RZ/A2M Group

Object Detection Sample Program Application Note

Introduction

This document describes the contents of the object detection sample program.

Target Device

RZ/A2M

Contents

1. Overview 2

2. Operation Confirmation Conditions 3

3. Folder Structure 5

4. Sample Program..6

4.1 Input/Output Specification 7

4.1.1 Input Specification 7

4.1.2 Output Specification 7

4.2 Processing details 8

4.2.1 Simple ISP... 8

4.2.2 Auto Exposure Correction (AE) 8

4.2.3 Image Shrinking.. 11

4.2.4 Noise Reduction 12

4.2.5 Edge Detection 13

4.2.6 Edge Dilation 14

4.2.7 Contour Detection 15

4.2.8 Color Image Output 16

4.3 Result Output.. 17

5. DRP Library..18

6. Reference Documents 19

Revision History 20

1. Overview

This application note explains the sample program which detects an object using the DRP (Dynamically Reconfigurable Processor) and the CPU.

This sample program can perform object detection by acquiring contour information from the image taken by a camera. By using the DRP in the image preprocessing stage to acquire contour information, even faster processing can be achieved compared to when processing is handled by the CPU. In addition, the result of the object detection process can be displayed on a monitor. The features of this sample program are shown below.

1. Robust object detection using the ISP* which avoids influence from surrounding brightness conditions.
2. Realtime object detection using the DRP (Dynamically Reconfigurable Processor) mounted on the RZ/A2M.
3. Output object image conforms on the monitor display.

Note: This is the Simple ISP function of DRP Library. For details, refer to 4.2.1 Simple ISP.

The following figure1.1 outlines the process of this sample program.

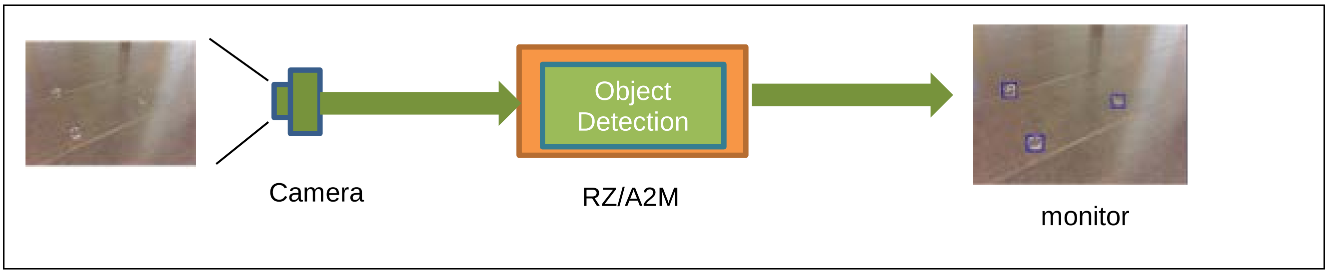


figure1.1 System overview of object detection sample program

2. Operation Confirmation Conditions

Figure 2.1 shows the environment for checking the operation of this sample program. Refer to the readme for DIP SW and jumper settings. For the display contents of the HDMI monitor, refer to Section 4.1.2 output specification.

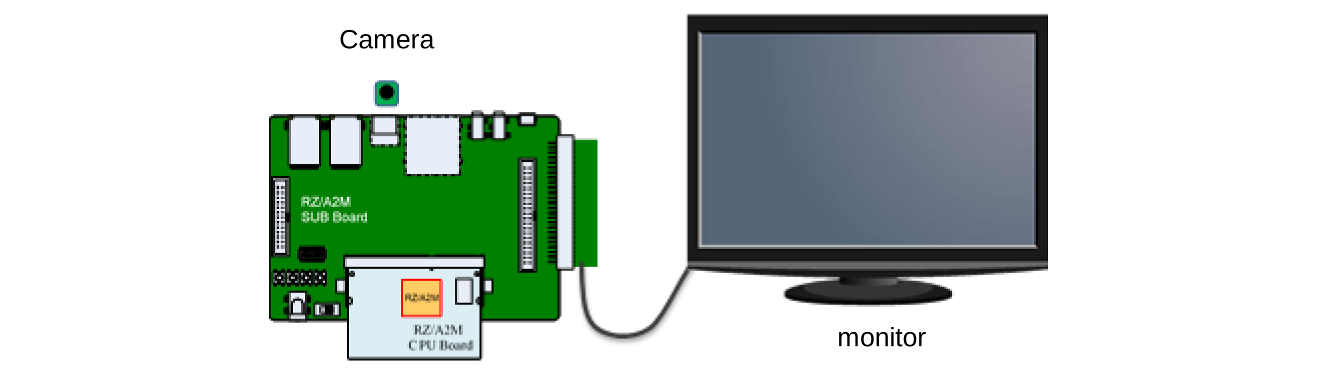


Figure 2.1 Operation check conditions

For this sample program, the cameras supplied with the RZ / A2M Evaluation Board Kit can be used.

The sample code of this application has been verified using the following conditions.

