



# PQ Tools User Guide

Version: 1.1.11

Release date: 2023-02-08

Copyright © 2020 CVITEK Co., Ltd. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of CVITEK Co., Ltd.

# Contents

<b>1</b>	<b>Disclaimer</b>	<b>2</b>
<b>2</b>	<b>Overview</b>	<b>3</b>
2.1	Tool Overview . . . . .	3
2.2	Environment Preparation Instructions . . . . .	4
2.2.1	Software and Hardware Requirements . . . . .	4
2.2.2	Physical Link Connection . . . . .	5
2.2.3	EVB Uart Connection Description . . . . .	5
2.2.4	Release Package Description . . . . .	5
2.2.5	Installation and operation of board-side software on Linux systems . . . . .	6
2.3	PC Software Installation . . . . .	7
2.4	Quick Start . . . . .	8
2.4.1	Interface of Connecting Single Board . . . . .	8
2.4.2	Tool Main Interface . . . . .	9
2.4.3	Common Operations . . . . .	10
2.4.3.1	Save Tuning Data Files . . . . .	10
2.4.3.2	Open Tuning Data File . . . . .	10
2.4.3.3	Connect the Single Board . . . . .	11
2.4.3.4	Disconnect from the Single Board . . . . .	11
2.4.3.5	Image Preview . . . . .	12
2.4.3.6	Capture Tool . . . . .	12
2.4.3.7	Bin Import and Export . . . . .	13
2.4.3.8	Calibration Tool . . . . .	13
2.4.3.9	Auxiliary Tools . . . . .	14
2.4.3.10	Shortcuts . . . . .	15
<b>3</b>	<b>Interface and Function Description</b>	<b>16</b>
3.1	Online Tuning Interface and Function Description . . . . .	16
3.1.1	Tuning Structure Interface . . . . .	16
3.1.2	Register / Algorithm Parameter Adjustment . . . . .	18
3.1.2.1	View and Modify Register / Algorithm Parameters . . . . .	19
3.1.2.2	Read Data from Board End . . . . .	21
3.1.2.3	Write Parameters to the Board End . . . . .	21
3.1.2.4	Parameter Temporary Storage and Recovery . . . . .	21
3.1.3	Batch Reading and Writing and Automatic Writing of Tuning Data . . . . .	22
3.1.4	Curve Visualized Tuning . . . . .	24
3.1.4.1	General Curve Function Description . . . . .	24
3.1.4.2	Gamma Visual Tuning . . . . .	25
3.1.4.3	VO Gamma . . . . .	26
3.1.5	Exposure Data Status and Parameter Setting . . . . .	28
3.1.5.1	Exposure Data Status . . . . .	28

3.1.5.2	Exposure Parameter Setting . . . . .	29
3.1.6	White Balance Data Status and Parameter Setting . . . . .	29
3.1.6.1	White Balance Data Status . . . . .	29
3.1.6.2	White Balance Parameter Setting . . . . .	30
3.1.7	WDR Exposure Parameter Setting and Local-Tone Switch . . . . .	31
3.1.7.1	WDR Exposure Parameter Setting . . . . .	31
3.1.7.2	Local-tone Switch . . . . .	31
3.1.8	3DNR and YNR Noise Reduction Tuning . . . . .	32
3.1.9	Y Sharpen Sharpening and Edge Enhancement Tuning . . . . .	33
3.1.10	Top Page . . . . .	34
3.1.11	VPSS Adjustment Page . . . . .	34
3.1.12	LDC Page . . . . .	35
3.2	ISP Calibration Tool . . . . .	35
3.3	Advanced Functions . . . . .	35
3.3.1	Export and Import of Parameters . . . . .	35
3.3.2	Communication Log . . . . .	35
3.4	Instructions for Other Auxiliary Tools . . . . .	36
3.4.1	Instructions for Use of Capture Tool . . . . .	36
3.4.1.1	Tool Interface . . . . .	36
3.4.1.2	Capture YUV Image Data . . . . .	37
3.4.1.3	Capture Raw Image Data . . . . .	38
3.4.1.4	Dump All . . . . .	38
3.4.1.5	File Name Format Description of YUV Image and RAW Image . . . . .	38
3.4.1.6	File Format and Content Description of Raw Info File . . . . .	39
3.4.2	Video Player Instructions (VLC) . . . . .	40
3.4.2.1	Steps to use VLC to play video stream on board . . . . .	40
3.4.3	Instructions for Auxiliary Focusing Tool . . . . .	42
3.4.3.1	Tool Interface . . . . .	42
3.4.3.2	Use the Auxiliary Focusing Tool to Focus . . . . .	43
3.4.3.3	Focusing Auxiliary Operation . . . . .	43
3.4.4	3A Analyser Tool . . . . .	44
3.4.4.1	Tool Interface . . . . .	44
3.4.4.2	Obtain Images and Statistics . . . . .	44
3.4.4.3	View AE Statistics . . . . .	45
3.4.4.4	View AWB Statistics . . . . .	46
3.4.5	Bracketing . . . . .	48
3.4.6	Continuous Raw Tool . . . . .	49
3.5	Detail Description of Parameters . . . . .	50
<b>4</b>	<b>Tool Application Reference</b>	<b>57</b>
4.1	How to Import and Export Tool Parameters? . . . . .	57
4.1.1	Import and Export Parameters Using Tool . . . . .	57
4.1.2	Import and export image quality parameters using the library . . . . .	58
4.1.3	API Reference . . . . .	59
4.1.3.1	CVI_BIN_GetBinTotalLen . . . . .	59
4.1.3.2	CVI_BIN_ExportBinData . . . . .	60
4.1.3.3	CVI_BIN_ImportBinData . . . . .	61
4.1.3.4	CVI_BIN_SaveParamToBin . . . . .	63
4.1.3.5	CVI_BIN_LoadParamFromBin . . . . .	64
4.1.3.6	CVI_BIN_LoadParamFromBinEx . . . . .	66
4.1.3.7	CVI_BIN_SetBinName . . . . .	66

4.1.3.8	CVI_BIN_GetBinName . . . . .	67
4.1.3.9	CVI_BIN_GetBinExtraAttr . . . . .	68
4.1.3.10	Error code 1 . . . . .	69
4.1.3.11	Error code 2 . . . . .	69
<b>5</b>	<b>FAQ</b>	<b>70</b>
5.1	Why Does the Tool Indicate a Version Mismatch When Opening the Debug Data File? . . . . .	70
5.2	Why Does the Tool Prompt SoC Mismatch When Connecting To the Board? . . . . .	72

## Revision History

Re- vi- sion	Date	Description
0.1	2020/02/07	Initial
0.2	2020/04/26	Supplement Chapter 1 and Chapter 2; Remove some sections
0.2.0.1	2020/04/26	Adjust Chapter 1 / 2
0.2.1.2	2020/05/17	Add section 2.4.2 (video stream preview tool)
0.2.1.3	2020/07/22	Update sections 1.2, 2.1.2, 2.4.1; Add sections 2.1.5, 2.1.6, 2.1.7, 2.1.8 and 2.1.9
0.2.1.4	2020/08/26	Add section 2.1.10
0.2.1.5	2021/01/15	Add sections 2.4.3 and 2.6
0.2.1.5	2021/01/15	Update the picture and description according to the version CviPQ-Tool_20210117A
0.2.1.7	2021/02/22	Update section 2.5
0.2.2	2021/03/04	Add chapter 4 FAQ
0.2.2.1	2021/03/24	Update sections 1.2.4, 2.1.4.1 and 2.1.4.2
0.2.2.2	2021/03/24	Update section 2.4.1.3
0.2.2.3	2021/03/24	Update section 1.1
0.2.3.0	2021/04/28	Add sections 4.1 and 4.2
0.2.3.1	2021/05/08	Add section 2.4.1.4, Dump All instructions
1.1.1	2021/06/17	Update version number
1.1.2	2021/10/27	1.3, 1.4, 2.1.1, 2.1.2, 2.1.4, update icons, modify and add features introduced
1.1.3	2021/11/04	Add section 2.1.4.3 VO Gamma, section 2.4.5 Bracketing, section 1.4.3.11 Shortcuts; Update other icons and function descriptions
1.1.4	2021/11/05	Add section 3.1.2 Import and export image quality parameters using the library
1.1.5	2021/12/27	Add section 1.2.4 Release package catalog description
1.1.6	2022/04/24	Add section 3.1.2 API and update some parameters description
1.1.7	2022/05/10	Add section 2.4.6 Continuous raw tool
1.1.8	2022/06/27	Modify some titles
1.1.9	2022/08/17	Update graphics and descriptions
1.1.10	2022/10/20	Remove the bin parameter API that overrides the PQbin
1.1.11	2023/02/08	Add instructions for using the LDC page
1.1.12	2023/04/12	Update the descriptions in 2.2.4 and 2.2.5

# 1 Disclaimer

---



## Terms and Conditions

The document and all information contained herein remain the CVITEK Co., Ltd' s ( "CVITEK" ) confidential information, and should not disclose to any third party or use it in any way without CVITEK' s prior written consent. User shall be liable for any damage and loss caused by unauthority use and disclosure.

CVITEK reserves the right to make changes to information contained in this document at any time and without notice.

All information contained herein is provided in "AS IS" basis, without warranties of any kind, expressed or implied, including without limitation mercantability, non-infringement and fitness for a particular purpose. In no event shall CVITEK be liable for any third party' s software provided herein, User shall only seek remedy against such third party. CVITEK especially claims that CVITEK shall have no liable for CVITEK' s work result based on Customer' s specification or published shandard.

