High Brightness LED Controller using the MC9RS08KA2

Designer Reference Manual

RS08
Microcontrollers

DRM080
Rev. 0
5/2006

freescale.com
High Brightness LED Controller using the MC9RS08KA2
Designer Reference Manual

by: Dennis Lui, Vincent Ko
Freescale Semiconductor, Inc.
Hong Kong

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify that you have the latest information available, refer to http://www.freescale.com

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

**Revision History**

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision Level</th>
<th>Description</th>
<th>Page Number(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/2006</td>
<td>0</td>
<td>Initial release</td>
<td>N/A</td>
</tr>
</tbody>
</table>
# Table of Contents

## Chapter 1
### Introduction
1.1 Introduction .................................................................................................................. 7
1.2 Features .......................................................................................................................... 7
1.3 System Overview .......................................................................................................... 7
1.4 Freescale’s New Generation Ultra Low Cost MCU ................................................... 8

## Chapter 2
### Control Theory
2.1 Switching Regulator ....................................................................................................... 11
2.2 Buck Converter Basics .................................................................................................... 11

## Chapter 3
### Hardware Implementation
3.1 Introduction .................................................................................................................. 13
3.2 Functional Pin Assignment ............................................................................................. 13
3.3 Hardware Functions ....................................................................................................... 13
3.3.1 PWM Control ............................................................................................................ 14
3.3.2 Feedback Loop .......................................................................................................... 15
3.3.3 Reference Voltage ..................................................................................................... 16
3.3.4 User Interface ........................................................................................................... 17
3.3.5 Background Debug Interface .................................................................................... 17
3.4 Design Procedures ....................................................................................................... 17
3.4.1 High Brightness LED .............................................................................................. 17
3.4.2 Current Sense Resistor ............................................................................................ 17
3.4.3 RC Network .............................................................................................................. 17
3.4.4 Inductor Value .......................................................................................................... 18
3.4.5 Output Capacitor ....................................................................................................... 18
3.4.6 P-Channel MOSFET ............................................................................................... 18
3.4.7 Schottky Diode Rating ............................................................................................ 18
3.4.8 PCB Layout Guidelines ............................................................................................ 18

## Chapter 4
### Software Implementation
4.1 Introduction .................................................................................................................. 19
4.2 Software Design ............................................................................................................ 19
4.2.1 Main Software Flow .................................................................................................. 19
4.2.2 Initialization .............................................................................................................. 20
4.2.3 KBI Detection .......................................................................................................... 20
4.2.4 Feedback Comparison .............................................................................................. 21
4.2.5 PWM Generation ..................................................................................................... 22
# Table of Contents

## Chapter 5
### Testing

5.1 Measurement Data ........................................................................................................... 23
5.2 Customization ...................................................................................................................... 24
5.2.1 Hardware ......................................................................................................................... 24

## Chapter 6
### Demo Setup

6.1 Introduction .......................................................................................................................... 25
6.2 Default Settings ..................................................................................................................... 25
6.3 Programming and Running the HB-LED Demo ................................................................. 25
6.4 Troubleshooting .................................................................................................................... 26

## Appendix A.
### Schematic

## Appendix B.
### Program Listing
Chapter 1
Introduction

1.1 Introduction

This document describes a reference design of a high brightness LED control solution using the MC9RS08KA2 microcontroller.

Recently, light emitting diodes (LEDs) have become very popular in general lighting area as a replacement technology for halogen low-voltage lighting. Customers are quickly recognizing the advantages of using LED lighting, which include long operating life, no fragile glass, no mercury, and low voltage DC operated.

In general, LEDs have a nonlinear I-V behavior and thus current limitation is required to prevent the power dissipation to exceed a maximum limit. Thus, the ideal source for LED driving is a constant current source. The concept of this application is a MCU based LED driver with closed-loop current control. A compact LED light source with dimming control is implemented to demonstrate the advantages of using MCU to drive a high brightness LED with different average current settings.

All hardware schematic diagrams and firmware source codes are available as reference materials.