Table2.1 Operation Confirmation Conditions

Item	Contents
Microcomputer	RZ/A2M
Operating frequency[*1] [MHz]	CPU Clock (Iφ): 528MHz Image Processing Clock (Gφ): 264MHz Internal Bus Clock (Bφ): 132MHz Peripheral Clock 1 (P1φ): 66MHz Peripheral Clock 0 (P0φ): 33MHz QSPI0_SPCLK: 66MHz CKIO: 132MHz
Operating voltage	Power supply voltage (I/O): 3.3 V Power supply voltage (either 1.8V or 3.3V I/O (PVcc SPI)): 3.3V Power supply voltage (internal): 1.2 V
Integrated development environment	e2 studio v7.4.0
C compiler	GNU Arm Embedded Toolchain 6-2017-q2-update
	Compiler options (except directory path) Release: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0
Operation mode	Boot mode 3 (Serial Flash boot 3.3V)
Board	RZ/A2M CPU board RTK7921053C00000BE RZ/A2M SUB board RTK79210XXB00000BE HDMI Stick Board RTK79210XXB00010BE
Camera	Camera for Raspberry Pi using IMX 219 CMOS sensor
Monitor	VGA (640 x 480) HDMI monitor compatible with resolution
Device (functionality to be used on the board)	Serial flash memory allocated to SPI multi-I/O bus space (channel 0) Manufacturer: Macronix Inc. Model Name: MX25L51245GXD HyperRAM™[*2] (Connected to HyperRAM ™ space) Manufacturer: Cypress Inc. Model Name: S27KS0641DPBHI020

Note: 1. The operating frequency used in clock mode 1 (Clock input of 24MHz from EXTAL pin).
2. HyperRAM ™ is a registered trademark of Cypress Semiconductor Corporation.

3. Folder Structure

The folder structure of this sample program and an outline of their contents are shown as follows.

```
+- rza2m_drp_dynamic_sample3_freertos_gcc folder
||- demos folder                   project folder
| +- renesas folder
|  +- rza2m-ebk folder             RZ/A2M Evaluation Kitfolder
|   |- common folder               Common program folder
|    |- application_code folder    main folder
|    +- config_files folder        Configuration file storage folder
|   +- e2studio folder
|    |- .settings                  various settings
|    |- doc                        document
|    |- denerate                   Folders generated by Smart Configurator
|    |- src                        Code used by the application
|- lib folder                      Software provided by Amazon
| |- freeRTOS folder               FreeRTOS
| |- include folder                Software not provided by Amazon
|  +- third_party folder
|   +- mcu_vendor folder
|    +- renesas_rz folder
|     +- amazon_freertos_common folder
+- LICENSE folder
```

Figure 3.1 Folder Structure

The following open source software is bundled with this sample program.

Table 3.1 List of open source software packaged

Item	Description
FreeRTOS	It is open source software distributed under the MIT license. Regarding the MIT license, Please refer to https://opensource.org/licenses/mit-license.php. FreeRTOS is a real-time operation system kernel for embedded microcomputers. In this sample program, Kernel v10.0.0 is used. For the location of the FreeRTOS source code, refer to Figure 3.1.

4. Sample Program

This chapter describes the input/output specifications, the details of the contour detection process used in the sample program, and the resulting output.

The sample program consists of three parts: Simple ISP, contour detection, and result output. The following processing is performed on the image taken with the camera. For details of each process, refer to Chapters 4.2 and 4.3.

1. Simple ISP: Improves image quality, leading to improved object detection accuracy and prevention of false detection of objects due to noise.
2. Contour detection: Detects an object by detecting the contours in an input image.
3. Result output: The detection result is output to a monitor.

Figgure. 4.1 below is the system block diagram for the object detection sample program, and Fig. 4.2 shows the object detection sample program flowchart.

