Trademarks and Permissions

 , HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes the functions, principles, and workflow of the Hi3518_ISP_3A. 3A functions indicate automatic exposure (AE), automatic white balance (AWB), and automatic focus (AF).

Related Version

The following table lists the product version related to this document.

Product Name	Version
Hi3518_ISP_3A	-

Intended Audience

This document is intended for:

- Technical support personnel
- R&D engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 DANGER	Alerts you to a high risk hazard that could, if not avoided, result in serious injury or death.
 WARNING	Alerts you to a medium or low risk hazard that could, if not avoided, result in moderate or minor injury.

Symbol	Description
CAUTION	Alerts you to a potentially hazardous situation that could, if not avoided, result in equipment damage, data loss, performance deterioration, or unanticipated results.
TIP	Provides a tip that may help you solve a problem or save time.
NOTE	Provides additional information to emphasize or supplement important points in the main text.

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 02 (2013-09-25)

This issue is the second official release.

Chapter 2 Usage Guidance

In section 2.1, descriptions are added.

Chapter 3 Developer Guidance

In section 3.2, notes are added to the **Example** field.

Chapter 4 Appendix

This chapter is added.

Issue 01 (2013-07-16)

This issue is the first official release.

Contents

Figures

Tables

1 Introduction to the Hi3518_ISP_3A

1.1 Overview

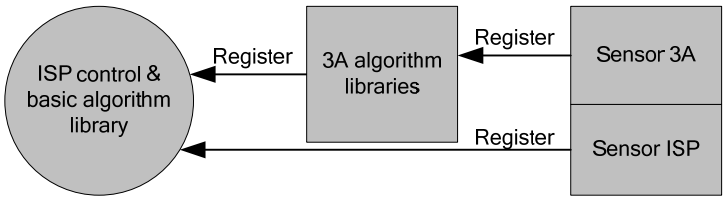
The Hi3518_ISP_3A depends on the software development kit (SDK) of a specific major release. It processes digital images by using a series of digital image processing algorithms. The Hi3518_ISP_3A includes the firmware framework and HiSilicon 3A libraries. The firmware processes statistics, drives digital image processing algorithms, and provides the basic algorithm framework and various functions such as defect pixel correction, denoising, color enhancement, and lens shading correction. The 3A libraries are registered with the firmware to implement exposure, white balance, and color reproduction.

1.2 Function Description

1.2.1 Design Methodology

The image signal processor (ISP) firmware consists of the ISP library (including the control unit and basic algorithm unit), AE/AWB/AF algorithm libraries (3A algorithm libraries), and sensor library. According to the firmware design methodology, separate 3A algorithm libraries are provided. The ISP control unit schedules the basic algorithm unit and 3A algorithm libraries, and the sensor library registers callback functions with the ISP library and 3A algorithm libraries to support various sensors. See Figure 1-1.

