
To set the green channel to maximum intensity, send the following from the master:

```
I2CWRITE2BYTES(INTENSITY_GREEN, 0xFFF);
```

INTENSITY_BLUE

Changes the relative colour intensity of the blue channel.
To set the blue channel to maximum intensity, send the following from the master

```
I2CWRITE2BYTES(INTENSITY_BLUE, 0xFFF);
```

INTENSITY_RGB

Changes the relative colour intensities of the red, green and blue channel.
To enable white light, send the following from the master

```
I2CWRITE2BYTES(INTENSITY_RGB, 0xFFF);
```

Recommended Colour Scheme

To ensure constant lamp brightness for different colors, keep the sum of intensities of the three channels constant.

Table 3

Colour	Channel Intensity			Possible commands to be sent from the master
	Red	Green	Blue	

Red	0xFFF	0x000	0x000	`I2CWRITE2BYTES(ADDRESS, INTENSITY_RED, 0xFFF);`
				`I2CWRITE2BYTES(ADDRESS, INTENSITY_GREEN, 0x000);`
				`I2CWRITE2BYTES(ADDRESS, INTENSITY_BLUE, 0x000);`
				OR
				`I2CWRITE6BYTES(ADDRESS, INTENSITY_RGB, 0xFFF, 0x000, 0x000);`

Green	0x000	0xFFF	0x000	`I2CWRITE2BYTES(ADDRESS, INTENSITY_RED, 0x000);`
				`I2CWRITE2BYTES(ADDRESS, INTENSITY_GREEN, 0xFFF);`
				`I2CWRITE2BYTES(ADDRESS, INTENSITY_BLUE, 0x000);`
				OR
				`I2CWRITE6BYTES(ADDRESS, INTENSITY_RGB, 0x000, 0xFFF, 0x000);`

Blue	0x000	0x000	0xFFF	`I2CWRITE2BYTES(ADDRESS, INTENSITY_RED, 0x000);`
				`I2CWRITE2BYTES(ADDRESS, INTENSITY_GREEN, 0x000);`
				`I2CWRITE2BYTES(ADDRESS, INTENSITY_BLUE, 0xFFF);`
				OR
				`I2CWRITE6BYTES(ADDRESS, INTENSITY_RGB, 0x000, 0x000, 0xFFF);`

Colour	Channel Intensity			Possible commands to be sent from the master
	Red	Green	Blue	
Yellow	0x800	0x800	0x000	`I2CWRITE6BYTES(ADDRESS, INTENSITY_RGB, 0x800, 0x800, 0x000)`
Cyan	0x000	0x800	0x800	`I2CWRITE6BYTES(ADDRESS, INTENSITY_RGB, 0x000, 0x800, 0x800)`
Magenta	0x800	0x000	0x800	`I2CWRITE6BYTES(ADDRESS, INTENSITY_RGB, 0x800, 0x000, 0x800)`
White	0x555	0x555	0x555	`I2CWRITE6BYTES(ADDRESS, INTENSITY_RGB, 0x555, 0x555, 0x555)`

4.2.2 Peak Current Reference (CURRENT_RED, CURRENT_GREEN, CURRENT_BLUE)

The LED current can be controlled by the RGB LED shield. When attached to the shield, the LED lamp is connected to a 3-channel DCDC buck LED driver.

An inductor, Schottky diode and MOSFET are used, in an inverted buck topology, to control the LED current with high efficiency. As with every DC-DC buck driver, this design results in ripples in the LED current. In the RGB LED shield, the ripple frequency is approximately 1-1.5 MHz to support fast modulation dimming and achieve high power density.