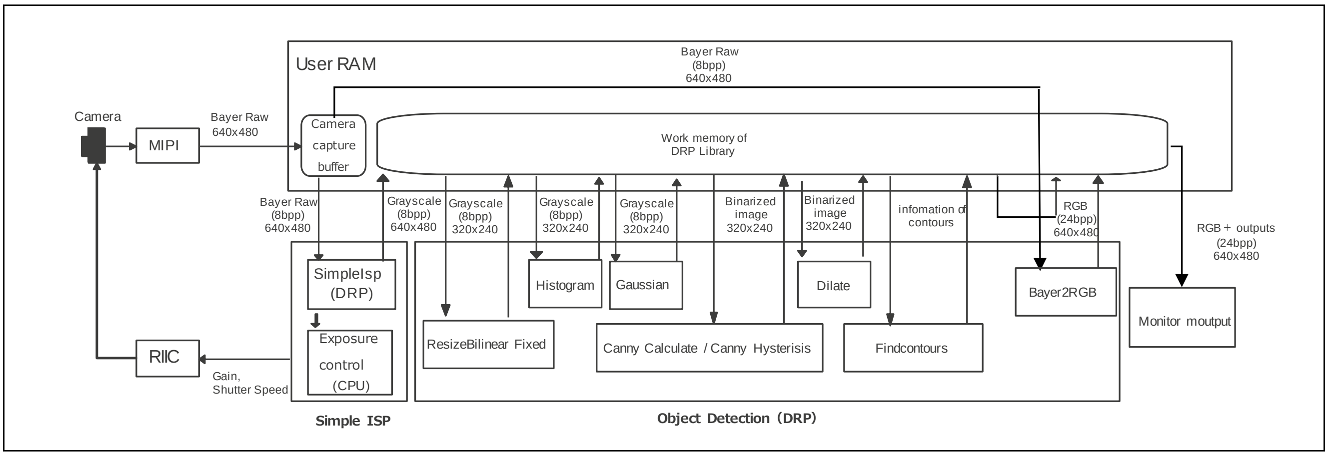


figure 4.1 System block of object detection sample program

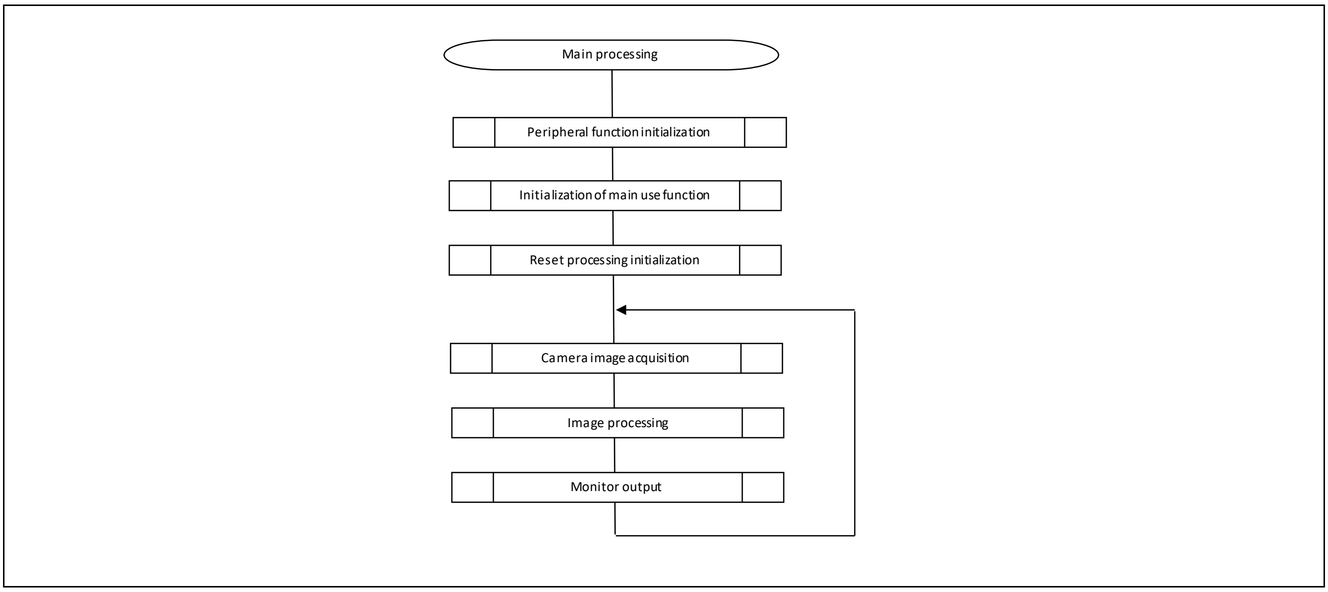


Figure 4.2 Flow chart of overall processing

4.1 Input/Output Specification

4.1.1 Input Specification

Table 4.1 shows the camera settings for this sample program.

Table 4.1 Camera setting

Input image format	Bayer format 8[bit per pixel]
Image capture size	640×480
Capture frame rate	Capture frame rate will fluctuate, by ON/OFF of 4.2.2Auto Exposure Correction (AE). ON：12～24fps*1 OFF：20fps*2

Note 1. The set value is 24 fps. However, it fluctuates between 12 and 24 fps due to changes in shutter speed.

2. The set value is 24 fps. However, because the shutter speed is 1/20 second, it is 20 fps.

4.1.2 Output Specification

This sample program allows the user to view the results of the object detection process on the monitor. Table 4.2 shows the monitor output specifications used in this sample program.

Table 4.2 HDMI monitor output specification

Image resolution	640×480
Image display frame rate	60 fps

4.2 Processing details

4.2.1 Simple ISP

Simple ISP controls camera exposure and suppresses noise for improved image quality and enhanced object detection accuracy.

The image data is converted to grayscale and output, because color information is not necessary for object detection.

This function uses Simple ISP of DRP Library. Please refer to "DRP Library user's manual (R01US0367)" for details of the Simple ISP.

The explanation for this section is continued in the sample program source code "r_bcd_main.c", onwards from the comment "/* Function : Simple ISP(AE, Bayer to grayscale conversion, Noise reduction) */".

4.2.2 Auto Exposure Correction (AE)

This function is not included in the current sample program.

Auto Exposure correction (AE) controls the exposure of the camera referring to the brightness output from Simple ISP.

The CPU handles calculations of the correction value and camera control. A flowchart describing the camera control process is shown in Figure 4.5, Figure 4.6 and Figure 4.7.