1.2 Features

- High brightness LED driver with 350mA current driving capability
- Control a buck converter to regulate supply voltage to match with different LED forward voltages
- Up to 80% efficiency
- Internally generated PWM switching frequency
- Feedback control on LED forward current through a current sense resistor
- Dimming control by a single button

1.3 System Overview

A block diagram of the system is shown in Figure 1-1.
1.4 Freescale’s New Generation Ultra Low Cost MCU

The MC9RS08KA2 (KA2) microcontroller unit (MCU) is an extremely low cost, small pin count device for home appliances, toys, small geometry, and LED control applications. This device is composed of standard on-chip modules including a very small and highly efficient RS08 CPU core, 62 bytes RAM, 2K bytes FLASH, an 8-bit modulo timer, keyboard interrupt, and analog comparator. The device is available in small 6- and 8-pin packages.

MC9RS08KA2 Features:
- Simplified S08 instruction set with added high-performance instructions
- 2048 bytes on-chip FLASH EEPROM
- 62 bytes on-chip RAM
- Internal clock source
- Up to 10-MHz internal bus operation
- Background debug system
- Power-saving modes
- Low-voltage detection
- 8-bit modulo timer
- Analog comparator
- Keyboard interrupt ports

Timer system features include:
- 8-bit up-counter
  - Free-running or 8-bit modulo limit
  - Software controllable interrupt on overflow
  - Counter reset bit (TRST)
  - Counter stop bit (TSTP)

Figure 1-1. System Block Diagram
Freescale’s New Generation Ultra Low Cost MCU

- Four software selectable clock sources for input to prescaler:
  - System bus clock — rising edge
  - Fixed frequency clock (XCLK) — rising edge
  - External clock source on the TCLK pin — rising edge
  - External clock source on the TCLK pin — falling edge
- Nine selectable clock prescale values:
  - Clock source divide by 1, 2, 4, 8, 16, 32, 64, 128, or 256

The analog comparator has the following features:
- Full rail-to-rail supply operation
- Less than 40 mV of input offset
- Less than 15 mV of hysteresis
- Selectable interrupt on rising edge, falling edge, or either rising or falling edges of comparator output
- Option to compare to fixed internal bandgap reference voltage
- Option to allow comparator output to be visible on a pin, ACMPO
- Remains operational in stop mode

The KBI features include:
- Each keyboard interrupt pin has individual pin enable bit
- Each keyboard interrupt pin is programmable as falling edge (or rising edge) only, or both falling edge and low level (or both rising edge and high level) interrupt sensitivity
- One software-enabled keyboard interrupt
- Exit from low-power modes
Chapter 2
Control Theory

2.1 Switching Regulator

A switching regulator regulates a current flow by chopping up the input voltage and controlling the average current by means of the duty cycle. When a higher load current is required by the load, the percentage of on-time is increased to accommodate the change. There are two basic types of switching regulators: forward-mode regulators and flyback-mode regulators. The name of each type is derived from the way the magnetic elements are used within the regulator.

Forward-mode switching regulators have four functional components: a power switch, a rectifier, a series inductor, and a capacitor (see Figure 2-1). The power switch may be a power transistor or a metal oxide semiconductor field-effect transistor (MOSFET) placed directly between the input voltage and the LC filter section. The shunt diode, series inductor, and shunt capacitor form an energy storage tank whose purpose is to store enough energy to maintain the load voltage and current over the entire off-time of the power switch. The power switch serves only to fill up the energy lost to the load during its off-time.

Flyback-mode switching regulators have the same four basic elements as the forward-mode regulators except that they have been rearranged in another configuration. In this design, forward-mode switching is used.

2.2 Buck Converter Basics

A “buck” or step-down converter is the most elementary forward-mode converter. Its basic schematic can be seen in Figure 2-1.

The operation of this regulator topology has two distinct time periods. The first one occurs when the series switch SW1 is on, the input voltage \( V_{in} \) is connected to the input of the inductor \( L \). The output of the inductor is the output voltage, and the rectifier (or catch diode) is reverse biased. During this period, since there is a constant voltage source connected across the inductor, the inductor current begins to linearly ramp upwards, as described by the following equation:

\[
I_{L(on)} = \frac{[(V_{in} - V_{out}) \times t_{on}]}{L}
\]  

(EQ 2-1)
During this “on” period, energy is stored within the core material in the form of magnetic flux. If the inductor is properly designed, there is sufficient energy stored to carry the requirements of the load during the “off” period.