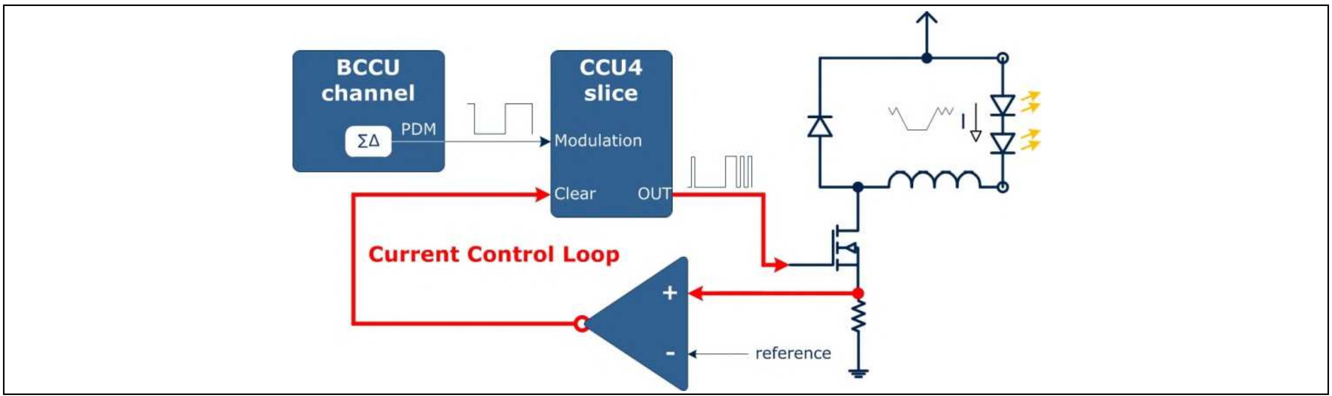

Figure 9 Peak Current Control

To adjust the LED current, the potential before the shunt resistor is fed into an on-chip comparator. The inductor in the setup causes the LED current to increase linearly and proportionately to the input voltage. As the current increases, the potential before the shunt resistor increases. When this potential exceeds the peak current reference value, the MOSFET is switched off by the MOSFET control output signal which switches to 0V. Current will continue to flow through the free-wheeling diode as the inductor’s magnetic field collapses. During this time, the current decreases linearly and proportionately to the forward voltage of the LED string. The process restarts when the MOSFET is switched on after a fixed off-time.

The RGB LED Shield will change the peak current reference parameter when the CURRENT_RED, CURRENT_GREEN or CURRENT_BLUE commands and the 12-bit peak-current reference parameter are sent from the master. A reference value of 0xFFF corresponds to 5V, and 0x000 corresponds to 0V.

To calculate the reference voltage, use:

Reference Value / 4096 * 5V

The maximum peak current reference value that can be set is 0x80, which is approximately 0.15625V. Should a value greater than this be sent to the RGB LED Shield, the value will be ignored and peak current reference set to 0x80. This corresponds to a theoretical peak current of 781mA flowing through the MOSFET.

CURRENT_RED

Changes the peak current reference parameter of the red channel.

To change the reference value to approximately 0.12V, send the following from the master:

```
I2CWRITE2BYTES(ADDRESS, CURRENT_RED, 0x64); // 0.12 = 100 / 4096 * 5
```

CURRENT_GREEN

Changes the peak current reference parameter of the green channel.

To change the reference value to approximately 0.12V, send the following from the master:

```
I2CWRITE2BYTES(ADDRESS, CURRENT_GREEN, 0x64);
```

CURRENT_BLUE

Changes the peak current reference parameter of the blue channel.

To change the reference value to approximately 0.12V, send the following from the master:

```
I2CWRITE2BYTES(ADDRESS, CURRENT_BLUE, 0x64);
```

4.2.3 Off-Time (OFFTIME_RED, OFFTIME_GREEN, OFFTIME_BLUE)

This parameter adjusts the ripple of the LED current.

When the comparator in the shield detects that the current in the lamp has reached the peak current reference, the MOSFET is switched off. This off-state is extended for a fixed duration determined by the off-time parameter value. In this off-state, the circuit is switched off and the LED current decreases.

The smaller the off-time value, the shorter the off-state, the less the LED current decreases, leading to a valley current which is closer in value to the peak current. As a result, the ripple in the current is reduced.

Conversely, when the off-state is extended for a longer duration, the LED current falls more, resulting in a smaller valley current, and a larger ripple.

Ideally, the LED current should not exceed the peak-current reference.