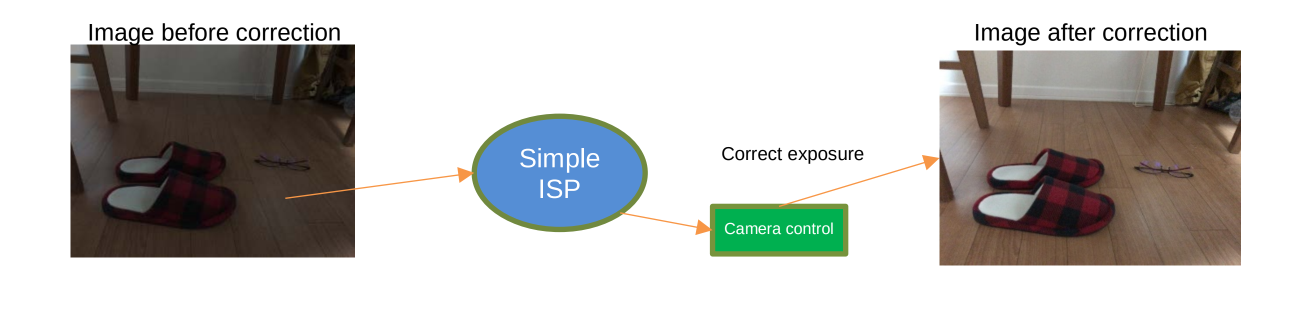
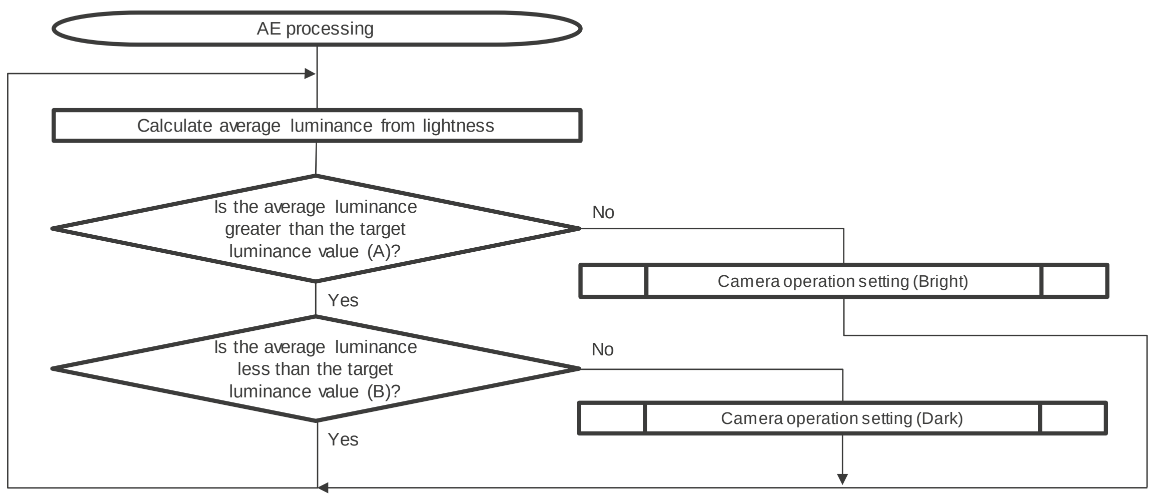
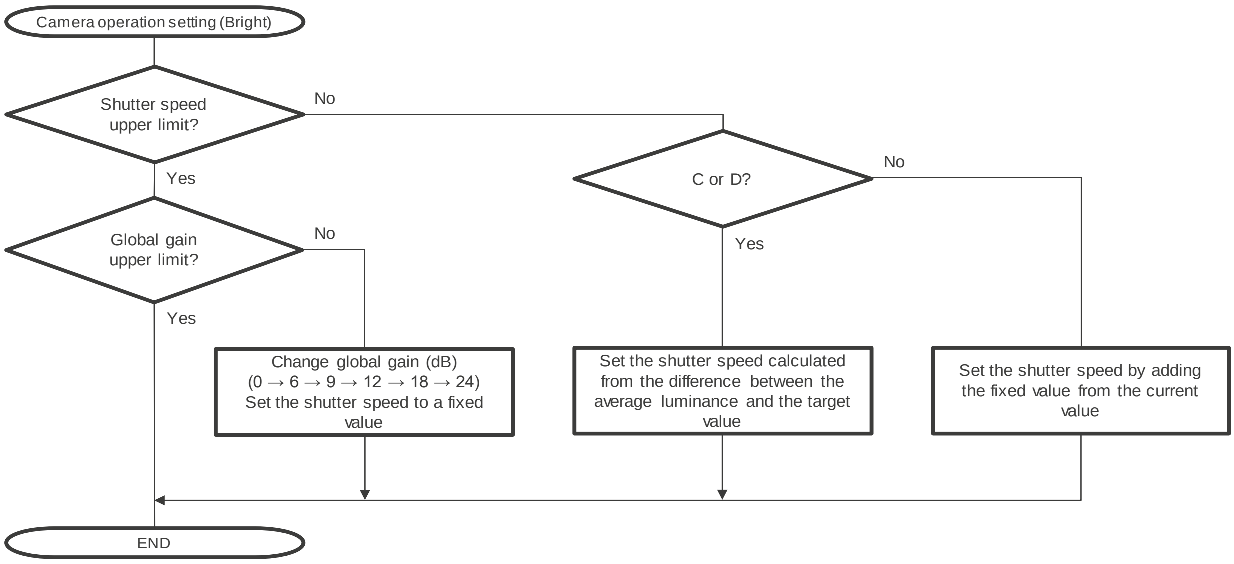