The next period is the “off” period of the power switch. When the power switch turns off, the voltage across the inductor reverses its polarity and is clamped at one diode voltage drop below ground by the catch diode. The current now flows through the catch diode thus maintaining the load current loop. This removes the stored energy from the inductor. The inductor current during this time is:

\[ I_{L(\text{off})} = \frac{[V_{\text{out}} - V_D] \times t_{\text{off}}}{L} \]  \hspace{1cm} (EQ 2-2)

This period ends when the power switch is once again turned on. Regulation of the converter is accomplished by varying the duty cycle of the power switch according to the loading conditions. To achieve this, the power switch requires electronic control for proper operation. It is possible to describe the duty cycle as follows:

\[ d = \frac{t_{\text{on}}}{T} \]  \hspace{1cm} (EQ 2-3)

where \( T \) is the switching period.

For the buck converter with ideal components, the duty cycle can also be described as:

\[ d = \frac{V_{\text{out}}}{V_{\text{in}}} \]  \hspace{1cm} (EQ 2-4)
Chapter 3
Hardware Implementation

3.1 Introduction
The system consists of the 8-pin packaged KA2 and external components. There are four major external components: P-channel MOSFET, schottky diode, inductor, and capacitor which are used for switching regulation. The KA2 is used as a PWM controller to control the HB-LED brightness level. A simple DC to DC step-down converter (buck topology) is implemented to convert the 5V input supply to a current regulated output for LED driving. The integrated open source BDM (background debug mode) design allows users to program the MCU FLASH and debug applications via a USB connection.

3.2 Functional Pin Assignment
- PA0 – Reference Voltage Input
- PA1 – Feedback Input
- PA2 – Push Button Input
- PA3 – Dimming Control DIM0
- PA4 – PWM Output
- PA5 – Dimming Control DIM1

3.3 Hardware Functions
Figure 3-2 shows a block diagram of the HB-LED reference design. The hardware functions can be divided into the following parts:
- PWM Control
- Feedback Loop
- Reference Voltage
- User Interface
- Background Debug Interface
3.3.1 PWM Control

A fixed frequency step-down (buck) DC to DC converter is implemented with the KA2 for usage in high brightness LED driving applications. The KA2 consists of all active functions required to directly implement a step-down converter with a minimum number of external components. The KA2 enables control of PWM on power switch device to regulate the LED current. The PMOS device ON/OFF switching is controlled by MCU timer module and the frequency is set to around 30kHz. Baseline analog comparator output result, the MCU adjusts the PWM duty cycle in closed-loop to keep the LED driving current as a constant. Since a P-channel MOSFET device is used as the power switch, the PWM control signal should be generated in active low polarity, that means the power switch will turn on if the PWM signal is in LOW level.

A switching cycle is initiated by the falling edge of the PWM signal. At the moment the power switch Q1 is turned on, the inductor is connected to \( V_{DD} \) and the inductor current begins to ramp up linearly. When the PWM signal is in HIGH state, the power switch Q1 turns off and the current flows through the diode D to maintain the current loop. The capacitor \( C_O \) filters the regulated output and provides the converter loop stability. The converter is designed to operate in continuous conduction mode, which means the inductor current will not decrease to zero before the new cycle start. The average inductor current \( I_L \) is almost equal to the average loading current \( I_O \).
The external pull-up resistor R4 ensures the PMOS device Q1 is in OFF state during power up period, and prevents a high current surge through the LED before MCU controls the system.

### 3.3.2 Feedback Loop

The KA2 features an analog comparator which can be used to determine the LED current. The inverting input pin and non-inverting input pin are used for current sense feedback input and reference voltage input respectively.

Current limiting is implemented by monitoring the LED forward current, which is one of the key parameters affecting the performance of LED, particularly on the operating lifetime, so it is important to control and make sure the average current and peak current are also within the limits specified in the LED specification.

The LED forward current is converted to a voltage by using an external resistor connected in series with the LED. The voltage across the current sense resistor $R_S$ is directly proportional to the current through LED. This feedback voltage is compared against a reference by the analog comparator. When the feedback voltage is higher than the level at reference input, the PMOS switch ON time reduces to limit the current accumulated at the inductor $L$. In contrast, the PMOS switch ON time increases again to pump more current into the inductor when the feedback voltage is lower than the reference level. The upper and lower threshold levels are defined by the hysteresis range of the comparator.
Hardware Implementation

In steady-state condition, the LED forward current $I_{LED}$ is equal to the reference voltage $V_{REF}$ divided by the current sense resistor $R_S$.