Figure 1-1 Design methodology of the ISP firmware

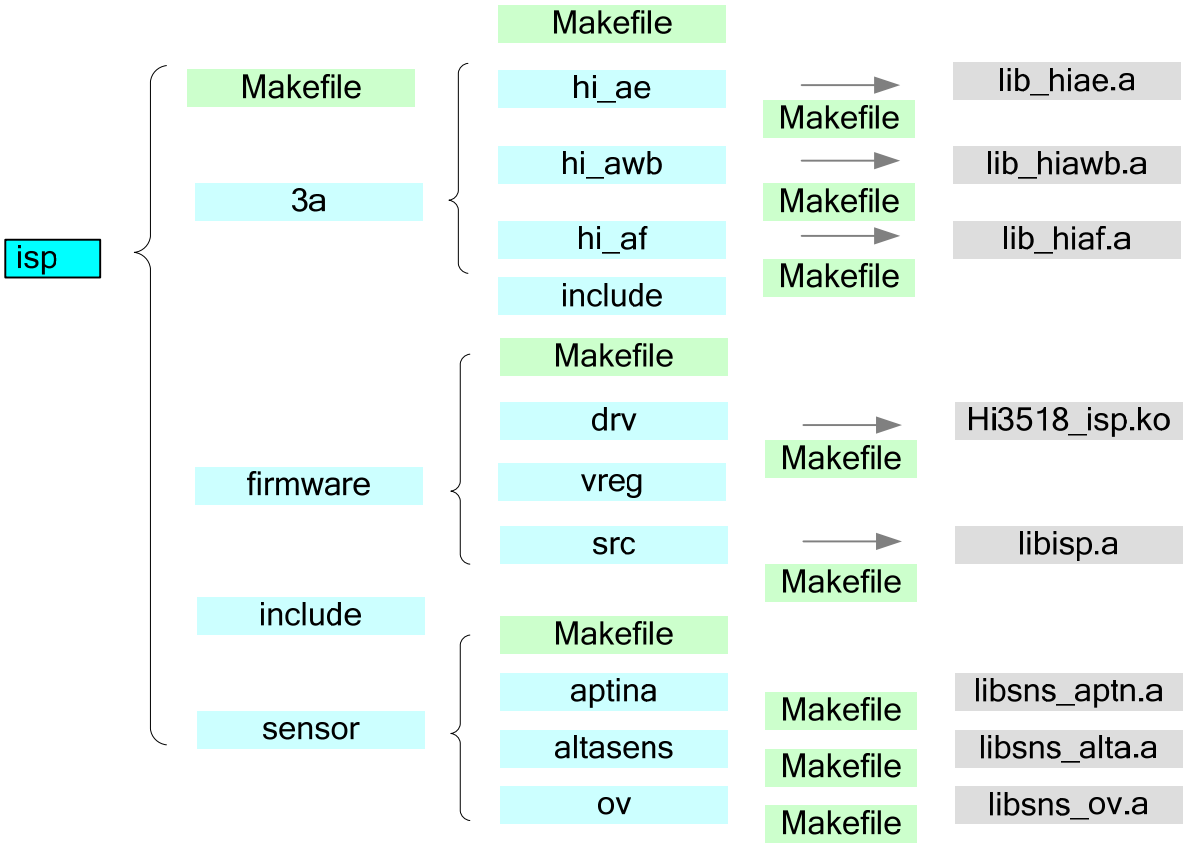


Different sensors register callback control functions with the ISP library and 3A algorithm libraries. When the ISP control unit schedules the basic algorithm unit and 3A algorithm libraries, the callback functions are called to obtain initialization parameters and control sensors, for example, adjusting the exposure time, analog gain, and digital gain, controlling the lens step focus, and rotating the iris.

1.2.2 File Structure

As shown in Figure 1-2, the files of the ISP library, 3A algorithm libraries, and sensor library are stored in different folders. The driver generated in the **drv** folder of the firmware reports the ISP interrupt. The interrupt is used to drive the ISP control unit. The ISP control unit obtains statistics from the driver, schedules the basic algorithm unit and 3A algorithm libraries, and notifies the driver of the registers to be configured. The **src** folder includes the code of the ISP control unit and basic algorithm unit. The ISP library **libisp.a** is generated after the code in the **src** folder is compiled. The **3a** folder includes the AE/AWB/AF algorithm library. You can develop your own 3A algorithms based on a unified interface. The **sensor** folder stores the drivers of all sensors as open source code. The driver code of each sensor is complied as a library to facilitate application compilation and connection between the ISP firmware and sensors. You can call related callback functions based on the features of the used sensor.

Figure 1-2 File structure of the ISP firmware



1.2.3 Development Mode

The SDK supports various development modes:

- Uses only the HiSilicon 3A algorithm libraries.
- Develops 3A algorithm libraries based on the 3A algorithm registration interfaces provided in the HiSilicon ISP library.