## Contact Us

**Address** Building 1, Yard 9, FengHao East Road, Haidian District, Beijing, 100094, China

Building T10, UpperCoast Park, Huizhanwan, Zhancheng Community, Fuhai Street, Baoan District, Shenzhen, 518100, China

**Phone** +86-10-57590723 +86-10-57590724

**Website** <https://www.sophgo.com/>

**Forum** <https://developer.sophgo.com/forum/index.html>

# 2 Overview

---

## 2.1 Tool Overview

CviPQ Tool is a professional image quality tuning tool.

After connecting the tool with a single board, it can provide users with online debugging of ISP module parameter adjustment, and at the same time, it can watch the effect of parameter setting in real time.

In addition, ISP calibration function is also provided to generate all kinds of data for the modules to be calibrated, which can be provided to the user to adjust the parameters to obtain better image quality.

The architecture of CviPQ Tool is shown in [Fig. 2.1](#) , which is mainly divided into PC Calibration Tools, online Tuning Tools, Analysis Tools and Dump Tools.

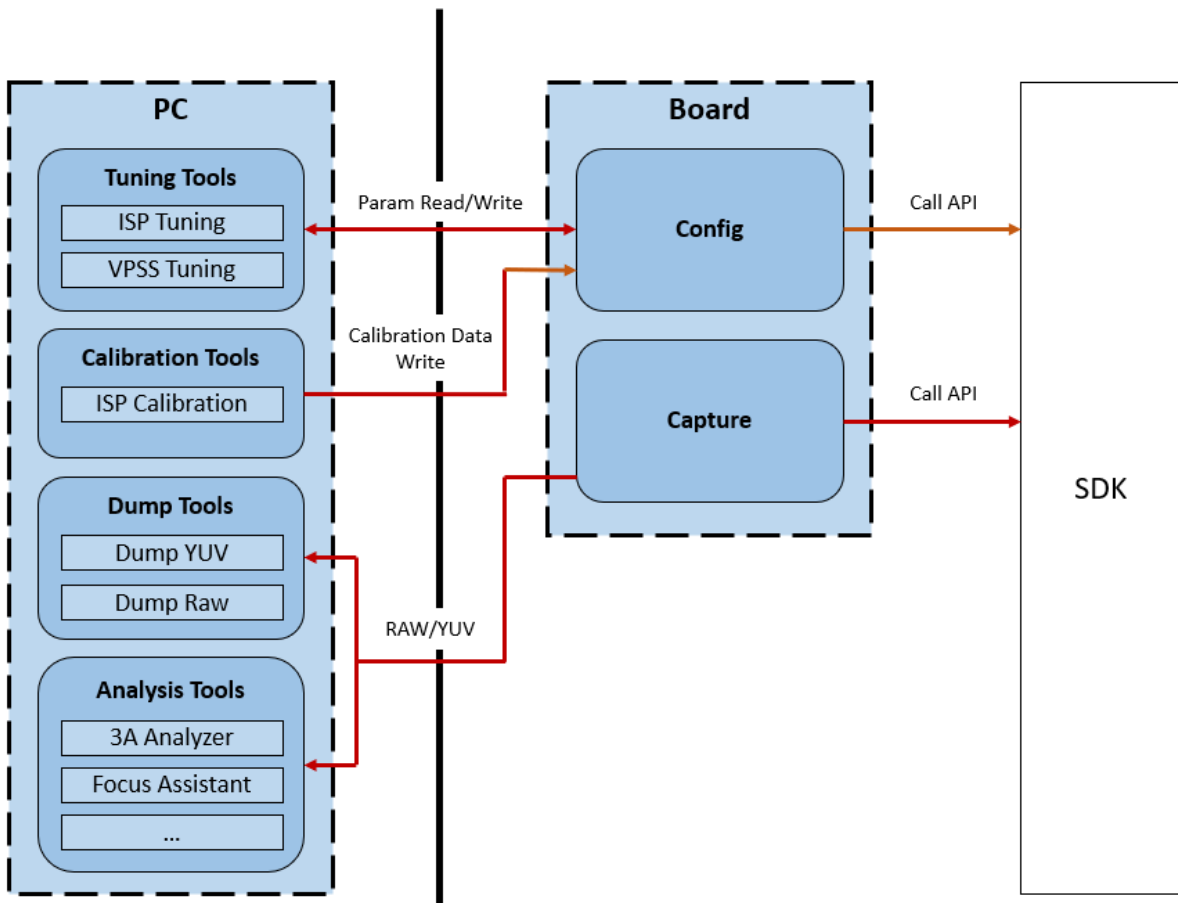


Fig. 2.1: CviPQ Tool Architecture

## 2.2 Environment Preparation Instructions

### 2.2.1 Software and Hardware Requirements

- Hardware Requirements
  - Desktop computer or portable computer
  - Single board hardware (with network port)
  - Network cable
  - A resolution of at least 1024 high and 768 wide for the display, respectively
- Software Requirements
  - Install at least Windows 7 64-bit or above
  - Media player, such as VLC media player



## 2.2.2 Physical Link Connection

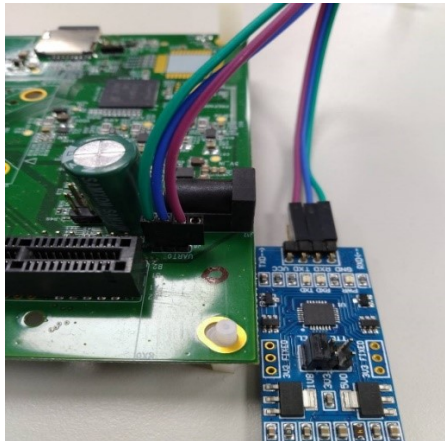
CviPQ Tool is divided into two parts: PC client software and board service software, which interact through network communication.

Physical link connection can use direct connection and local area network connection:

- Direct connection:
  - Connect the two ends of the network cable to the network ports of PC and board respectively.
- Local area network:
  - Connect the network port at the board end to the local port (LAN port) of the router with the network cable.
  - If the PC uses wireless network, connect the PC to the wireless hotspot of the current router according to the access method of wireless hotspot; if the PC uses wired network, connect the PC network port and the local port (LAN port) of the router with network cable.

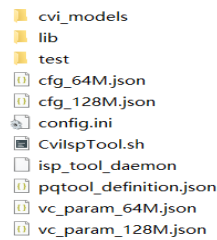
## 2.2.3 EVB Uart Connection Description

Please refer to the picture below to connect UART with EVB. (*take CV183X EVB as an example*). For detailed board interface description, please refer to “EVB hardware guide” .



## 2.2.4 Release Package Description

Copy `isp_tool_daemon.tar.gz` from the release package to the board side, unzip it to generate the install directory, the directory structure is shown in the following figure.



- config.ini is the configuration file when running isp\_tool\_daemon, including configuring the log printing level, setting the default path for PQbin generation, and the path to the sensor type file.
- CvIspTool.sh is the script file to quickly start isp\_tool\_daemon.

## 2.2.5 Installation and operation of board-side software on Linux systems

**Step1.** The directory structure of isp\_tool\_daemon.tar.gz after unpacking is shown in [section 2.2.4](#).

**Step2.** Configure board-side IP, ifconfig eth0 xxx.xxx.xxx.xxx

**Step3.** Enter the unzip and generate the install directory. cd /mnt/sd/install

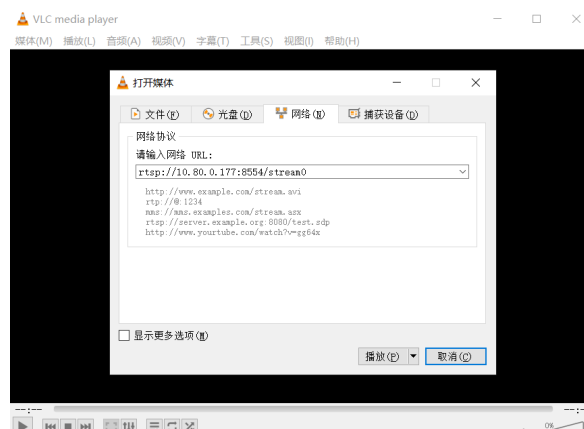
**Step4.** Excute the command “./CviISPTool.sh 128M” to start the board program.

```

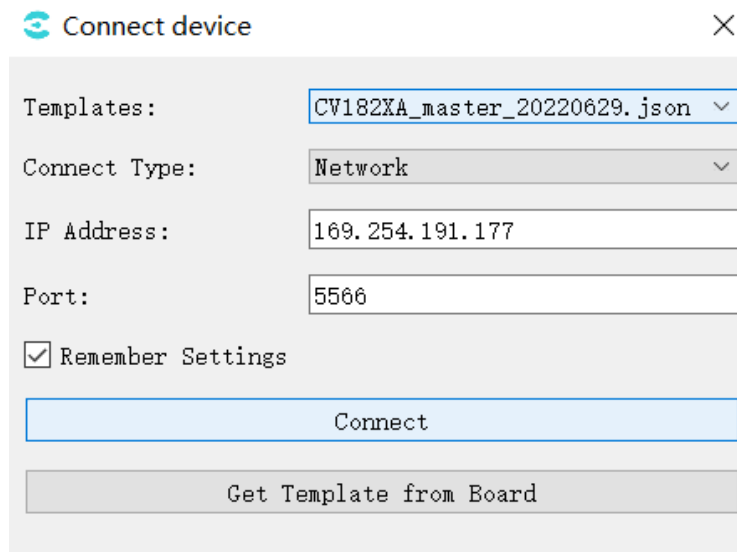
root@cvitek]/#
root@cvitek]/#
root@cvitek]/#
root@cvitek]/# mount /dev/mmcblk0p1 /mnt/sd/
root@cvitek]/# cd /mnt/sd/install
root@cvitek] /mnt/sd/install# ifconfig eth0 192.168.1.177
root@cvitek] /mnt/sd/install# ./CviIspTool.sh 128M

```

**Step5.** Connect VLC to display image



**Step6.** Open CviPQTool on the PC and enter the board IP to connect board



## 2.3 PC Software Installation

Cvipq tool PC software is a green installation-free software. You only need to decompress the software package to any writable directory. Find CviPQTool.exe in the decompressed directory and double-click to run.