The LED forward current is determined by the following equation:

$$I_{LED} = \frac{V_{REF}}{R_S}$$  \hspace{1cm} (EQ 3-1)

A simple low-pass filter is added between the current sense resistor $R_S$ and the feedback path to eliminate any high frequency noises coupling into the feedback pin. However, the response time of the filter should be fast enough to track the deviation of the voltage across the current sense resistor $R_S$.

In this reference design, the PWM switching frequency is around 30kHz and the filter cut-off frequency is set to 5kHz.

The following equation shows the relationship between the filter cut-off frequency and the RC values:

$$f = \frac{1}{2\pi RC}$$  \hspace{1cm} (EQ 3-2)

### 3.3.3 Reference Voltage

The reference voltage in the comparator non-inverting input is determined by a resistor network which consists of three resistors and the connection paths are controlled by the MCU. The pin DIM0 is always configured as output pin and set to output LOW. The resistor R2 connects to ground and generates a reference voltage divided from $V_{DD}$. This is the default reference voltage and corresponds to 100% brightness level. When the pin DIM1 is set to output LOW, R3 is also connected to ground and in parallel with R2, the reference voltage will be further reduced to a lower level. The DIM1 pin should be configured as input state if it is not used in dimming selection.

The equations for determining the reference voltage with the divider network are:

For High brightness:

$$V_{REF} = V_{DD}\left(\frac{R_2}{R_1 + R_2}\right)$$  \hspace{1cm} (EQ 3-3)

For Low brightness:

$$V_{REF} = V_{DD}\left(\frac{R_p}{R_1 + R_p}\right)$$  \hspace{1cm} (EQ 3-4)

where $R_p$ is equal to $R_2$ parallel with $R_3$. 

3.3.4 User Interface

A single push button, SW, is used as user interface to toggle the LED brightness level between 100% and 40% (default brightness level is 100%). The dimming control is accomplished by changing the LED forward current which is set by the reference voltage applied at the comparator’s non-inverting input.

Table 3-1. Dimming Control

<table>
<thead>
<tr>
<th>Control Pins</th>
<th>100% Brightness</th>
<th>40% Brightness</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIM0</td>
<td>Output LOW</td>
<td>Output LOW</td>
</tr>
<tr>
<td>DIM1</td>
<td>Hi-Z</td>
<td>Output LOW</td>
</tr>
</tbody>
</table>

3.3.5 Background Debug Interface

The on-board open source BDM provides a cost effective debug solution for Freescale RS08 MCUs. The module utilizes the modified version of popular open-source BDM solution from the Internet for in-circuit emulation and device programming. The board comprises a 12V DC-DC converter for RS08 FLASH programming support.

NOTE

In circuit debug can not be used for this application program since the BKGD pin is used for dimming control purpose. Use JP3 to switch between BDM and user mode.

3.4 Design Procedures

This section presents guidelines for selecting external components.

3.4.1 High Brightness LED

The system is designed to drive a high brightness LED with typical 350mA forward current at 3.5V forward voltage, i.e. the power dissipation on the LED is around 1.2W.

3.4.2 Current Sense Resistor

The value of the current sense resistor $R_S$ is determined by two factors: power dissipation on $R_S$ and the threshold level for comparator input. Smaller $R_S$ reduces power dissipation but the detection of feedback signal in comparator is more difficult.

Base on equation (EQ 3-1), setting $R_S$ to 1Ω, the feedback voltage is equal to 350mV.

Power dissipation on $R_S$ is around 120mW, $I^2R = (350\text{mA})^2 \times 1\Omega$, which is reasonable compared to LED power.

3.4.3 RC Network

Base on equations (EQ 3-3) and (EQ 3-4), set $V_{DD} = 5V$, $R1 = 10k\Omega$, $R2 = 750\Omega$, $R3 = 500\Omega$.