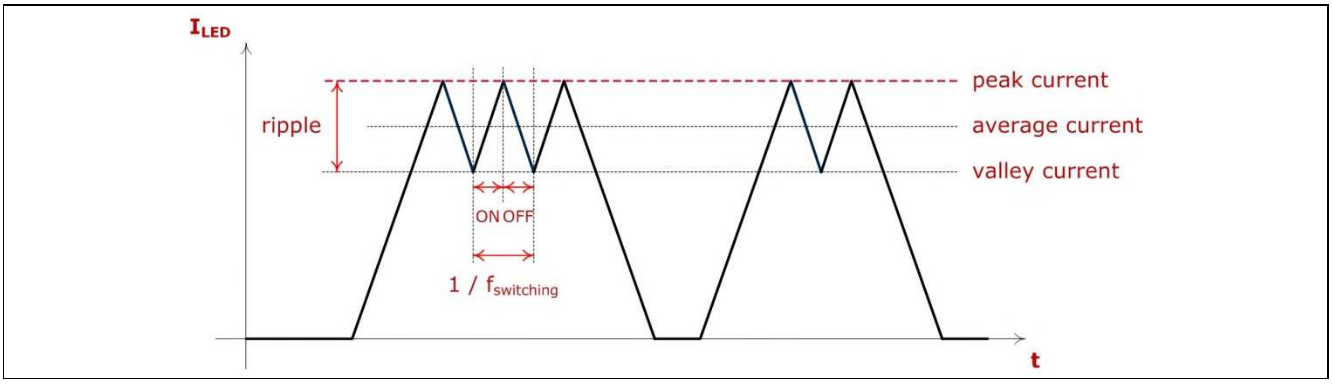

Figure 10 LED Current Ripple

Due to non-negligible propagation delays in the comparator and the connected on-chip circuits, the LED current peaks invariably exceed the peak-current reference. There is a time delay between the LED current reaching the peak-current reference value and the comparator detecting it. A short off-state can result in the current not dropping enough before the MOSFET is switched on again. The comparator may no longer be able to accurately detect the peak current reference, leading to exceedingly high currents. To avoid catastrophically high currents, the off-state is generated after the LED current has dropped below the peak reference level.

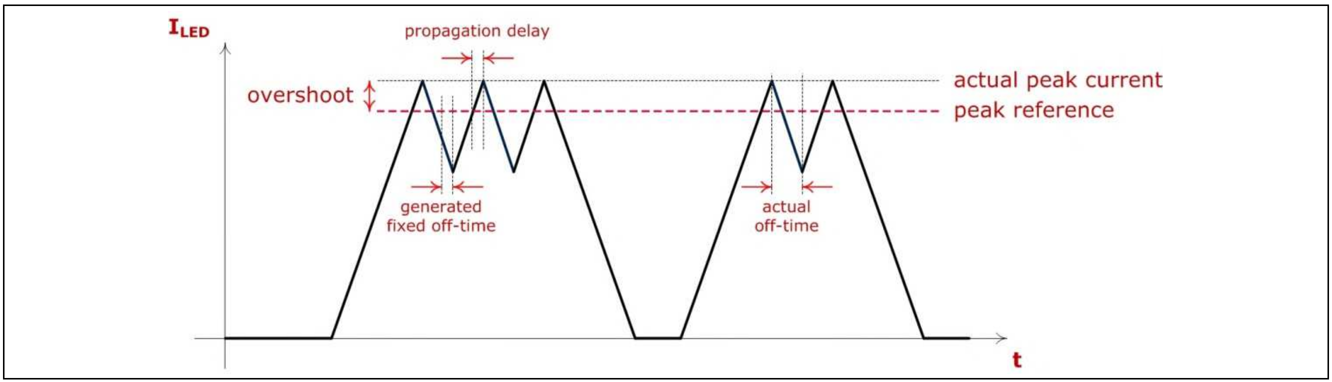

Figure 11 Propagation Delay leading to LED Current Over-shooting Peak Current Reference

The MOSFET will remain off while a counter counts up to the off-time parameter value. When the off-time value is reached, the counter resets and the MOSFET is switched on.