Figure 4.3 Lightness integrating process



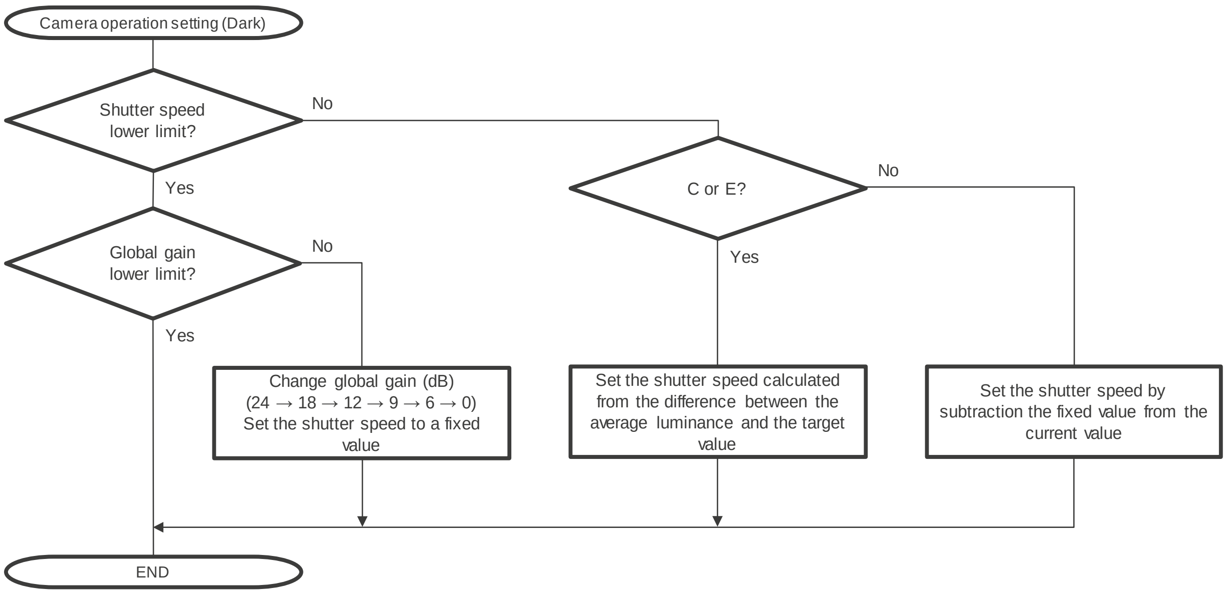
【注】 The target brightness value is ±10% of the brightness value (100) suitable for object detection with camera. (A = 90, B = 110)

Figure 4.4 AE processing flow



【注】 C. AE changed from OFF to ON.
D. The average luminance value was much lower than the target luminance value (A).

Figure 4.5 Camera operation setting (Bright) flowchart



【注】 C. AE changed from OFF to ON.
E. The average luminance value greatly exceeded the target luminance value (B).

Figure 4.6 Camera operation setting (Dark) flowchart

Figure 4.7 shows the flow of object detection in this sample program, which makes use of the DRP for recognition processing.

The details of each process of contour detection process are shown below.

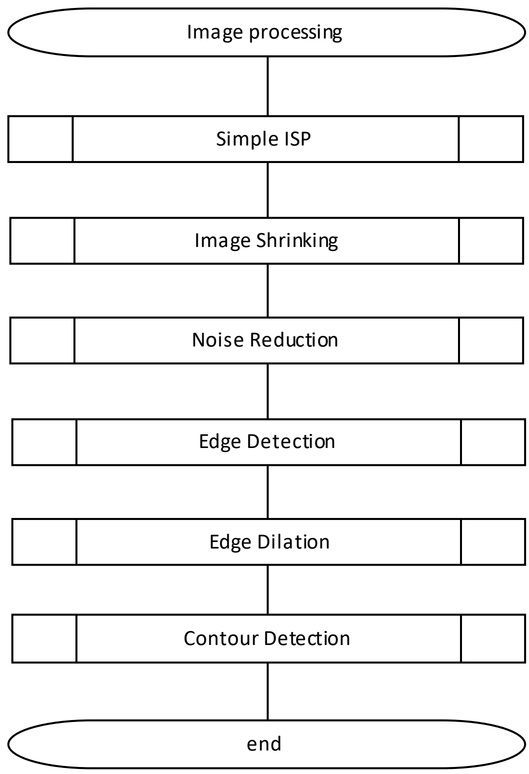


Figure 4.7 Flow chart of object detection processing

4.2.3 Image Shrinking

Converts the Bayer format image obtained from the camera into a grayscale image using the Simple ISP. In addition, the generated grayscale image (640 × 480) is reduced and converted to a smaller image (320 × 240) by the Resize application process.