The reference voltage for high and low brightness settings are equal to 350mV and 146mV respectively. Low brightness setting is corresponds to around 40% brightness level.
3.4.4 Inductor Value

We can determine the inductor value by using the following equations:

\[ V_L = L \frac{di}{dt} \]  \hspace{1cm} (EQ 3-5)

\[ L = (V_S - V_O) \frac{D \times T}{I_{\text{ripple}}} \]  \hspace{1cm} (EQ 3-6)

\[ L = (5 - 3.85) \times \frac{3.85 \times 1}{0.35 \times 0.25} = 337 \mu \text{H} \]  \hspace{1cm} (EQ 3-7)

where: \( V_L \) is inductor voltage, \( V_S \) is source voltage, \( V_O \) is output voltage
\( D \) is conduction duty cycle, \( T \) is conduction period
\( I_{\text{ripple}} \) is inductor peak-to-peak ripple current, set to 25% of output current

The inductor current rating must be higher than the maximum peak current flowing through the inductor.

3.4.5 Output Capacitor

Low ESR (equivalent series resistance) aluminium or solid tantalum capacitor is recommended for low output ripple voltage. The ESR of the output capacitor and the inductor ripple current are two major factors contributing to the output ripple voltage.

\[ \text{Output ripple voltage} = \text{Inductor ripple current} \times \text{ESR} \]  \hspace{1cm} (EQ 3-8)

3.4.6 P-Channel MOSFET

For superior switching performance, a P-channel MOSFET device with low on-state resistance and low gate charge should be selected. The current rating must be at least 1.2 times greater than the maximum load current and the drain-to-source breakdown voltage should be at least 1.25 times the maximum input voltage. The P-channel MOSFET NTR4502 from On Semiconductor is selected in this design.

3.4.7 Schottky Diode Rating

A diode with a high forward voltage drop or long turn-on delay time should not be used, so a fast switching and low drop schottky diode is selected. The current and reverse voltage rating requirements should be similar to the P-channel MOSFET device. The schottky diode MBR130 from On Semiconductor is used in this design.

3.4.8 PCB Layout Guidelines

The PCB layout is very important for switching converter design and is critical to reduce noise and ensure specified performance. Care should be taken in PCB layout to avoid rapidly switching currents to generate voltage transients and affect the desired operation.

- Minimize inductance and ground loops, the length of the leads indicated by heavy lines should be kept as short as possible. (e.g. PMOS / coil / schottky diode.)
- Minimize the trace from the current sense resistor to the feedback input and keep it away from noise sources to avoid noise pick up.
- Locate the sensitive voltage reference divider network close to MCU.
- Single ground point or ground plane design should be used.
Chapter 4
Software Implementation

4.1 Introduction

The software is designed with a main loop to monitor the feedback voltage from the LED and generate a PWM controlled waveform for DC to DC switching operation. The PWM timing is constructed by the modulo timer overflow periods of PWM ON time & OFF time and the software overhead using for user interface detection & feedback loop control (see Figure 4-1). The timing used for each branch routine paths should be keep as constant such that the period of the generated PWM is stable and independent of the executed routine path.

4.2 Software Design

This section describes the design of the software blocks. The software description comprises these topics:

- Main Software Flow
- Initialization
- KBI Detection
- Feedback Comparison
- PWM Generation

4.2.1 Main Software Flow

Figure 4-2 shows the flow of the main program, it also indicates the functions that KA2 are to perform. The initialization subroutine initializes the modules of real-time interrupt, modulo timer, analog comparator, and keyboard interrupt.
4.2.2 Initialization

The Initialization routine initializes the MCU with following configurations:

- Initializes ICS and sets bus clock to 10MHz
- Initializes page register, disables COP and BKGD
- Enables LVI, and sets RTI to 128ms
- Initializes RAM variables
- Initializes GPIO
- Initializes KBI
- Initializes the ACMP
- Initializes modulo timer
  - Timer prescaler = 1, timer clock = 10MHz
  - Maximum of period = 25.6μs
  - Timer resolution = 100ns

4.2.3 KBI Detection

The state of the user interface is scanned every 128ms to see if any button press event has occurred. When a key press event is detected, it sets the corresponding “Pressed” flag. The output state of DIM1 pin will toggle to setup a new reference voltage for ACMP in the next 128ms period. This is to set up a key debounce period of 128ms.