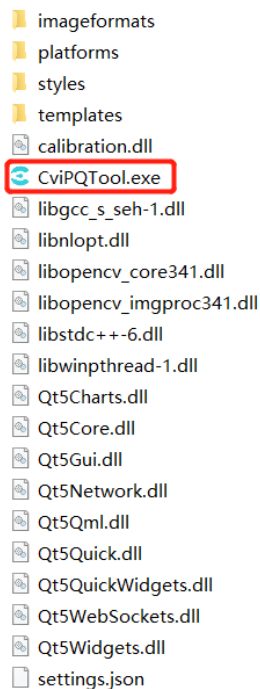


Fig. 2.2: Decompressed directory structure

- CviPQTool.exe is the executive program of CviPQ Tool, which can be run by double-click or right-click.
- settings.json is the configuration file for CviPQ Tool, where the parameters are saved when

the user clicks to remember the settings when connecting to the single board interface

- The directory of imageformats/platforms/styles is the library file of plug-ins that CviPQ Tool depends on. Other DLL library files are mainly the runtime libraries of Qt and opencv.

## 2.4 Quick Start

### 2.4.1 Interface of Connecting Single Board

Whenever the user is running the CviPQTool.exe, the connection board window will pop up as shown in Fig. 2.3 to allow users to quickly connect the board for image quality tuning.

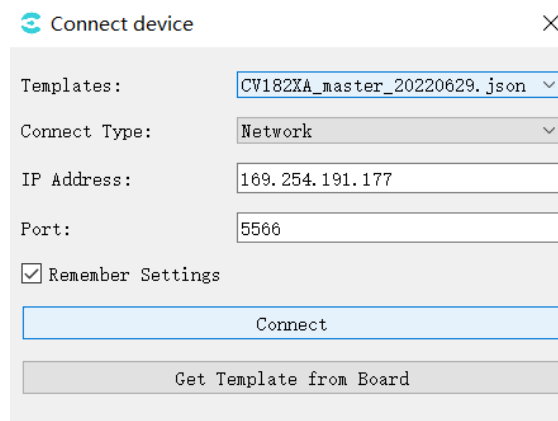


Fig. 2.3: Example of connecting single board interface

**Templates:** Select tool ui template;

**Connect Type:** Network connection;

**IP Address:** Board-side IP address;

**Port:** Port number;

**Remember Settings:** Save the current window settings to settings.json and import them automatically the next time you open the tool.

**Get Template from Board:** Get the ui template from the board ( or manually put the ui json file into the templates directory, restart tool, and then select from the Templates dropdown box)

**After setting the options, click the “Connect” button to enter the main interface of the tool. The tool will automatically connect the PC and the board through the network, and will automatically read the parameter values of all debugging items from the board.**

---

**Note:** When open the tool, the initial default ui template is in Templates, you can click the Get Template from Board button to get the latest matching ui template from the board, then select the json just obtained from the Templates drop-down box, then click the Connect button and wait for the initialized Ui to enter the main interface of the tool.

---

## 2.4.2 Tool Main Interface

The main interface of the tool is shown in Figure 2-4.

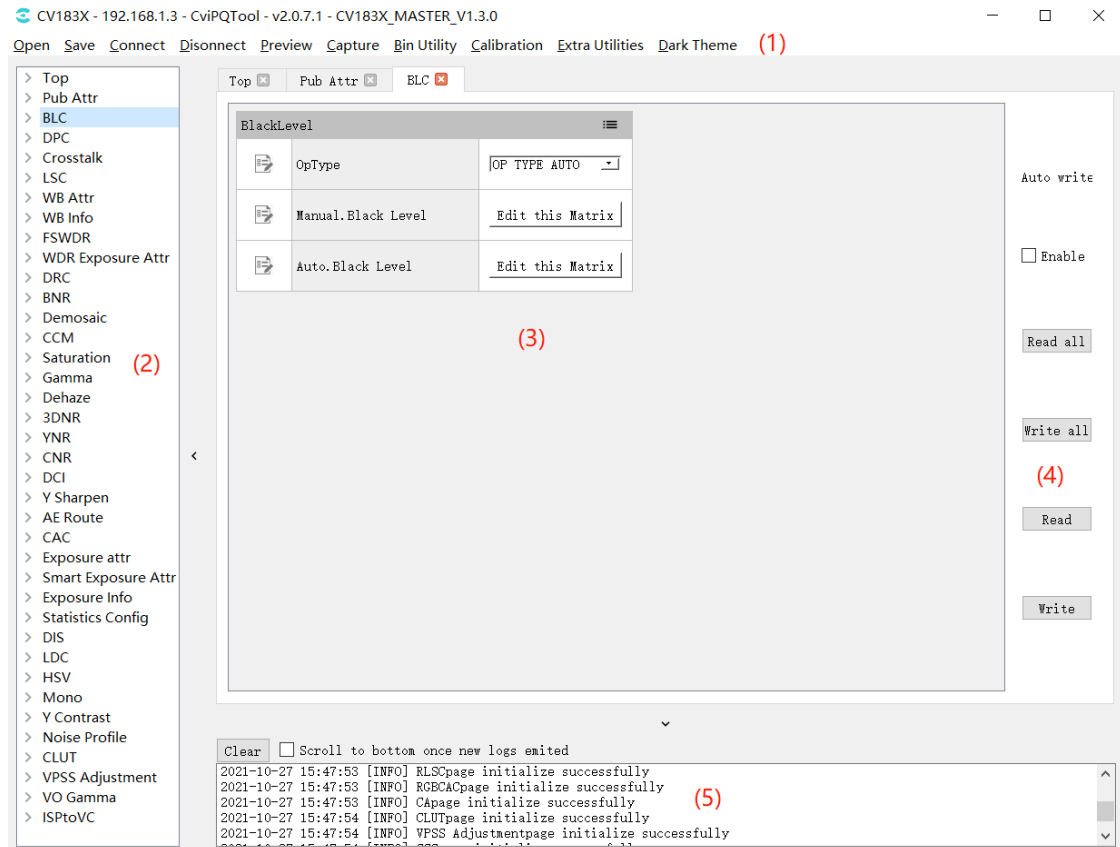


Fig. 2.4: Tool Main Interface

The main interface of CviPQ Tool can be divided into the following areas:

- (1). Toolbar: provides some common shortcut options
- (2). Tuning panel: displays the tunable items of all modules
- (3). Tuning area: this area will display the tuning page corresponding to the module selected in the tuning panel
- (4). Read / Write: automatically read/write and read/write all pages or current page data can be chose here.
- (5). Prompt Bar: display communication log

## 2.4.3 Common Operations

### 2.4.3.1 Save Tuning Data Files

When clicking the “Save” button in the toolbar, a dialog box for selecting a path will pop up. When the user selects a storage path, the tool will save the parameters of the current debug panel table in the format of \*.json, storing the parameters and the structure of the debug table.

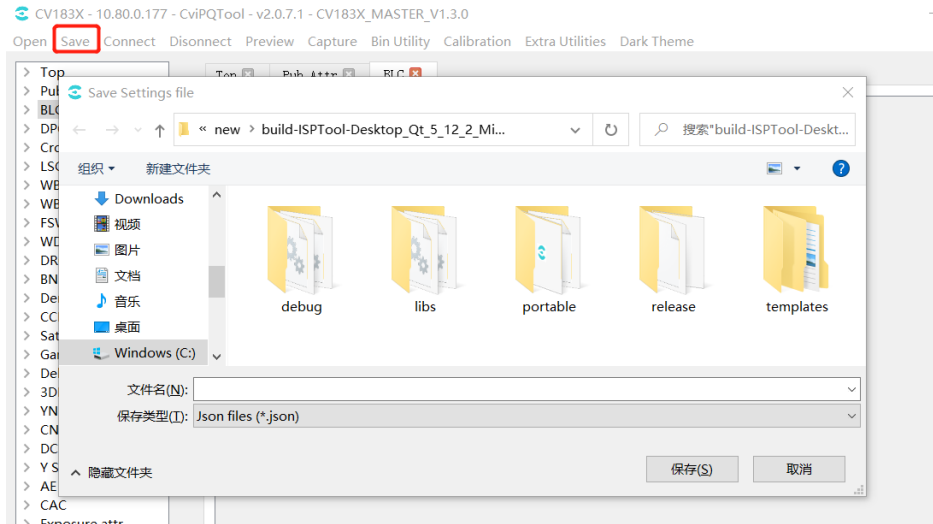


Fig. 2.5: Example of interface for storing tuning data files

### 2.4.3.2 Open Tuning Data File

When clicking the “Open” button in the toolbar, a dialog box will pop up for the user to select the data file to be opened. When the data file is opened, the tool will load the relevant parameters in the file and display them in the tool tuning panel.