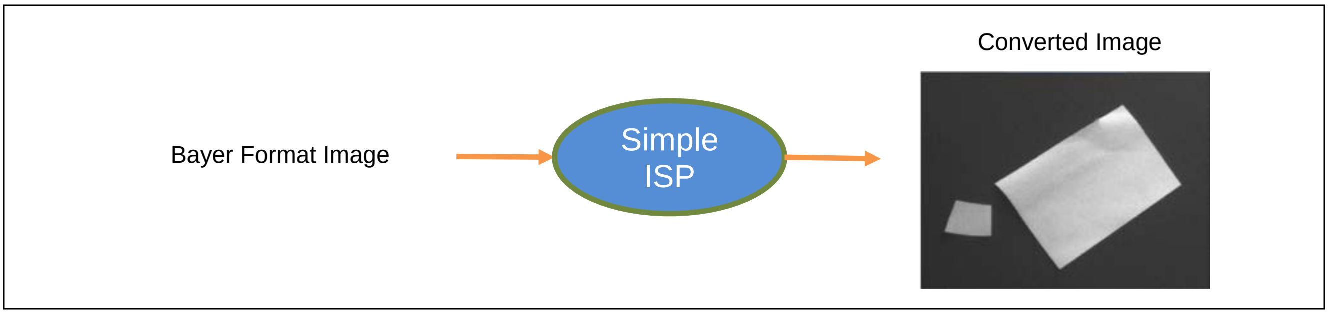


Figure 4.8 Simple ISP Process

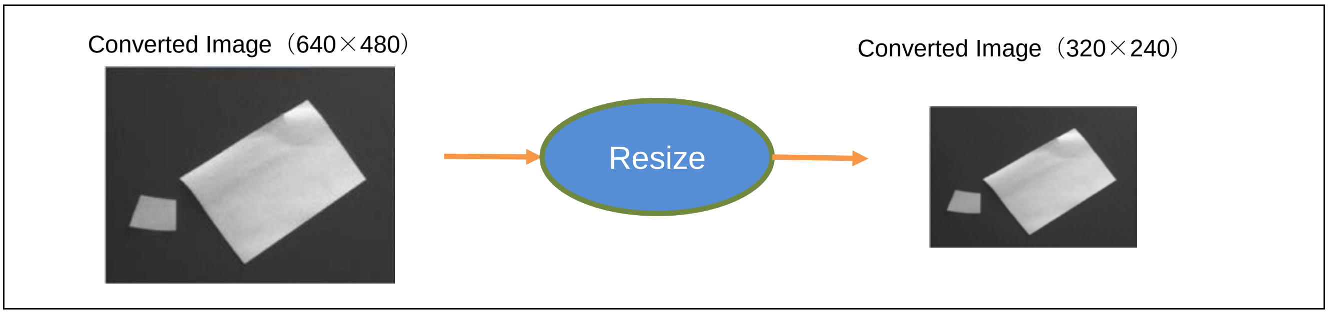


Figure 4.9 Image Resize Process

4.2.4 Noise Reduction

A Gaussian filter is used to remove noise from the reduced grayscale image.

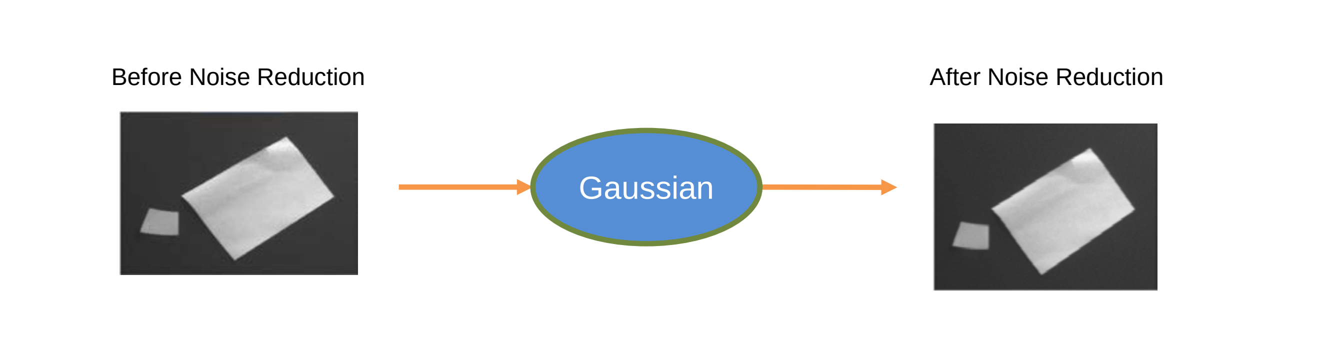


Figure 4.10 Noise Reduction Process (Gaussian)

4.2.5 Edge Detection

Performs edge detection processing using a Canny filter (Canny Calculate/Canny Hysteresis) on the grayscale image that has been subjected to noise removal processing.