- Uses both the HiSilicon 3A algorithm libraries and your own 3A algorithm libraries. For example, you can use **lib_hiae.a** to implement AE, and use your own 3A algorithm libraries to implement AWB.

1.2.3.1 Using the HiSilicon 3A Algorithm Libraries

You need to call the sensor adaptation interfaces provided in the ISP basic algorithm unit and HiSilicon 3A algorithm libraries to adapt to various sensors. The **sensor** folder includes the following two key files:

- **sensor_cmos.c**

  This file is used to implement the callback functions required by the ISP. The callback functions include the adaptation algorithms of sensors and vary according to sensors.

- **sensor_ctrl.c**

  This file is the underlying driver of sensors and is used to read, write to, and initialize sensors. You can develop these two files based on the data sheets of sensors and seek help from sensor vendors.

1.2.3.2 Developing 3A Algorithm Libraries

You need to call the sensor adaptation interfaces provided in the ISP basic algorithm unit to adapt to various sensors. Data interfaces and callback functions need to be customized for the developed 3A algorithm libraries to adapt to and control various sensors.

NOTE

The senior engineers who are familiar with statistics and capable of algorithm development can develop algorithm libraries based on firmware statistics.

1.2.3.3 Differences Between the Original Hi3518 ISP and the New Hi3518 ISP with Separate 3A Algorithm Libraries

The previous Hi3518 SDK provides the ISP whose 3A algorithms are included in the ISP library. Note the following when you use the new Hi3518 ISP with separate 3A algorithm libraries:

- **Makefile** may need to be modified. Only **libisp.a** needs to be linked for the old ISP, and **libisp.a**, **lib_hiae.a**, **lib_hiawb.a**, and **lib_hiaf.a** need to be linked for the new ISP.
- The required header files are added. Only the header files **mpi_isp.h**, **hi_comm_isp.h**, **hi_comm_sns.h**, and **hi_sns_ctrl.h** are required for the old ISP. In addition to the preceding header files, extra header files **hi_comm_3a.h**, **hi_ae_comm.h**, **hi_af_comm.h**, **hi_awb_comm.h**, **mpi_ae.h**, **mpi_af.h**, and **mpi_awb.h** are required for the new ISP.
- The initialization process changes. In the old initialization process, sensor_init and sensor_register_callback need to be called. In the new initialization process, besides sensor_register_callback, HI_MPI_AE_Register, HI_MPI_AWB_Register, and HI_MPI_AF_Register need to be called to register AE, AWB, and AF algorithms respectively. For details about samples, see section 2.1 "Software Process."
- The process for switching between wide dynamic range (WDR) and linear modes changes. In the old switching processing, sensor_mode_set, sensor_register_callback, and HI_MPI_ISP_Init need to be called. In the new switching processing, only HI_MPI_ISP_SetWdrAttr needs to be called.
- The usage of other media processing platform interfaces (MPIs) is the same as that of old MPIs.

- If you have adjusted the parameters in the old ***xxx*_cmos.c** file, you need to modify the new ***xxx*_cmos.c** file based on the adjustments. For details about data structures, see section 2.2 "Sensor Interconnection" or the *HiISP Development Reference*.

1.2.4 Internal Process

Figure 1-3 shows the internal process of the ISP firmware. Firstly, the ISP control unit, basic algorithm unit, and 3A algorithm libraries are initialized, and the sensor callback function is called to obtain sensor initialization parameters. After initialization, driven by the interrupt, the firmware obtains statistics of each frame in kernel mode, drives the basic algorithm unit and 3A algorithm libraries to complete calculation, and provides the calculation results for configuring ISP registers and sensor registers. You can call ISP MPIs to control and change the internal data and status of the basic algorithm unit of the firmware, achieving the required image quality. If you use the HiSilicon 3A algorithm libraries, you can call the MPIs related to 3A algorithm libraries to change its internal data and status, adjusting the exposure, white balance, and color reproduction effects.