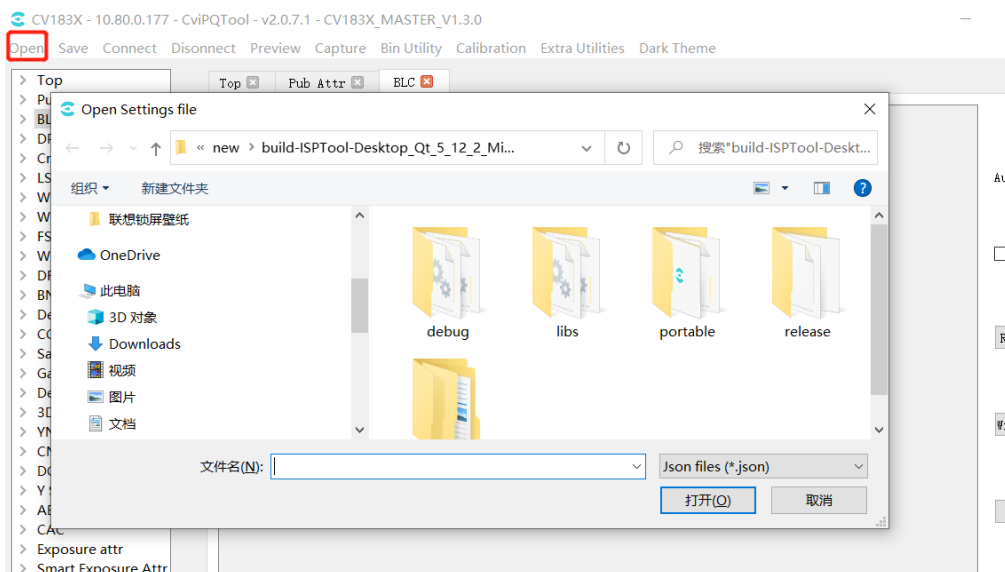


Fig. 2.6: Example of Opening Tuning Data File Interface

### 2.4.3.3 Connect the Single Board

Click the “Connect” button in the toolbar, and the connect board window will pop up, as shown in Fig. 2.7 . The user can enter the IP address of the board in the “IP address” field and the port number in the “Port” field, and then click “Connect” to connect the tool to the single board.

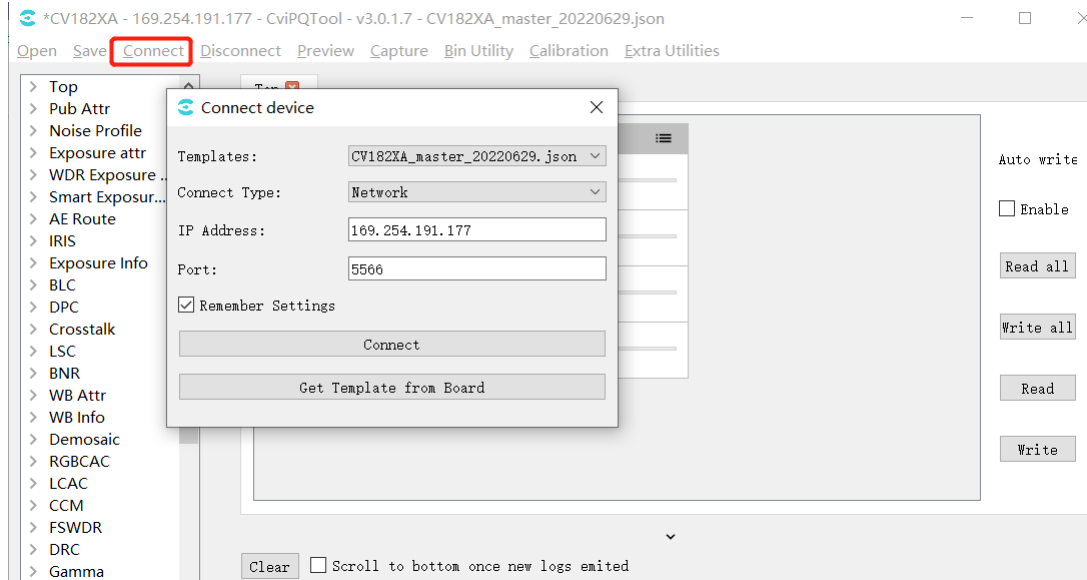


Fig. 2.7: Connect Window

### 2.4.3.4 Disconnect from the Single Board

When the tool is connected to a single board, click on the “Disconnect” button in the toolbar to disconnect the tool from the single board

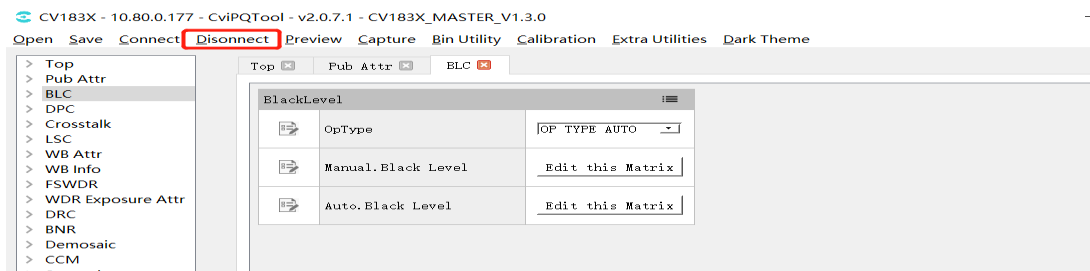


Fig. 2.8: Disconnect button

### 2.4.3.5 Image Preview

Click the “Preview” button on the toolbar, the Preview window will pop up, click the “Get Single Image” button, the image will be captured and displayed. (The tool is connected to the single board, and the single board is connected to the VLC or the display to show the image normally)

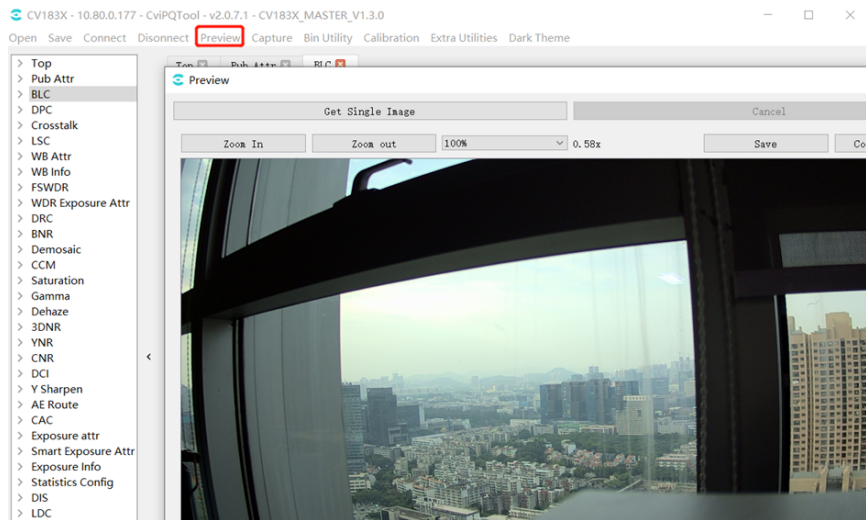


Fig. 2.9: Image preview window

### 2.4.3.6 Capture Tool

Click “Capture” button on the toolbar and the Capture Tool window will pop up as shown in the figure, see [section 3.4.1](#) for details.

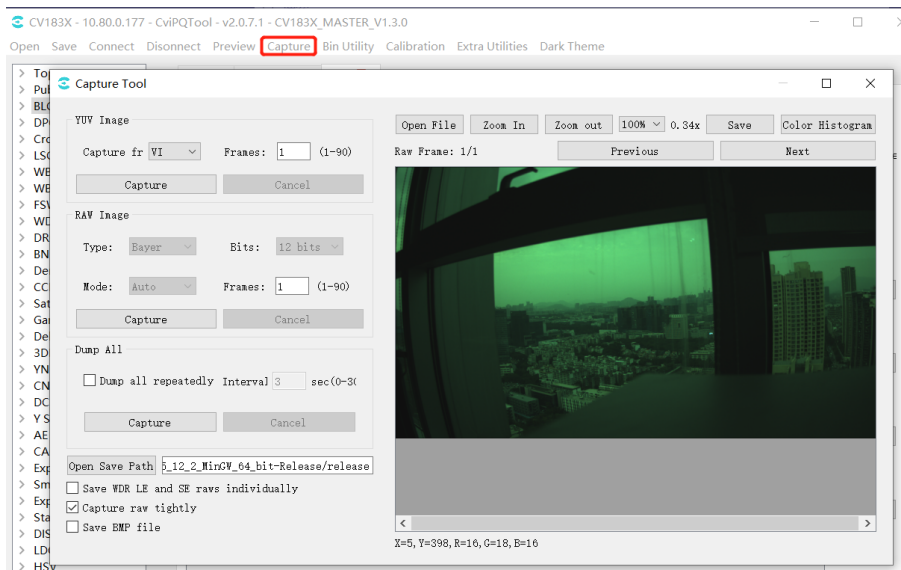


Fig. 2.10: Capture tool window



### 2.4.3.7 Bin Import and Export

Click ” Bin Utility Tool ” button on the toolbar and the Bin Utility Tool window will pop up as shown in the figure

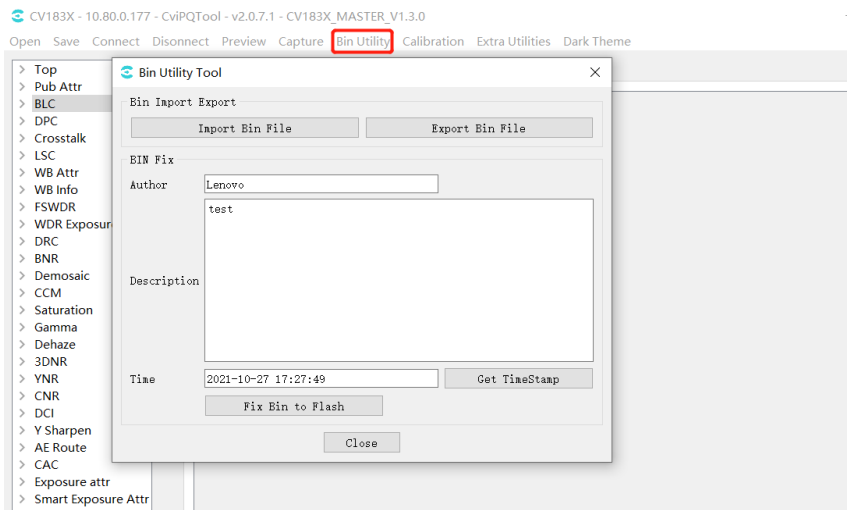


Fig. 2.11: Import and Export

Export Bin File: exporting board-side parameters to the pc and save them as bin files.