The real-time interrupt function is used to generate a periodic interrupt for KBI scanning. The RTI is driven from the 1kHz internal clock reference and the wake-up period is set to 128ms.
4.2.4 Feedback Comparison

The function of the feedback comparison routine is to control the PWM duty cycle periodically. When it detects a crossover point between the feedback voltage and the reference voltage, the ONTime and OFFTime variables will be updated according to the result of the analog comparator output that is used as a feedback signal to indicate whether the LED current is higher or lower than the expected level.

These two variables should be adjusted in opposite polarity such the overall period is kept constant (i.e. OFFTime = 255 – ONTime). The PWM duty cycle is determined by the value of ONTime. The variables ONTime and OFFTime should be set in the range of 0 to 255.
4.2.5 PWM Generation

The 8-bit modulo timer is configured to generate the PWM signal. The timer is running in modulo mode such that the overflow flag will be set when the count value matches the modulo value in the MTIM module register, MTIMMOD. PWM output is implemented by programming two timer overflow variables, ONTime and OFFTime alternately into the MTIM module register.

The overall PWM period is equal to the sum of PWM ON, PWM OFF and the time used for software looping in KBI detection and feedback comparison routines.
Chapter 5
Testing

The reference design was tested with Lumileds high brightness LED in typical conditions. The LED forward current was regulated with expected performance with 5V supply input. Dimming control was achieved by changing the LED current with a single button user interface.

5.1 Measurement Data

Figure 5-1 shows the voltage waveform measured across the current sense resistor $R_S$ in 100% brightness setting. The LED forward current was determined by this voltage divided by one ($R_S = 1\Omega$), which was equal to around 385mA and the peak-to-peak ripple current was 26mA. The measurements show that the LED forward current was regulated and tracking to the reference input accurately. The difference between the calculated target (350mV) and measured voltage (385mV) showing in Figure 5-2 is due to the tolerance of resistors used.
The efficiency of the whole converter was measured as:

1. Supply input = 5V, supply current = 302mA, DC to DC output voltage = 3.5V
2. Voltage across $R_S$ = 385mV, LED forward voltage = 3.115V

\[
\text{Efficiency} = \frac{\text{Output Power}}{\text{Input Power}}
\]

(EQ 5-1)

\[
\text{Efficiency} = \frac{3.115 \times 0.385}{5 \times 0.302} = 79.4\% 
\]

(EQ 5-2)

3. Overall efficiency = 79.4%

5.2 Customization

5.2.1 Hardware

The on-board open source BDM is used for development purposes only. It can be removed to make the design more compact and fit into a smaller housing.

The inductor value can be adjusted to match with new LED forward current requirement.
Chapter 6
Demo Setup

6.1 Introduction

This section shows how to setup the MC9RS08KA2 HB-LED demo board and run the included demo program. Users can use the on-board open source BDM module to program the MCU FLASH and debug new applications via the USB connection.

6.2 Default Settings

Figure 6-1 shows the default settings for the demo board. The black blocks indicate the default position of the jumpers. Please check these settings before continuing.

![Diagram of default settings](image)

Figure 6-1. Demo Board Default Settings

6.3 Programming and Running the HB-LED Demo

Further application development and debug for the MC9RS08KA2 is supported through the on-board OS-BDM interface. A USB type B connector (S1) provides the connection between the demo board and your host PC. This tool is tested to run on Windows XP with Freescale CodeWarrior version 5.1. Use the procedure below to operate the HB-LED demo board.

The board must be powered by external 5V supply when HB-LED is turned on. Do not use the USB 5V for powering the HB-LED.

Program:

1. Move jumper JP3 to the 1-2 position “BDM” to power up device in background debug mode.
2. Programme the device with the corresponding software via the USB connection. Refer to documentation in the CD-ROM that came with the HB-LED demo board.
Demo Setup

**NOTE**

In-circuit debugging cannot be used for this application program since the BKGD pin is used for dimming control purpose. Use JP3 to switch between BDM and user mode.

Run:

1. Move jumper JP3 to the 2-3 position “USER” to isolate the “BDC_BKGD” pin.
2. Power up the device for software execution in user mode.

**NOTE**

All RS08 devices require a power-on-reset to switch between user mode and background debug mode.

### 6.4 Troubleshooting

The high brightness LED does not turn on:
- Make sure jumper JP3 is set to “USER” (2-3) position.
- Make sure supply input selection (JP1) is selected for external power supply.

