RZ/A2M Group
DRP Library User's Manual and Functional Design Specifications
First Edition (Rev. 1.00)

Bayer2LrgbColorCorrection

September 24, 2019		
Renesas Electronics		
Approved by	Examined by	Author

User's Manual

RZ/A2M Group

DRP Single Library User's Manual

Bayer2LrgbColorCorrection

Renesas Electronics
www.renesas.com

Rev.1.00 September. 2019

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Purpose and Target Readers

This manual is intended to provide the user with an understanding of the functions of the DRP library and how to utilize them. It is aimed at users designing application systems making use of the DRP library. In order to use this manual, you will need a basic knowledge of programming languages and microprocessors.

> Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section.

> The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

Contents

1. Introduction

1.1 Summary

This manual describes the functions and usage of the DRP library, which run on the dynamically reconfigurable processor (DRP) of RZ/A2M Group Microprocessors.

The DRP can perform various functions according to user's setting. In this document, the function performed by DRP is called "circuit", and the data representing circuit information is called "configuration data". Writing of the circuit to DRP can be performed by loading the configuration data using DRP Driver[*1]. DRP Library is a collection of configuration data with various functions, mainly image processing.

Note 1. For details of DRP Driver, refer to "RZ/A2M Group DRP Driver User's Manual (R01US0355)".

1.2 Functions

The functions of the configuration data contained in the DRP library are listed below.

Table 1.1 DRP Library Functions

Category	Function Name	Outline	Page
Image conversion	Bayer2LrgbColorCorrection	Converts from RAW data acquired from CMOS camera to LRGB (With color correction), L contains grey scale picture	11

2. Operation Conditions

The DRP library operates under the conditions listed below.

Table 2.1 Operation Conditions

Item	Description
Microprocessor	RZ/A2M Group Microprocessors*1 • R7S921051VCBG • R7S921052VCBG • R7S921053VCBG

Note 1. The DRP library operates on RZ/A2M Group Microprocessors equipped with a DRP function module. It will not operate on RZ/A2M Group Microprocessors without a DRP function module.

This library was confirmed to operate in the following development environment:

Renesas e^2 studio 7.5.0

The following toolchain is compatible:

GCC ARM Embedded Toolchain 6-2017-q2-update

3. File Structure

Figure 3.1 ,shows the file structure of configuration data and header files in the DRP library.

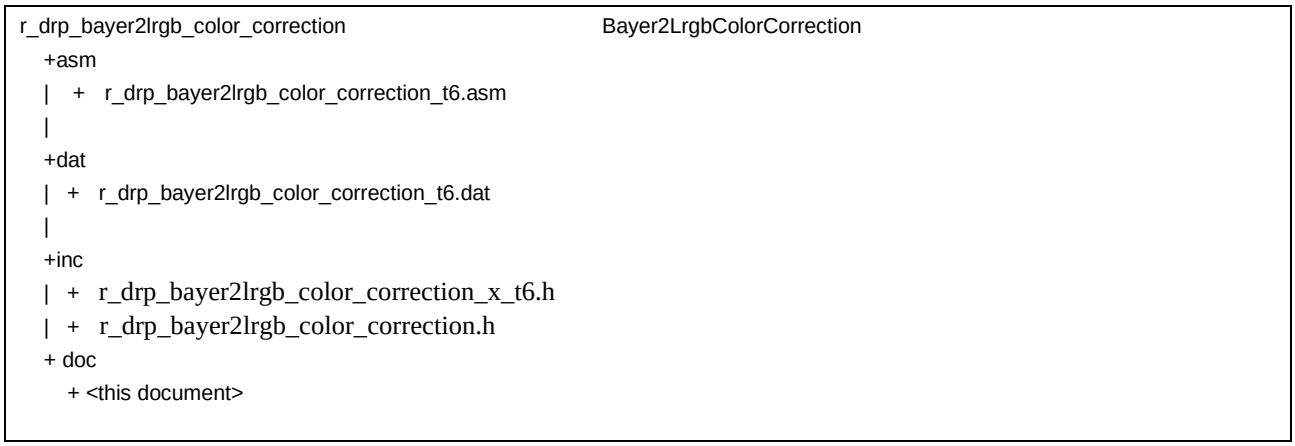


Figure 3.1 File Structure

4. DRP Library Reference

4.1 How to Read the DRP Library Reference

In this section the specifications of the configuration data contained in the DRP library are presented in the format shown below.