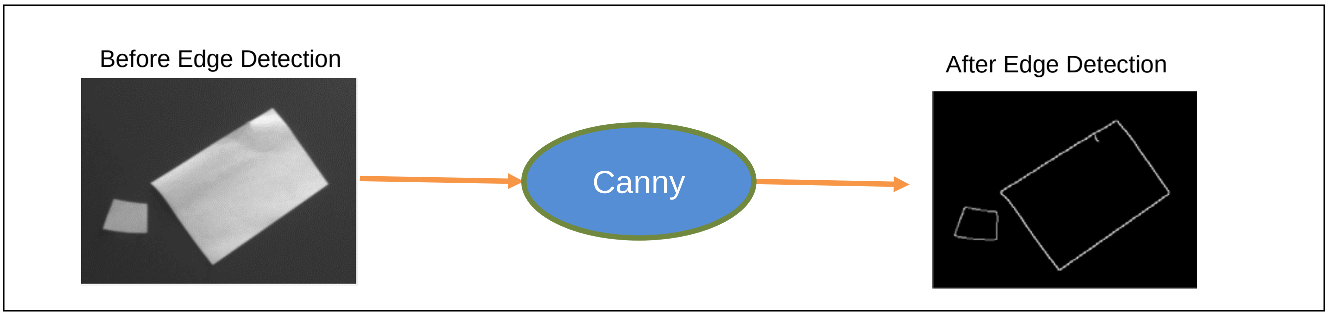


Figure 4.11 Edge Detection Process（Canny）

4.2.6 Edge Dilation

Performs edge dilation using the Dilate application process on the binarized image. By performing this process, it is possible to complement the outlines that have been cut.

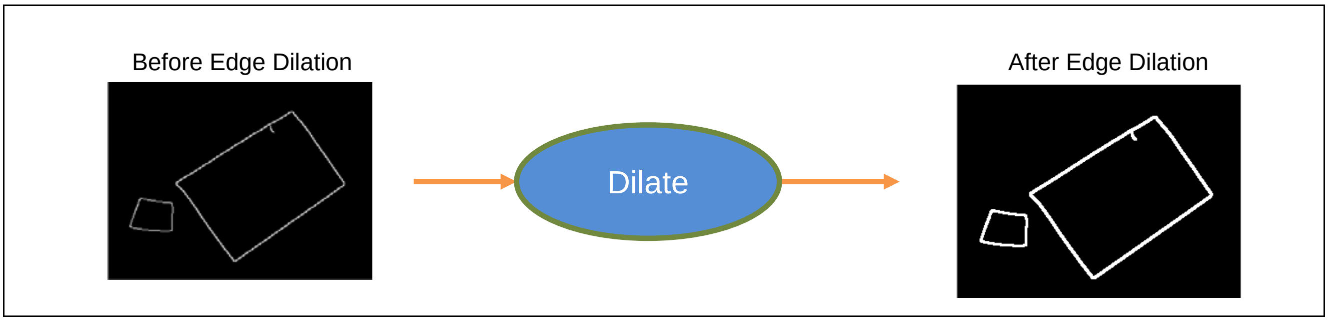


Figure 4.12 Edge Dilation Process（Dilate）

4.2.7 Contour Detection

Performs contour detection processing using the Findcontours application process on the image which was subjected to edge dilation. This process will output the circumscribed rectangle which is drawn around the detected contours. In this sample program, only the size of a certain range of rectangles is acquired. Refer to the "DRP Library User's Manual (R01US0367)" for details on the Findcontours process.

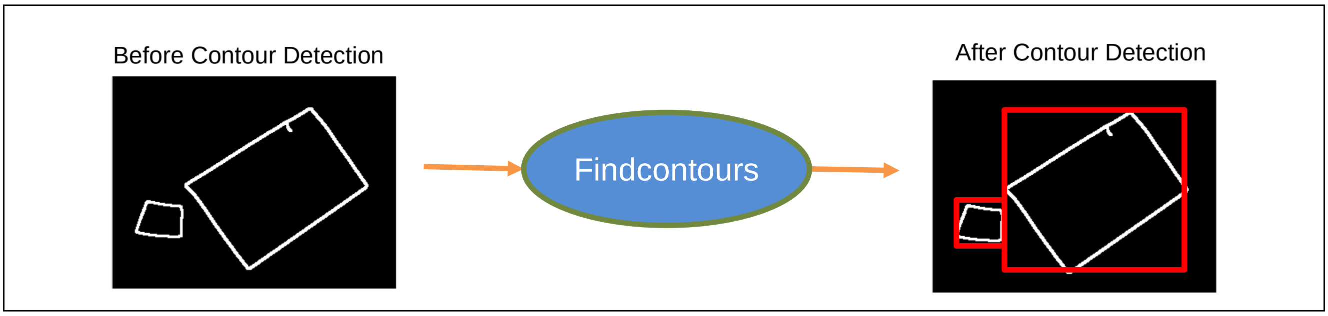


Figure 4.13 Contour Detection Processing（Findcontours）

4.2.8 Color Image Output

Uses the Bayer2RgbColorCorrection application process to generate a color image from the Bayer format image obtained from the camera. In this sample program, this color image is used to check the results on the monitor only (and is not used for object detection processing).