Unable to program the MCU using the on-board OS-BDM:
- Make sure jumper JP3 is set to “BDM” (1-2) position.
/**
 * HB LED Coding for 9RS08KA2
 *
 * Author: Vincent Ko
 * Date: Apr 2006
 *
 * PTA0/KBI0/ACMP+  LED Driver
 * PTA1/KBI1/ACMP-  Voltage Reference
 * PTA2/KBI2/TCLK/RESETb/VPP  Dimming Button
 * PTA3/ACMP0/BKGD/MS  DIM0
 * PTA4/KBI4  PWM
 * PTA5/KBI5  DIM1
 *
 * include derivative specific macros
 * XDEF Entry
 * include "MC9RS08KA2.inc"
 */

;******************************************************************************
; ICS Definition
;******************************************************************************
ICS_DIV_1   equ $00
ICS_DIV_2   equ $40
ICS_DIV_4   equ $80
ICS_DIV_8   equ $c0

;******************************************************************************
; MTIM Definition
;******************************************************************************
MTIM_DIV_1  equ $00
MTIM_DIV_2  equ $01
MTIM_DIV_4  equ $02
MTIM_DIV_8  equ $03
MTIM_DIV_16 equ $04
MTIM_DIV_32 equ $05
MTIM_DIV_64 equ $06
MTIM_DIV_128 equ $07
MTIM_DIV_256 equ $08

MTIM_BUS_CLK equ $00
MTIM_XCLK    equ $10
Demo Setup

MTIM_TCLK_FALLING   equ  $20
MTIM_TCLK_RISING    equ  $30

;=========================================================================
; ACMP Definition
;=========================================================================
ACMP_OUTPUT_FALLING equ $00
ACMP_OUTPUT_RAISING equ $01
ACMP_OUTPUT_BOTH    equ  $03

;=========================================================================
; RTI Definition
;=========================================================================
RTI_DISABLE    equ  $00
RTI_8MS        equ  $01
RTI_32MS       equ  $02
RTI_64MS       equ  $03
RTI_128MS      equ  $04
RTI_256MS      equ  $05
RTI_512MS      equ  $06
RTI_1024MS     equ  $07

;=========================================================================
; Application Definition
;=========================================================================
VREF equ PTAD_PTAD0
mVREF equ mPTAD_PTAD0
LED equ PTAD_PTAD1
mLED equ mPTAD_PTAD1
BUTTON equ PTAD_PTAD2
mBUTTON equ mPTAD_PTAD2
DIM0 equ PTAD_PTAD3
mDIM0 equ mPTAD_PTAD3
PWM equ PTAD_PTAD4
mPWM equ mPTAD_PTAD4
DIM1 equ PTAD_PTAD5
mDIM1 equ mPTAD_PTAD5

;=========================================================================
; Application Macro
;=========================================================================
Delay_1_cycle: macro
  nop ;1 cycles
endm
Delay_2_cycles: macro
  tsta ;2 cycles
endm
Delay_3_cycles: macro
  brn ;3 cycles
endm
Delay_4_cycles: macro
  brn ;3 cycles
  nop ;1 cycles
endm
Delay_5_cycles: macro
   tstx ; 5 cycles
endm
Delay_10_cycles: macro
   tstx ; 5 cycles
tstx ; 5 cycles
endm
Delay_15_cycles: macro
   tstx ; 5 cycles
tstx ; 5 cycles
tstx ; 5 cycles
endm