Function name*1

Function outline

Configuration data file	The name of the configuration data file. Use the DRP Driver's R_DK2_Load() function to load the data in the DRP.
Supported version	Lists the version of the configuration data that operates under present specification. Use the DRP Driver's R_DK2_GetInfo() function to get the version.
Configuration data size (byte)	Lists the size of the configuration data. Lists all versions, if there are different versions.
Header file	The name of the header file for using the configuration data. Use #include "header file" to include the file.
Parameter	Lists the parameters required by the circuit. Parameters are passed from the CPU to the DRP by means of the DRP driver's R_DK2_Start() function. Parameters are defined as a structure within the header file. Before running the circuit, set the parameters on the CPU side. The data type defined in stdint.h is used. Also, the area where parameters are stored and the area indicated by parameters representing addresses such as 'src' and 'dst' must be located in physical memory. *2
I/O details	Lists the details of the data specified by the parameters. Unless otherwise indicated, the same address may be specified for the input buffer address and output buffer address.
Number of tiles	The number of tiles used by the circuit. The DRP has 6 tiles. The DRP Driver's R_DK2_Load() function is used to assign circuits to tiles.
Segmented processing	Indicates that the function can be processed in parallel by multiple circuits. In parallel processing, the input image is divided up in the vertical direction and processed accordingly. The segmented processing can be executed by utilizing the 6 tiles of DRP and loading multiple configuration data of 3 tiles or less. For details on loading multiple configuration data of 3 tiles or less into DRP, see the explanation of R_DK2_Load () function in "RZ/A2M Group DRP Driver User's Manual".

Example: A case where the input image is divided into three portions in the vertical direction

Description	Describes the specifications of the configuration data.
Note	Additional notes appear here.

Note 1. The function name of configuration data is a character string that can be obtained from the configuration data by using the DRP Driver's R_DK2_GetInfo() function.

Note 2. If the values of physical memory in the area of parameters and input/output data of the circuit are incorrect because the values are in the Cortex-A9 cache, etc., the circuit does not work properly. It must be necessary to clean the cache before calling the DRP driver's R_DK2_Start() function or to allocate the parameters and input/output data of circuit to a non-cached area.

For information on using the API functions of the DRP Driver, refer to "RZ/A2M Group DRP Driver User's Manual (R01US0355)".

4.2 Image Conversion

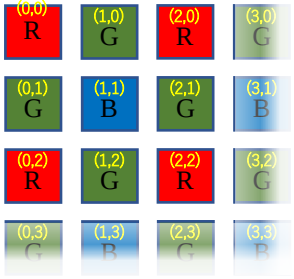
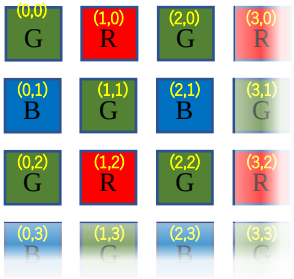
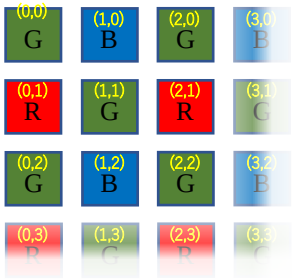
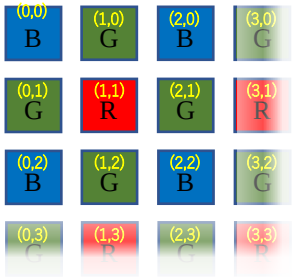
4.2.1 Bayer2LrgbColorCorrection