Import Bin File: import pc-side bin files to the board side.

Fix Bin To Flash: generate bin files on the board side.

Author,Description,Time are file description information, which will be printed on the serial port when the bin is loaded by running on the board side.

### 2.4.3.8 Calibration Tool

Click ” Calibration ” button on the toolbar and the Calibration Tool window will pop up as shown in the figure, see [section 3.2](#) for details.

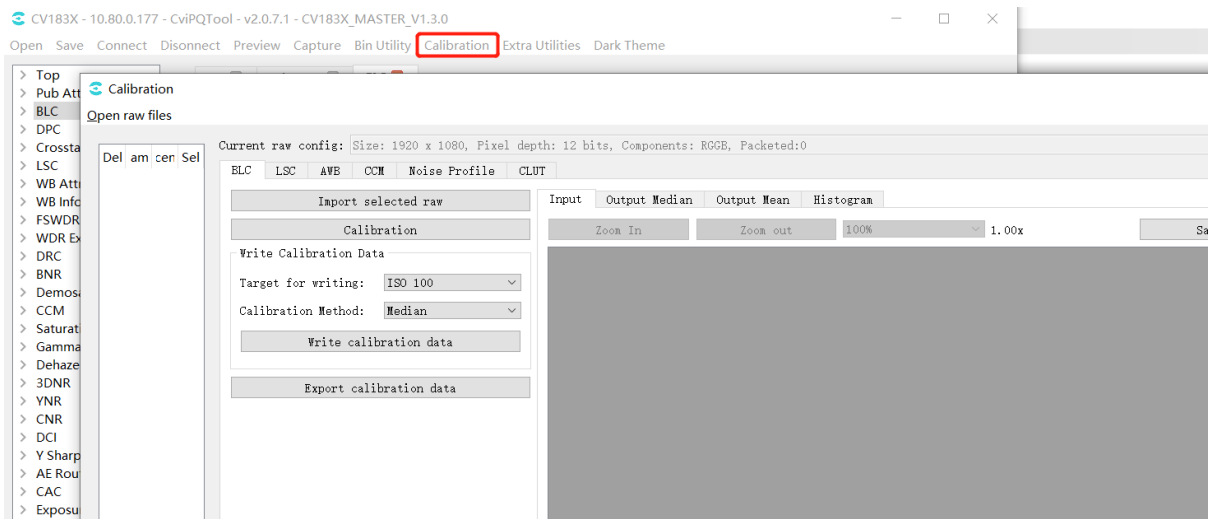


Fig. 2.12: Calibration tool window

### 2.4.3.9 Auxiliary Tools

Click the toolbar “Extra Utilities” button to bring up the drop-down options for 3A Analyser, Focus Assistant, Bracketing, Continuous Raw and DRC Tone Viewer tools, see [section 3.4](#) for details.

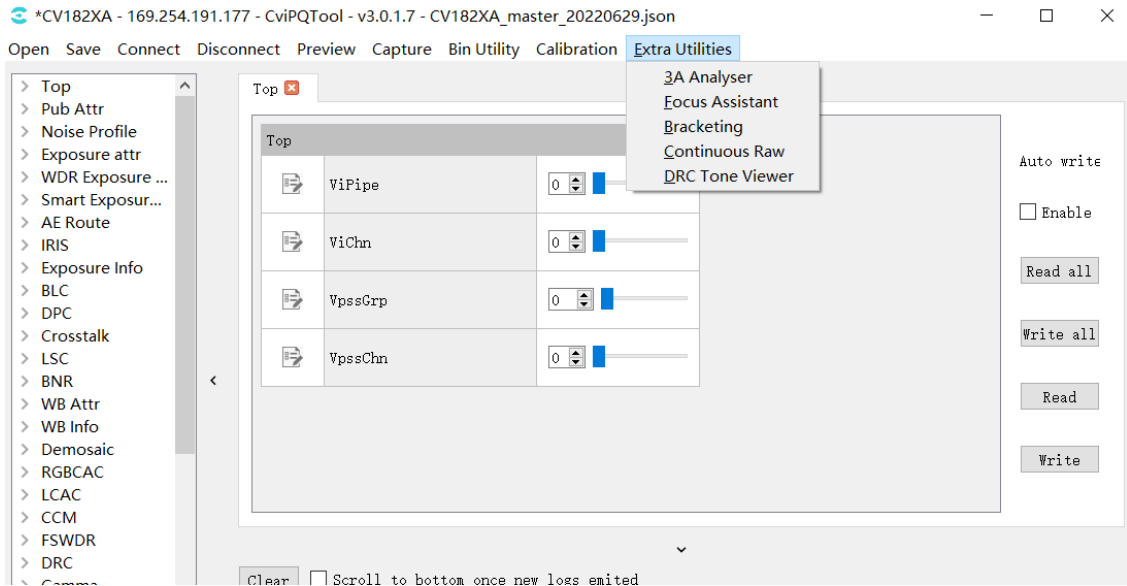
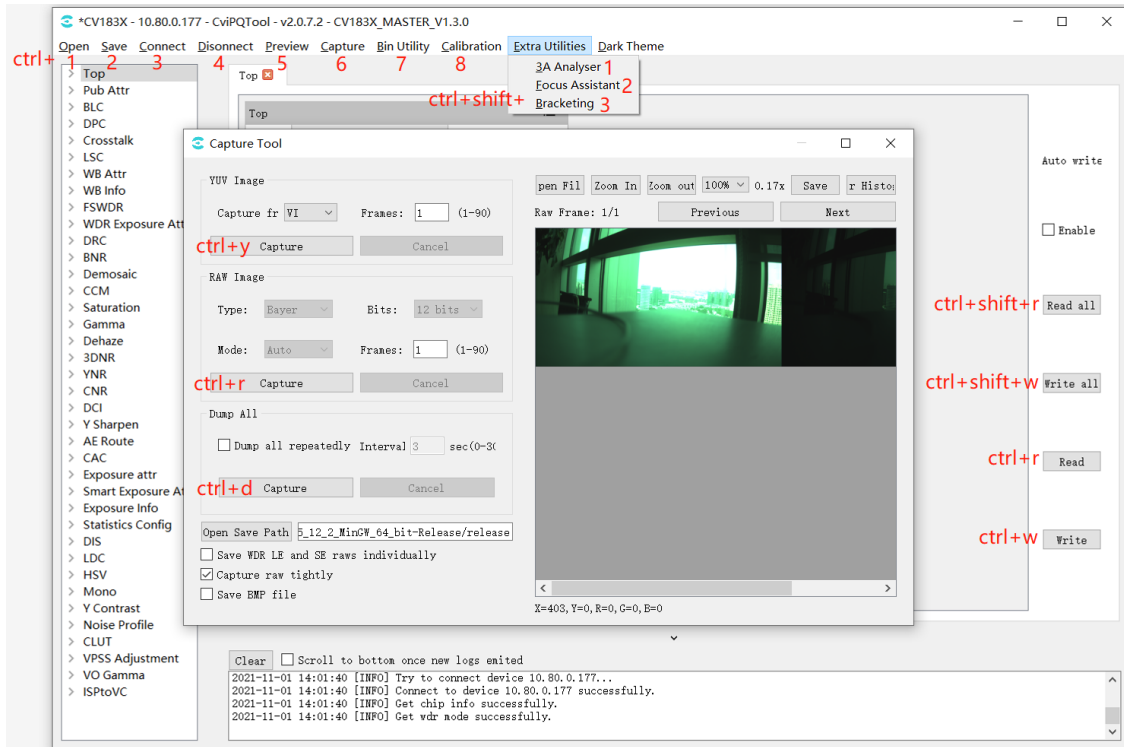


Fig. 2.13: Auxiliary tools window

### 2.4.3.10 Shortcuts

The shortcuts are shown below.



# 3 Interface and Function Description

---

## 3.1 Online Tuning Interface and Function Description

### 3.1.1 Tuning Structure Interface

After opening the CviPQ tool for online operation, the tuning table panel on the left side of the tool displays the tunable items of all modules. The structure is shown in [Fig. 3.1](#).

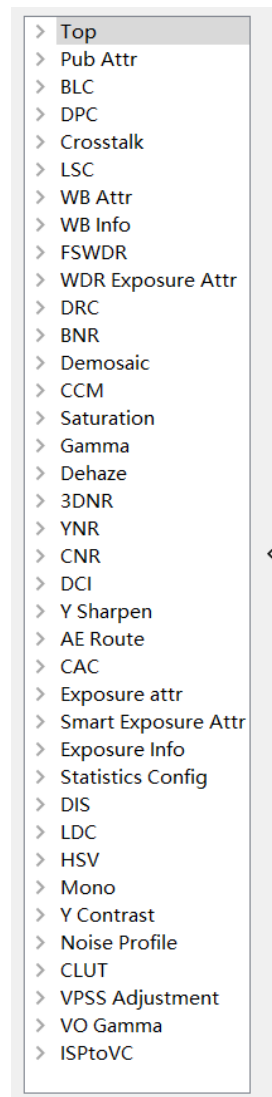


Fig. 3.1: The structure of the left side of the tool

Click the tunable items of these modules, and the corresponding tuning page of the module will be displayed in the tuning area on the right side of the tool, providing users with adjustment parameters, as shown in Fig. 3.2.