org TINY_RAMStart
; variable/data section
ONTime ds.b 1
Pressed ds.b 1

org RAMStart
; variable/data section

org ROMStart
; code section
main:
Entry:
;-------------------------------------------------------
; Config ICS
; Device is pre-trim to 20MHz ICLK frequency
; TRIM value are stored in $3FFA:$3FFB
;-------------------------------------------------------
mov #HIGH_6_13(NV_ICSTRM), PAGESEL
; mov MAP_ADDR_6(NV_FTRIM), ICSSC ; $3FB
; mov MAP_ADDR_6(NV_ICSTRM), ICSTRM ; $3FA
mov #ICS_DIV_1, ICSC2 ; Use 10MHz
;-------------------------------------------------------
; Config System
;-------------------------------------------------------
mov #HIGH_6_13(SOPT), PAGESEL ; Init Page register
mov #(mSOPT_COPT|mSOPT_STOPE), MAP_ADDR_6(SOPT); BKGD disable,COP disabled
mov #(mSPMSC1_LVDE|mSPMSC1_LVDRE), MAP_ADDR_6(SPMSC1); LVI enable
mov #(RTI_128MS), MAP_ADDR_6(SRTISC); 128ms RTI
;-------------------------------------------------------
; Init RAM
;-------------------------------------------------------
clr ONTime
clr Pressed
;-------------------------------------------------------
; Config GPIO
; PWM - init H
Demo Setup

;-------------------------------------------------------
mov   #(mPWM), PTAD ; Initial port
mov   #(mDIM0|mPWM), PTADD ; DIM0 and PWM Output pins
;-------------------------------------------------------
;Config KBI
;-------------------------------------------------------
lda   #mBUTTON
sta    MAP_ADDR_6(PTAPE) ;Enable Pullup
sta    KBIE ;KBI Enable
bset   KBISC_KBACK, KBISC
;-------------------------------------------------------
;Config ACMP
;-------------------------------------------------------
mov   #(mACMPSC_ACME), ACMPSC ; Enable ACMP
;-------------------------------------------------------
;Config MTIM
;
;Timer prescalar=1 -> Timer clk = 10MHz
;Bus = 10MHz
;Max OF period = 25.6us
;Timer resolution = 100ns
;-------------------------------------------------------
mov   #(MTIM_BUS_CLK|MTIM_DIV_1), MTIMCLK
mov    #255, MTIMMOD
;-------------------------------------------------------
;PWM Control Loop
;-------------------------------------------------------
PWMControlLoop:
    bclr   PWM, PTAD ; Switch close
    ; --- Check Button ---
    brset  SRTISC_RTIIF, MAP_ADDR_6(SRTISC), CheckPress;5
    Delay_10_cycles
    Delay_10_cycles
    Delay_4_cycles
    bra    CheckACMP ;3
CheckPress:
    bset   SRTISC_RTIACK, MAP_ADDR_6(SRTISC);5
    brset  0, Pressed, ToggleDim ;5
CheckButton:
    brset  KBISC_KBF, KBISC, ButtonPressed;5
    Delay_5_cycles
    Delay_4_cycles
    bra    CheckACMP ;3
ButtonPressed:
    bset   KBISC_KBACK, KBISC ;5
    inc    Pressed ;4
    bra    CheckACMP ;3
ToggleDim:
bset KBISC_KBACK, KBISC ;5
lda PTADD ;3
eor #mDIM1 ;2
sta PTADD ;2
clr Pressed ;2
Delay_3_cycles

; --- Check ACMP ---
CheckACMP:
    lda ONTime ;3
    brset ACMPSC_ACO, ACMPSC, LowerThanRef;5
HigherThanRef:
    cbeqa #0, DummyCycles ;4
dec ;1
bra PWMGen ;3
LowerThanRef:
    cbeqa #255, DummyCycles ;4
inc ;1
bra PWMGen ;3
DummyCycles:
    nop ;1
bra PWMGen ;2
PWMGen:
    sta ONTime ;2
    bne EnableOnTime ;3
Delay_10_cycles
Delay_3_cycles
bra EnableOffTime ;3
EnableOnTime:
    sta MTIMMOD ;2 ON period
    mov #(mMTIMSC_TRST|mMTIMSC_TOIE), MTIMSC;4 Reset and Start Timer
    lda #255 ;2
    sub ONTime ;3
    wait ;2+
    beq ZeroOffTime ;3 Duty cycle is off
EnableOffTime:
    bset PWM, PTAD ;5 Switch open
    sta MTIMMOD ;2 OFF period
    mov #(mMTIMSC_TRST|mMTIMSC_TOIE), MTIMSC;4 Reset and Start Timer
    wait ;2+
    jmp PWMControlLoop ;3
ZeroOffTime:
    Delay_10_cycles
    Delay_3_cycles
    jmp PWMControlLoop ;3

; Reset Vector
org $3ffe
Security:
dc.b $FF
jmp main
RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale’s Environmental Products program, go to http://www.freescale.com/epp.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2006. All rights reserved.