The counter counts at a frequency of 64MHz (resolution of 15.625ns).

The circuit will be in the generated off-state for 1µs when the off-time value is set to:

0x40 (1 / 64M * 64)

OFFTIME_RED

Changes the fixed off-time parameter of the red channel.
To change the off-time to 1µs, send the following from the master:

```
I2CWRITE2BYTES(ADDRESS, OFFTIME_RED, 0x40);
```

OFFTIME_GREEN

Changes the fixed off-time parameter of the green channel.
To change the off-time to 1µs, send the following from the master:

```
I2CWRITE2BYTES(ADDRESS, OFFTIME_GREEN, 0x40);
```

OFFTIME_BLUE

Changes the fixed off-time parameter of the blue channel.
To change the off-time to 1µs, send the following from the master:

```
I2CWRITE2BYTES(ADDRESS, OFFTIME_BLUE, 0x40);
```

4.2.4 Walk time (WALKTIME)

A linear walk is used to smoothly change the colour intensities. The intensities change linearly over time. The time taken for the channels to reach their target intensities is called the linear walk time. The linear walk time can be adjusted.

The RGB LED Shield calculates the actual linear walk time with the formula:

Linear Walk Time = WALKTIME * 0.01024

A WALKTIME value of 0x10 means that the actual linear walk time is 164ms. The channels will take 164ms to reach their target intensities.

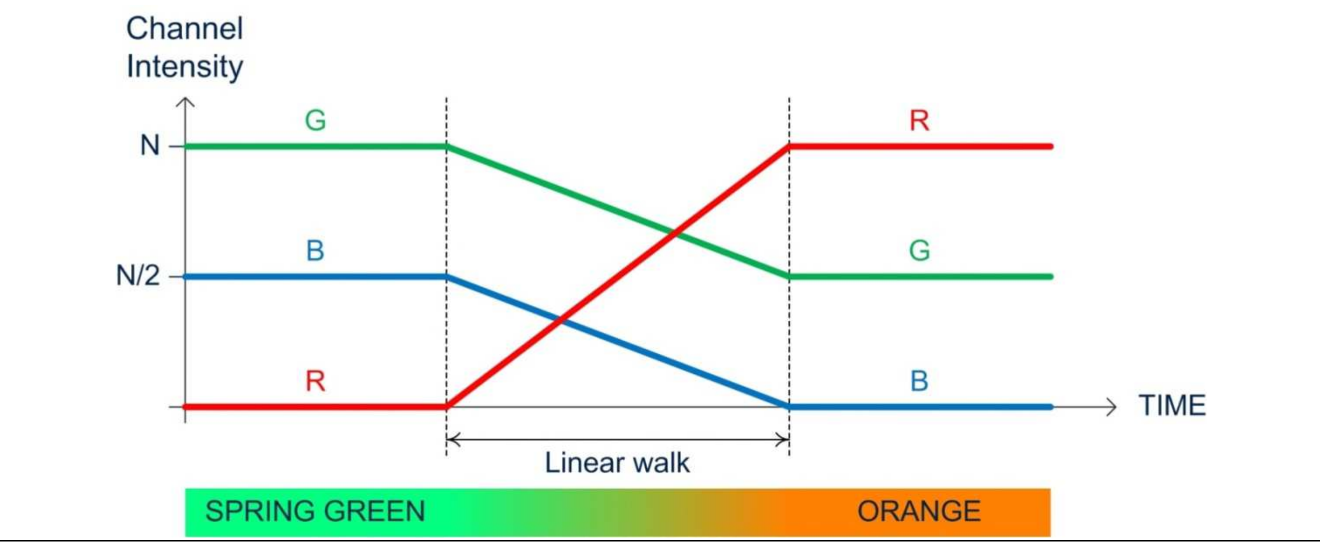

Figure 12 Walk time – Time taken for channels to reach their target intensities

WALKTIME

This command can only be used to change the WALKTIME parameter for all three channels together.

To change the linear walk time to 164ms, send the following from the master:

```
I2CWRITE2BYTES(ADDRESS, WALKTIME, 0x10);
```