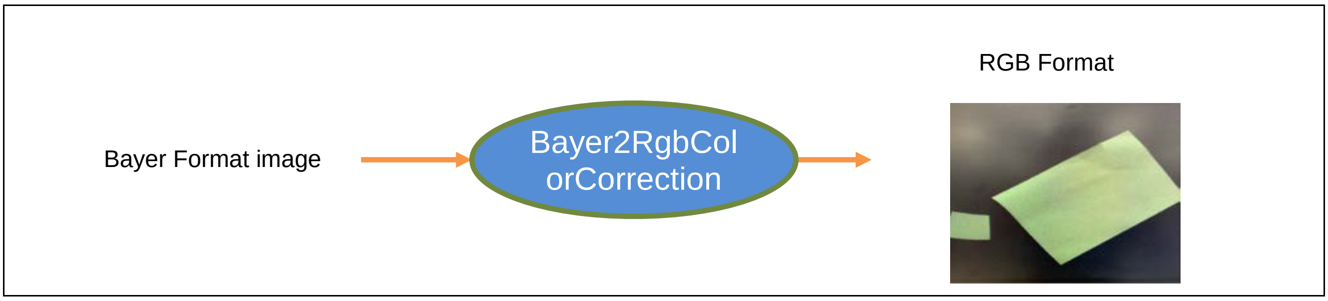


Figure 4.14 Color Image Output (Bayer2RgbColorCorrection)

4.3 Result Output

Figure 4.15 shows the monitor display contents. Table 4.3 lists the processes which are output to the display.

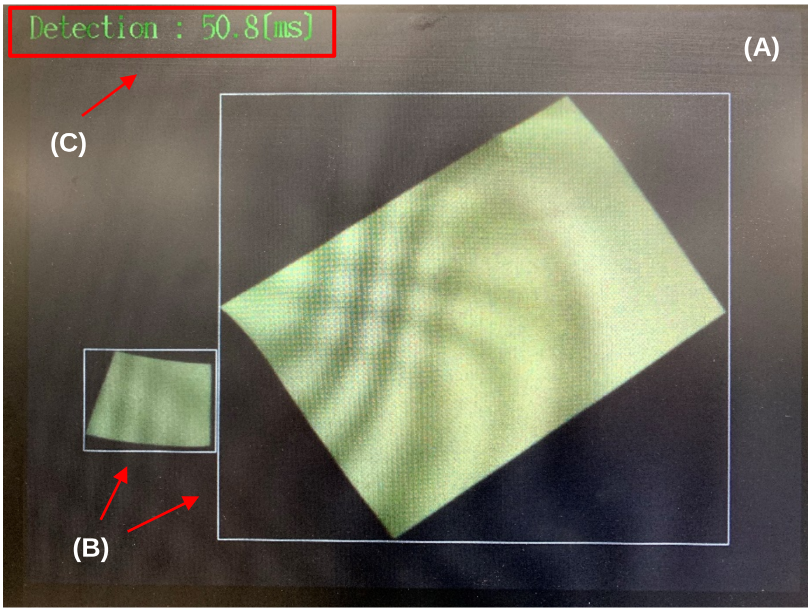


Figure 4.15 Monitor Display Details

(A) Displays the image taken by the camera. The size of the image is 640 x 480 (color image).
(B) If an object is detected, the object is enclosed in a rectangle.
(C) Displays the time of each process required for object detection.

Table 4.3 List of Process in Object Detection

Process Name	Details	Section
Simple ISP	Simple ISP	4.2.1
Resize	Image Resize	4.2.3
Gaussian	Noise Reduction	4.2.4
Canny	Edge Detection	4.2.5
Dilate	Edge Dilation	4.2.6
FindContours	Contour Detection	4.2.7
Bayer2Rgb	Color Image Output	4.2.8

5. DRP Library

This sample program detects objects using the DRP Library shown in Table 5.1 .

Refer to the DRP Library user's manual (R01US0367) for the DRP Library specifications.

Table 5.1 List of functions of the DRP Library to be used

Category	Function name	usage	Section
Simple ISP	SimpleISP	Calculation of correction value used for AE, conversion to gray scale, noise removal.	4.2.1
Image filter	GaussianBlur	remove noise from the reduced grayscale image.	4.2.4
	Dilate	Inflates the white part of the binarized image	4.2.6
Inage conversion	ResizeBilinear	Reducing the image	4.2.3
Feature detection	CannyCalculate	Performs edge detection using Canny method.	4.2.5
	CannyHysterisis		
	FindContours	Detects contours and calculates the circumscribed rectangle	4.2.7

6. Reference Documents

Application note

RZ/A2M Group RZ/A2M Software Core Package (R01AN4775)
The latest version can be downloaded from the Renesas Electronics website.

User's Manual: Software

RZ/A2M Group DRP Driver User's Manual (R01US0355)
The latest version can be downloaded from the Renesas Electronics website.

RZ/A2M Group DRP Library User's Manual (R01US0367)
The latest version can be downloaded from the Renesas Electronics website.

User's Manual: Hardware

RZ/A2M Group User's Manual: Hardware
The latest version can be downloaded from the Renesas Electronics website.

RTK7921053C00000BE (RZ/A2M CPU board) User's Manual
The latest version can be downloaded from the Renesas Electronics website.

RTK79210XXB00000BE (RZ/A2M SUB board) User's Manual
The latest version can be downloaded from the Renesas Electronics website.

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C
The latest version can be downloaded from the ARM website.

ARM CortexTM-A9 Technical Reference Manual Revision: r4p1
The latest version can be downloaded from the ARM website.

ARM Generic Interrupt Controller Architecture Specification - Architecture version 2.0
The latest version can be downloaded from the ARM website.

ARM CoreLinkTM Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p3
The latest version can be downloaded from the ARM website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User's Manual: Development Tools

Integrated development environment e2studio User’s Manual can be downloaded from the Renesas Electronics website.
The latest version can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.0	May. 31, 2019	-	First edition issued

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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.