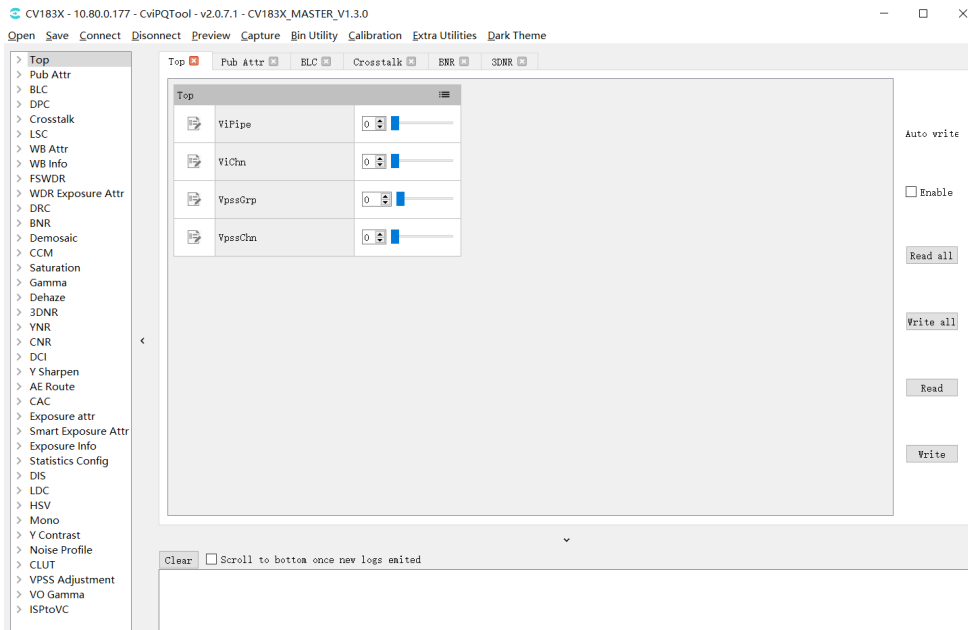


Fig. 3.2: Tuning page

### 3.1.2 Register / Algorithm Parameter Adjustment

Each item in the list on the left side of the tool interface is the “register / algorithm parameters” tuning page. The tuning interface corresponding to these pages is shown in Fig. 3.3. All parameters in the debugging page corresponding to each module will be divided into several groups according to different functions.

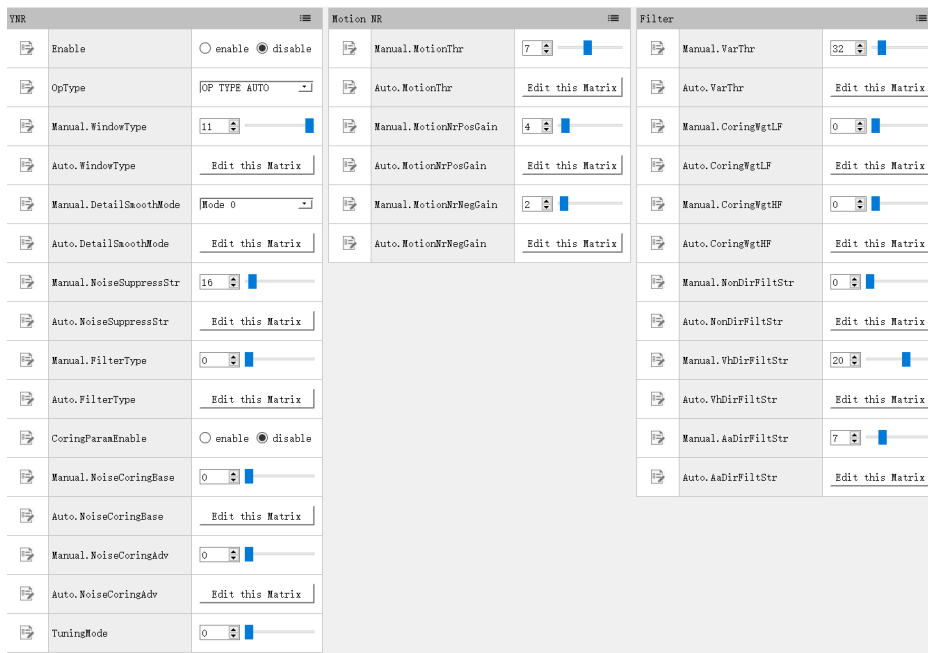


Fig. 3.3: Tuning page of register / algorithm parameters (take YNR as an example)

### 3.1.2.1 View and Modify Register / Algorithm Parameters







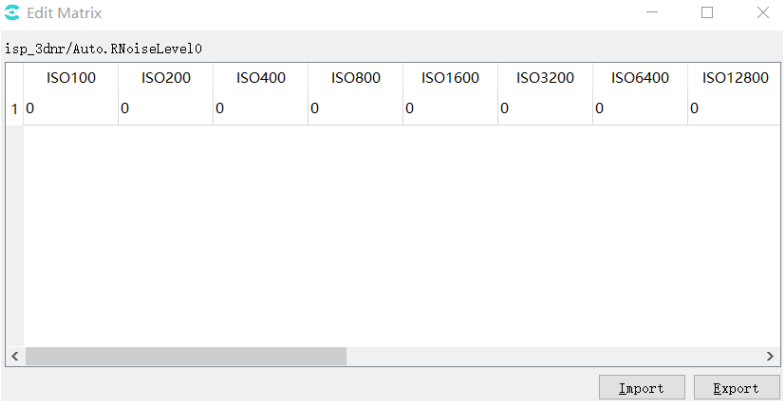

Each group contains registers and algorithm parameters, in which the parameters marked with “” are read-write and the parameters marked with “” are read-only. Parameters are divided into the following categories according to their properties and value ranges, which are presented in different forms of controls, as shown in [Table 3.1](#) .

Table 3.1: Tuning item types of register algorithm parameters

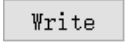
Type	Description	Control Operation																		
Boole	either-or	<p>Set and view through the check box:</p> 																		
Neumerical Value	Real number with a certain range of values	<p>Use the following controls to adjust and view values:</p>  <p>The values can be adjusted in the following wa</p> <ol style="list-style-type: none"> <li>1.Enter the value directly in the text box</li> <li>2.Click the small arrow on the right side of the text box to fine tune up and down</li> <li>3.Drag the right slider</li> </ol>																		
Enumeration	Multiple choice	<p>Set and view values through the drop-down box:</p> 																		
Matrix	Multibyte Sequence	<p>The tuning page will display a button to edit the matrix. If the matrix is read-only, the button to view the matrix will be displayed.</p>  <p>Click the button to open a dialog box, in which the corresponding matrix values are displayed in the form of a table. The user can modify the values in the table. If the matrix is read-only, it can only be viewed.</p> <p>Users can click Import / Export to import and export matrix values, and only the “Export” button is available for read-only matrix.</p>  <table border="1" data-bbox="608 1664 1394 1742"> <thead> <tr> <th></th> <th>ISO100</th> <th>ISO200</th> <th>ISO400</th> <th>ISO800</th> <th>ISO1600</th> <th>ISO3200</th> <th>ISO6400</th> <th>ISO12800</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		ISO100	ISO200	ISO400	ISO800	ISO1600	ISO3200	ISO6400	ISO12800	1	0	0	0	0	0	0	0	0
	ISO100	ISO200	ISO400	ISO800	ISO1600	ISO3200	ISO6400	ISO12800												
1	0	0	0	0	0	0	0	0												



### 3.1.2.2 Read Data from Board End

When the tool is connected to the board end, click the read button (  ) on the rightmost side of each module page. The tool will read the value of each parameter on the current board and update it to the interface.

### 3.1.2.3 Write Parameters to the Board End

When the tool has been connected to the board, click the write button (  ), on the far right of each module page, and the tool will write each writable parameter on the module page to the board.

### 3.1.2.4 Parameter Temporary Storage and Recovery

The tool provides four groups of data temporary storage space for each tuning group. The user can use the “To Set” button to save the adjusted values of parameters, and then use the “Load Set” button to restore and compare. The detailed operation instructions are shown in [Fig. 3.4](#) . Please refer to the following instructions.

1. When this icon is selected, the quick buttons of parameter temporary storage and recovery can be displayed
2. Click the “To Set” button to save the current parameters temporarily. Different values can be stored in different Sets temporarily for quick recovery later.
3. Click the “Load Set” button to restore the previously stored value. Take [Fig. 3.4](#) as an example: If the user has stored “Set1” and “Set2” two groups of values temporarily, “Load Set1” or “Load set2” button can be used to quickly recover and compare them.