Bayer2 LrgbColorCorrection

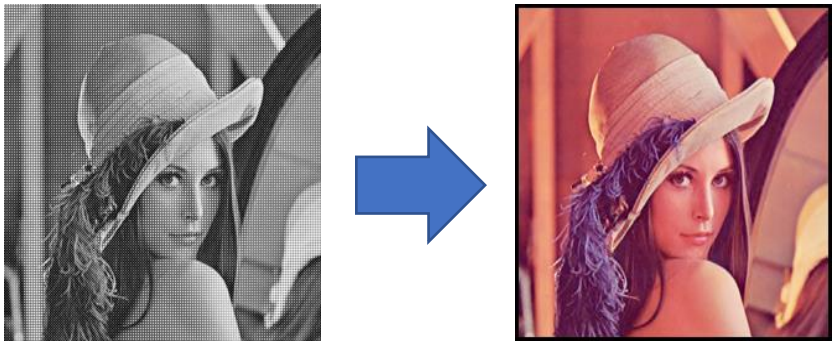
Converts from RAW data acquired from CMOS camera to LRGB

RGB(with color correction), L contains grey scale picture (without color correction)

Configuration data file			r_drp_bayer2lrgb_color_correction_t6.dat
Supported version			0.90
Configuration data size (byte)			188384
Header file			r_drp_bayer2lrgb_color_correction_x_t6.h (r_drp_bayer2lrgb_color_correction.h)
Parameter	Structure name		
	r_drp_bayer2lrgb_color_correction_t		
	Member name	Type	Description
	src	uint32_t	Input image address
	dst	uint32_t	Output image address
	dst_layersum	uint32_t	Sum of L values Output to struct address (r_drp_bayer2lrgb_color_correction_layer_t)
	Width	uint16_t	Image width (pixels)
	Height	uint16_t	Image height (pixels)
	gain_r	uint16_t	Gain correction value of image (R component). The upper 4 bits are an integer part, the lower 12 bits are a decimal part.
	gain_g	uint16_t	Gain correction value of image (G component). The upper 4 bits are an integer part, the lower 12 bits are a decimal part.
	gain_b	uint16_t	Gain correction value of image (B component). The upper 4 bits are an integer part, the lower 12 bits are a decimal part.
	Pattern	uint8_t	Specify the bayer pattern of input image 0: RGGB 1: GRBG 2: GBRG 3: BGGR
	r_drp_bayer2lrgb_color_correction_layer_t		
	Member name	Type	Description
	layersum	uint32_t	Output: sum of all active l channel pixel values
	layeractivepixel	uint32_t	Output: number of active l channel pixel
			average = layersum / layeractivepixel

I/O details	Input image	Address: Width (pixels): Height (pixels): Data size:	Specified by src. Specified by width. (16 to 1280) Specified by height. (4 to 960) (width) × (height) × 1 byte
		Format The input image formats are 4 patterns shown below. RGGB:  (X coordinate, Y coordinate) = (even, even): red (even, odd): green (odd, even): green (odd, odd): blue GRBG:  (X coordinate, Y coordinate) = (even, even): green (even, odd): blue (odd, even): red (odd, odd): green GBRG:  (X coordinate, Y coordinate) = (even, even): green (even, odd): red (odd, even): blue (odd, odd): green BGGR:  (X coordinate, Y coordinate) = (even, even): blue (even, odd): green (odd, even): green (odd, odd): red	
	Output image	Address: Width (pixels): Height (pixels): Format: Data size: Video buffer:	Specified by dst. Same as input image Same as input image LRGB (4 bytes per pixel) (width) × (height) × 4 bytes LRGB can be written to RGB888 (32bit) video buffer directly
Number of tiles	6		