Figure 1-3 Internal process of the ISP firmware

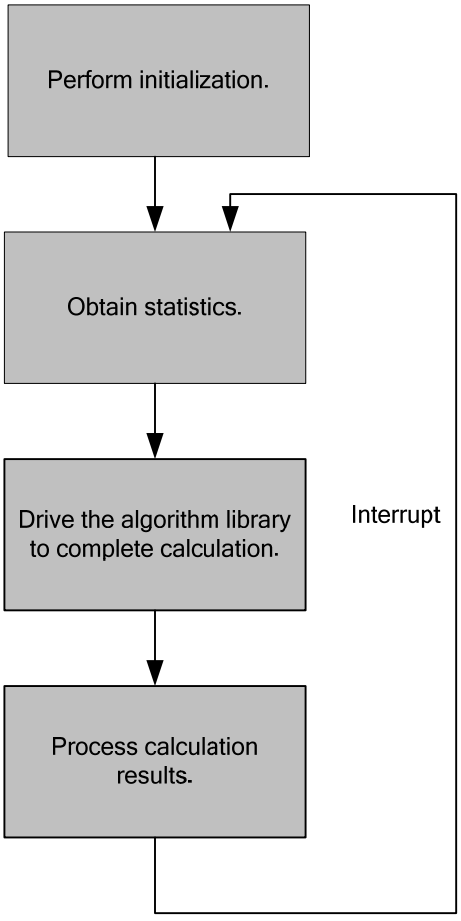
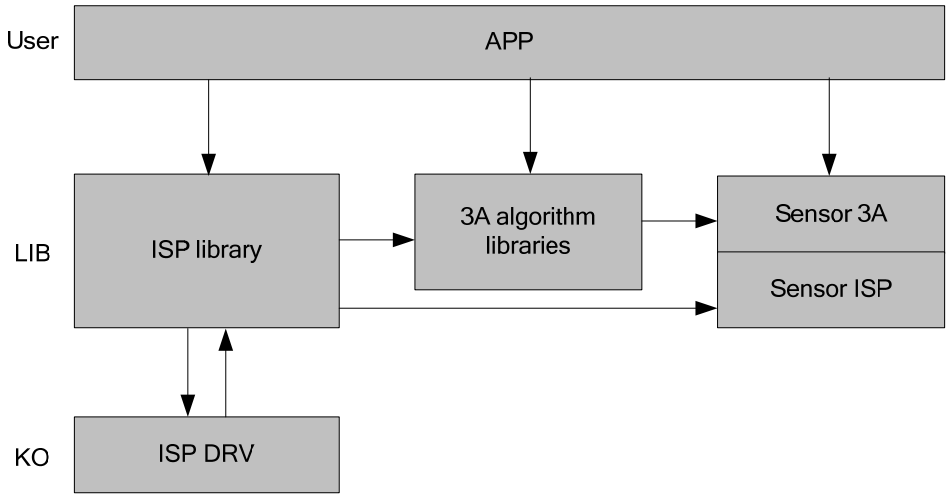


Figure 1-4 shows the architecture of the ISP firmware.

Figure 1-4 Architecture of the ISP firmware



2 Usage Guidance

2.1 Software Process

As the front-end capturing unit, the ISP must work with the video input unit (VIU). After the ISP is initialized and configured, the interface timings of the VIU must be configured. This ensures that the input timings of different sensors are compatible, and the input timings of the ISP are correct. After timings are configured, the ISP can be started to dynamically adjust the image quality. The output images are captured by the VIU, stored in the DDR, and then transmitted for displaying or encoding. Figure 2-1 shows the software process.

The PC Tuning tool dynamically adjusts the image quality at the PC end by tuning the values of related parameters such as the denoising strength, color conversion matrix, and saturation. If the PC Tuning tool is not provided in the product release phase, you can call the image quality adjustment MPI to tune the image effect.