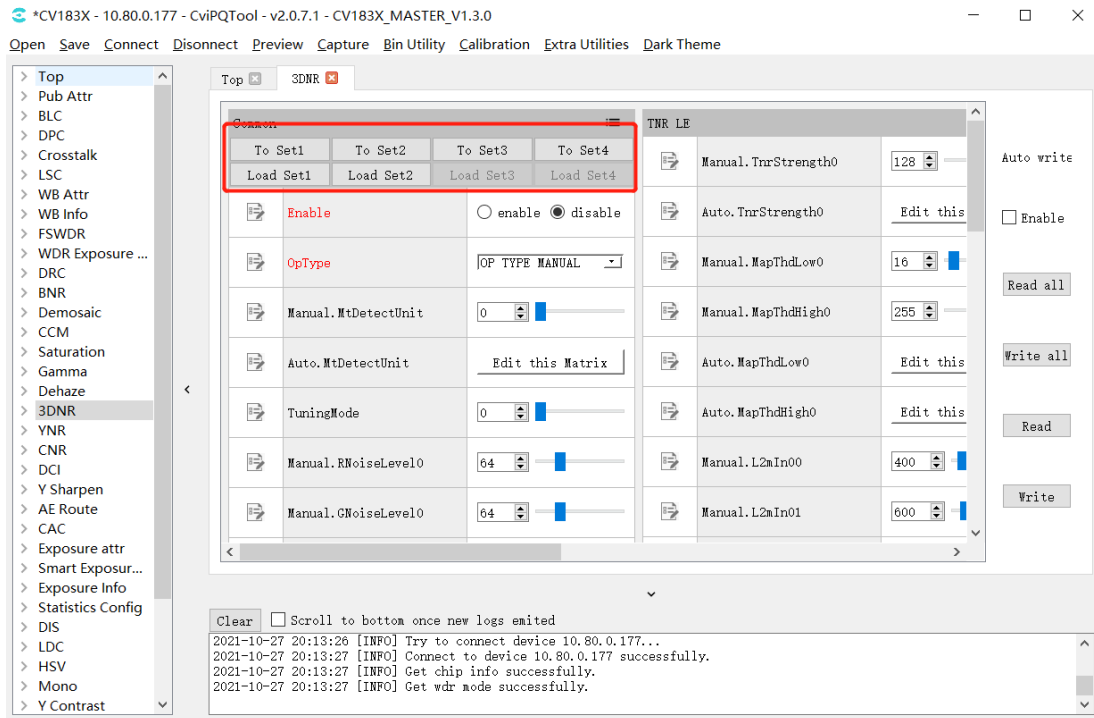


Fig. 3.4: Parameter temporary storage operation interface (taking 3DNR as an example)

### 3.1.3 Batch Reading and Writing and Automatic Writing of Tuning Data

When open the tuning tool and enter the main interface, you can see the data read / write control interface on the far right, as shown in Fig. 3.5.

Auto write

Enable

Read all

Write all

Read

Write

Fig. 3.5: Data read / write control interface

To operate the read / write console, you need to connect the tool to the board first, with the following specific functions:

- **Read All Button:** When this button is clicked, the tool will read the current value of all module page parameter items on the board. It is suggested that before opening the tool for tuning or after importing the Bin file, all read operations should be performed once.
- **Write All Button:** When this button is clicked, the tool will write the values of all module page parameter items to the board.
- **Auto Write Switch:** When this option is checked, the tool will write the new value of the parameter to the board each time a writable parameter item is adjusted. It is recommended to open this item during tuning to ensure that the adjustment value can take effect immediately.
- **Write Button:** When this button is clicked, the tool will write the values of all parameter items in the currently opened module page to the board end.

### 3.1.4 Curve Visualized Tuning

#### 3.1.4.1 General Curve Function Description

The generic curve in the tool is shown in Fig. 3.6. For example, in the 3DNR debug page, click the Edit this Curve button to enter the curve window.

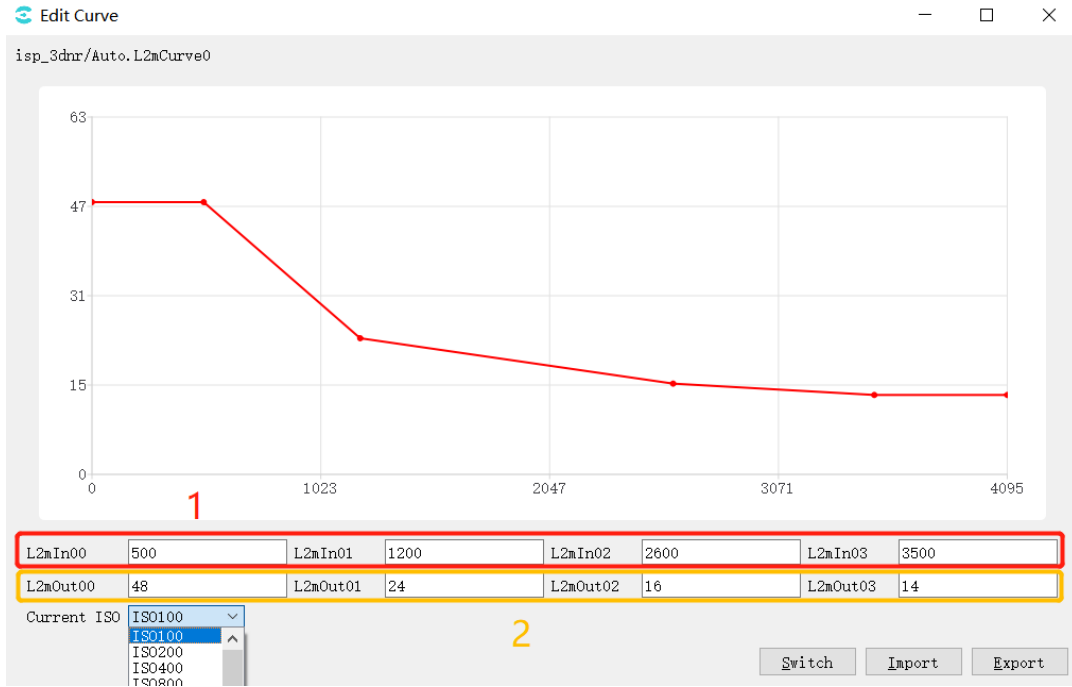


Fig. 3.6: 3DNR page curve operation example diagram

In the figure, 1 is the horizontal coordinate of the curve, 2 is the vertical coordinate of the curve, corresponding to the 4 points in the curve, modify their values, and the points on the curve will change accordingly; you can also drag the points on the curve by the mouse, corresponding to the horizontal and vertical coordinate values change with it.

Current ISO: drop-down option to select different ISO values and view their corresponding curves.

Switch: mode switch button, click in curve mode, enter the following table mode, the first 4 lines of the table are horizontal coordinate values, the last 4 lines are vertical coordinate values, the table can be edited.

	ISO100	ISO200	ISO400	ISO800	ISO1600	ISO3200	ISO6400	ISO12800	ISO25600	ISO51200	ISO102400	ISO204800	ISO409600	ISO819200	ISO1638400	ISO3276800
1	1111	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500
2	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200	1200
3	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600
4	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500
5	63	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48
6	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
7	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
8	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14

Import:import parameters from the file.

Export:export parameters to the file.

### 3.1.4.2 Gamma Visual Tuning

Click Gamma page on the left list of tuning tools to enter Gamma visual tuning page, as shown in Fig. 3.7.

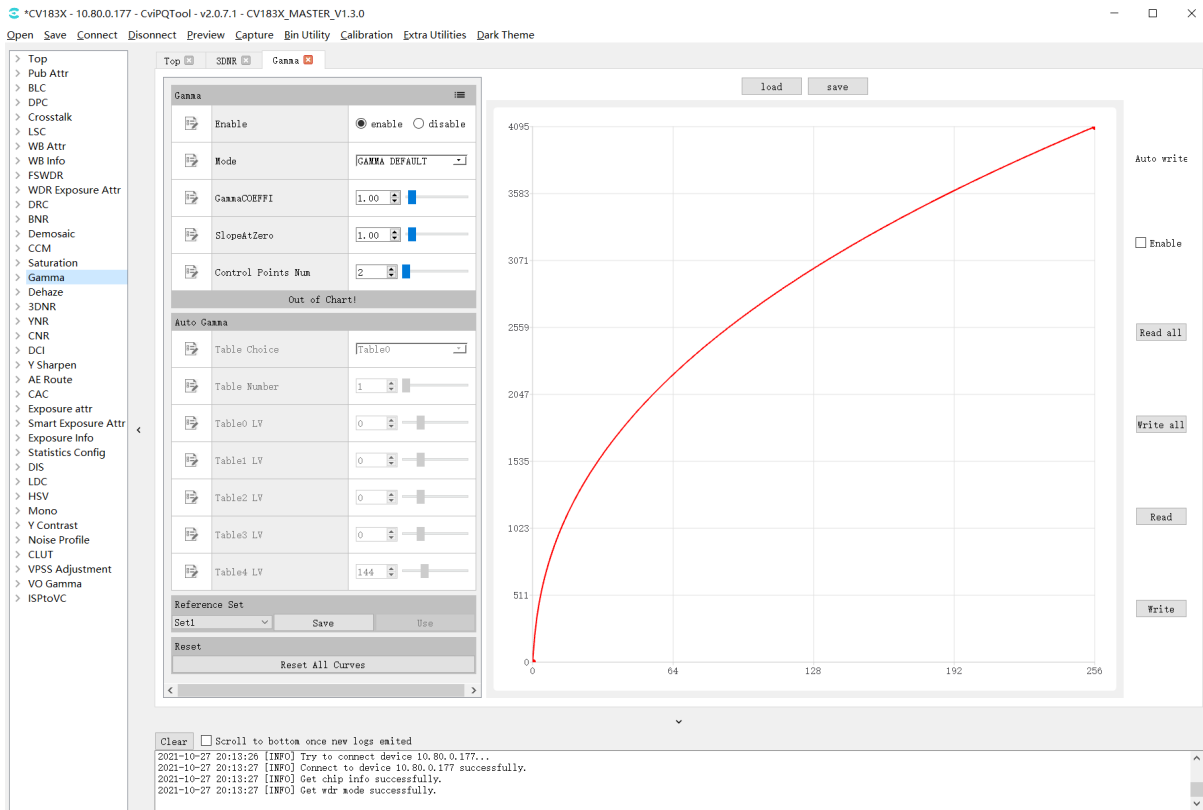


Fig. 3.7: Gamma Tuning Page

The following is the specific parameter description and usage:

- Gamma
- Enable parameter, set it to enable Gamma.
  - Mode parameter, represents the curve adjustment mode, when GAMMA\_DEFAULT/GAMMA\_sRGB is chosen, the curve will be reset to the default curve, and the user can't adjust the curve manually. When GAMMA\_USER\_DEFINE is chosen, the the curve will be reset to a straight line with the starting point of (0,0) and the ending point of the maximum value of X / Y axis. The user can manually adjust the direction of the curve.
- GammaCOEFFI parameter, restore the current curve to a standard gamma curve (coefficient range 0.01 to 20.0) by entering a value in the text box or dragging the slider.
- SlopeAtZero parameter, adjust the slope of gamma curve at zero point (the slope range is 0.01 to 20.0) by inputting a value in the text box or dragging the slider.