Segmented processing	Not supported
Description	This function converts the image at the address specified by src from Bayer format to RGB format using Advanced Color Plane Interpolation (ACPI) and outputs the result to the address specified by dst. The ACPI is a method to obtain sharp color images by adding high frequency components to the linear interpolation value of surrounding pixels to be interpolated. This method calculates interpolation values from two directions, vertical and horizontal, then it adopts interpolation values in the direction is more continuous at the original pixel to be processed. Also, it calculates the missing component pixel using the information of other component. **This function outputs black pixels at the top, bottom, left, and right 3 pixels of the output image as shown below because it does not execute border processing at the image edge.**



Input Image
(Bayer format)

RGB output Image part of
LRGB

This function correct respective component pixel values of RGB converted from Bayer by setting correction value to the parameter “gain_*”. But, Set the value of “Actual value multiplied by 4096” to “gain_*” because it is fixed-point (the upper 4 bits are an integer part, the lower 12 bits are a decimal part).

The L channel data represents a grey picture and will be calculated by the following equations:

Lp = (R(no gain) × 76
+ G(no gain) × 150
+ B(no gain) × 37) ÷ 256
L = min(Lp,255)

The average brightness of the L channel can be calculated on the following way:
averageL = layersum / layeractivepixel

L output Image part of
LRGB

.

Note	None

5. Using the DRP Library

To use this library, it is necessary to initialize the DRP, load configuration data, etc. Also, since the parameters are different for each configuration data, set the parameters based on the specification of the configuration data to be used. For application example of DRP library, refer to "RZ/A2M Group 2D Barcode Application Note (R01AN4503)".

Usage example:

```
#include "r_drp_bayer2lrgb_color_correction_x_t6.h"
```

….

// variable declaration

```
uint16_t brightness = 120;

uint8_t  *input_bufadr;
uint8_t  *output_bufadr;

int32_t ret_val;

static uint8_t drp_lib_id[R_DK2_TILE_NUM] = {0};
static volatile uint8_t drp_lib_status[R_DK2_TILE_NUM] = {DRP_NOT_FINISH};

static r_drp_bayer2lrgb_color_correction_t param_bayer2lrgb_color_correction
       __attribute__ ((section("Uncache_IRAM")));

static r_drp_bayer2lrgb_color_correction_layer_t bayer2lrgb_layersum
      __attribute__ ((section("Uncache_IRAM")));


static void cb_drp_finish(uint8_t id)
{
    uint32_t tile_no;
    /* Change the operation state of the DRP library notified by the argument to finish
*/
    for (tile_no = 0; tile_no < R_DK2_TILE_NUM; tile_no++)
    {
        if (drp_lib_id[tile_no] == id)
        {
            drp_lib_status[tile_no] = DRP_FINISH;
            break;
        }
    }
    return;
}

/***********************************************************************
* End of function cb_drp_finish
***********************************************************************/
```

…

// usage example

```
/*****************************************/
/* Load DRP Library                      */
/*        +----------------------------+ */
/* tile 0 |                            | */
/*        +                            + */
/* tile 1 |                            | */
/*        +                            + */
/* tile 2 |                            | */
/*        + Bayer2lrgbColorCorrection  + */
/* tile 3 |      (WhiteBalance)        | */
/*        +                            + */
/* tile 4 |                            | */
/*        +                            + */
/* tile 5 |                            | */
/*        +----------------------------+ */
/*****************************************/

ret_val = R_DK2_Load(&g_drp_lib_bayer2lrgb_color_correction_t6[0],
                     R_DK2_TILE_0,
                     R_DK2_TILE_PATTERN_6, NULL, &cb_drp_finish, &drp_lib_id[0]);
DRP_DRV_ASSERT(ret_val);

/************************/
/* Activate DRP Library */
/************************/
ret_val = R_DK2_Activate(drp_lib_id[TILE_0], 0);
DRP_DRV_ASSERT(ret_val);

/*************************************/
/* Set R_DK2_Start function parameters */
/*************************************/

/* Set Bayer2LrgbColorCorrection parameters */
input_bufadr  = R_BCD_CameraGetFrameAddress(frame_buf_id);
output_bufadr = R_BCD_LcdGetVramAddress();  /* address video buffer which is not currently displayed */

/*************************************
Set R_DK2_Start function parameters
*************************************/

/* Set the address of buffer to be read/write by DRP */
R_MMU_VAtoPA((uint32_t)input_bufadr,  &(param_bayer2lrgb_color_correction.src));
R_MMU_VAtoPA((uint32_t)output_bufadr, &(param_bayer2lrgb_color_correction.dst));

/* Set Image size */
param_bayer2lrgb_color_correction.width  = R_BCD_CAMERA_WIDTH;
param_bayer2lrgb_color_correction.height = R_BCD_CAMERA_HEIGHT;

/* Set magnification */
param_bayer2lrgb_color_correction.gain_r = 0x148F;
param_bayer2lrgb_color_correction.gain_g = 0x1000;
param_bayer2lrgb_color_correction.gain_b = 0x1A72

param_bayer2lrgb_color_correction.pattern = 0x00; // RGGB pattern

param_bayer2lrgb_color_correction.dst_layersum = (uint32_t)&bayer2lrgb_layersum;
```

```
R_MMU_VAtoPA((uint32_t)param_bayer2lrgb_color_correction.dst_layersum,
                        &(param_bayer2lrgb_color_correction.dst_layersum));

/* Initialize variables to be used in termination judgment of the DRP library */
drp_lib_status[TILE_0] = DRP_NOT_FINISH;

/*********************/
* Start DRP Library */
/*********************/
ret_val = R_DK2_Start(drp_lib_id[TILE_0], (void *)& param_bayer2lrgb_color_correction,
                                                  sizeof(r_drp_param_bayer2lrgb_color_correction_t));
DRP_DRV_ASSERT(ret_val);

/*************************************/
/* Wait until DRP processing is finish */
/*************************************/
while (drp_lib_status[TILE_0] == DRP_NOT_FINISH);


…..

// clean up


/***********************/
/* Unload DRP Library */
/***********************/
ret_val = R_DK2_Unload(drp_lib_id[TILE_0], &drp_lib_id[TILE_0]);
DRP_DRV_ASSERT(ret_val);


/**********************/
/* Set AE             */
/**********************/
if (bayer2lrgb_layersum.layeractivepixel != 0) {
      brightness = (uint16_t)(bayer2lrgb_layersum.layersum /
                              bayer2lrgb_layersum.layeractivepixel);
      R_BCD_AeRunAutoExpousure(&ae_setting, brightness);
}
```

6. Reference Documents

User's Manual: Hardware

RZ/A2M Group User's Manual: Hardware (R01UH0746)

(Download the latest version of the update or news from the Renesas Electronics website.)

User's Manual: Software

RZ/A2M Group DRP Driver User's Manual (R01US0355)

(Download the latest version of the update or news from the Renesas Electronics website.)

RZ/A2M Group 2D Barcode Sample Program Application Note (R01AN4503)

(Download the latest version of the update or news from the Renesas Electronics website.)

User's Manual: Development environment

For the Renesas Electronics integrated development environment (e^2 studio), visit the Renesas Electronics website to download the latest version.

Technical Update/Technical News

(Download the latest version of the update or news from the Renesas Electronics website.)

Revision History	RZ/A2M Group DRP Library User's Manual

Rev.	Date	Description	
		Page	Summary
1.00	Sep 20, 2019	—	First Edition issued

RZ/A2M Group DRP Custom Library User’s Manual

Publication Date: Rev.1.00 Sept. 20, 2019

Published by: Renesas Electronics Corporation

RZ/A2M Group

Renesas Electronics Corporation