- Control Points Num, When GAMMA\_USER\_DEFINE/GAMMA\_AUTO is selected, by adjusting this parameter, the user can set the number of control points that can be operated.
- Auto Gamma (gamma mode set to GAMMA\_AUTO is valid )
  - Table Choice , select gamma table 0 ~ 5 according to light value, currently using the first 4 groups.
  - Table Number, number of tables currently in use.
  - Table0~4 LV, corresponds to 5 light value values (LV:-5~15).
- Reference Set
 

Curve data temporary storage area, the tool provides three sets of temporary storage space from set1 to set3, the user clicks the save button the current curve data will be saved to the selected set, and the user clicks the use button the temporary stored data can be restored to the curve interface. By default, the use button is grayed out and unclickable when no temporary storage operation is performed.
- Reset
 

Reset is the reset button. After clicking, the curve will be reset to a straight line with the starting point of (0,0) and the ending point of the maximum value of X / Y axis.
- Load&Sav
  - Click the “Load” button to read the curve data from the file, write it back to the tool data cache, and refresh the display curve at the same time.
  - Click the “Save” button to save the current curve data to the local text file. The tool supports two formats of data files: Cvitek Json file and Other text file
- Click the “Read” button of the page read / write control interface, the tool will read the curve data from the board end and refresh the curve display.
- Click the “Write button” , and the tool will write the data of the current curve to the board end.
- If “Auto Write” is checked, when the curve is adjusted, the data of the curve will be automatically written to the board end to take effect.

### 3.1.4.3 VO Gamma

Click VO Gamma in the left list of tuning tool to enter the VO Gamma visual tuning page, as shown in Fig. 3.8.

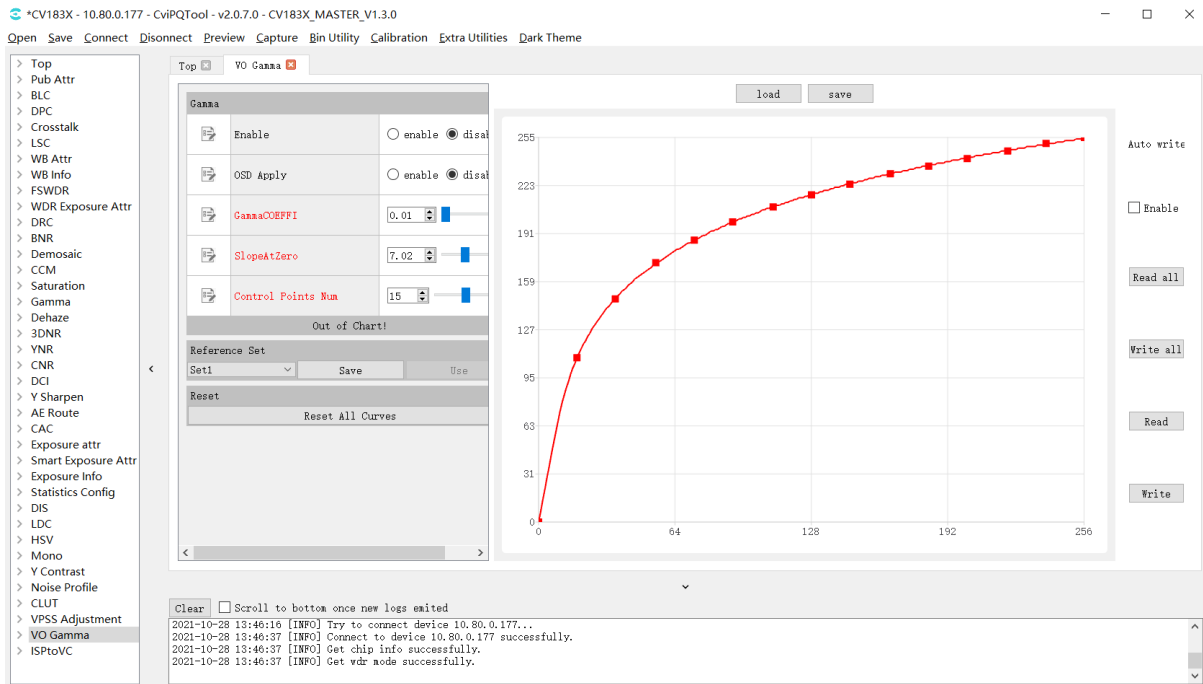


Fig. 3.8: VO Gamma tuning page

#### Parameters description:

- Gamma
  - Enable, set it to enable VO Gamma
    - OSD Apply, display watermark switch.
    - GammaCOEFFI parameter, restore the current curve to a standard gamma curve (coefficient range 0.01 to 20.0) by entering a value in the text box or dragging the slider.
    - SlopeAtZero parameter, adjust the slope of gamma curve at zero point (the slope range is 0.01 to 20.0) by inputting a value in the text box or dragging the slider.
    - Control Points Num, when GAMMA\_USER\_DEFINE/GAMMA\_AUTO is selected, by adjusting this parameter, the user can set the number of control points that can be operated.

- Reference Set

Curve data temporary storage area, the tool provides three sets of temporary storage space from set1 to set3, the user clicks the save button the current curve data will be saved to the selected set, and the user clicks the use button the temporary stored data can be restored to the curve interface. By default, the use button is grayed out and unclickable when no temporary storage operation is performed.

- Reset

Reset is the reset button. After clicking, the curve will be reset to a straight line with the starting point of (0,0) and the ending point of the maximum value of X / Y axis.

- Load&Sav

- Click the “Load” button to read the curve data from the file, write it back to the tool data cache, and refresh the display curve at the same time.
- Click the “Save” button to save the current curve data to the local text file. The tool supports two formats of data files: Cvitek Json file and Other text file
- Click the “Read” button of the page read / write control interface, the tool will read the curve data from the board end and refresh the curve display.
- Click the “Write button” , and the tool will write the data of the current curve to the board end.
- If “Auto Write” is checked, when the curve is adjusted, the data of the curve will be automatically written to the board end to take effect.

Usage:

1. Tool is connected to the board end normally, and the board end is connected to the display to output images normally.
2. Use the “GammaCOEFFI” , “SlopeAtZero” , “Control Points Num” functions to adjust a desired curve.
3. Select enable for “Enable” and click write.
4. Observe the display and the image brightness will change.

## 3.1.5 Exposure Data Status and Parameter Setting

### 3.1.5.1 Exposure Data Status

The Exposure Info page can display the current exposure data status, including exposure time, gain, ISO, etc. As shown in Fig. 3.9.

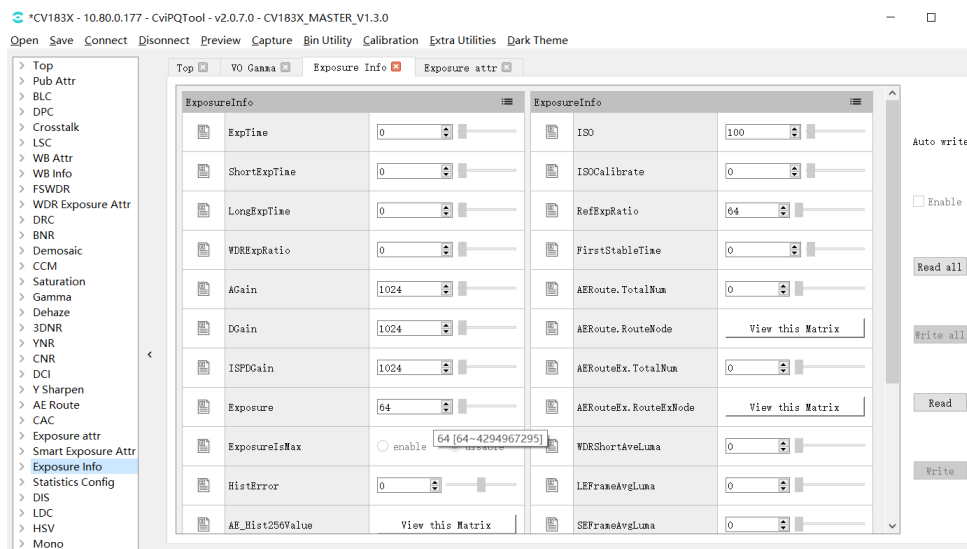


Fig. 3.9: Exposure data status page



### 3.1.5.2 Exposure Parameter Setting

The Exposure Attr page can set the exposure related parameters, including exposure time, gain, ISO, etc. As shown in Fig. 3.10.

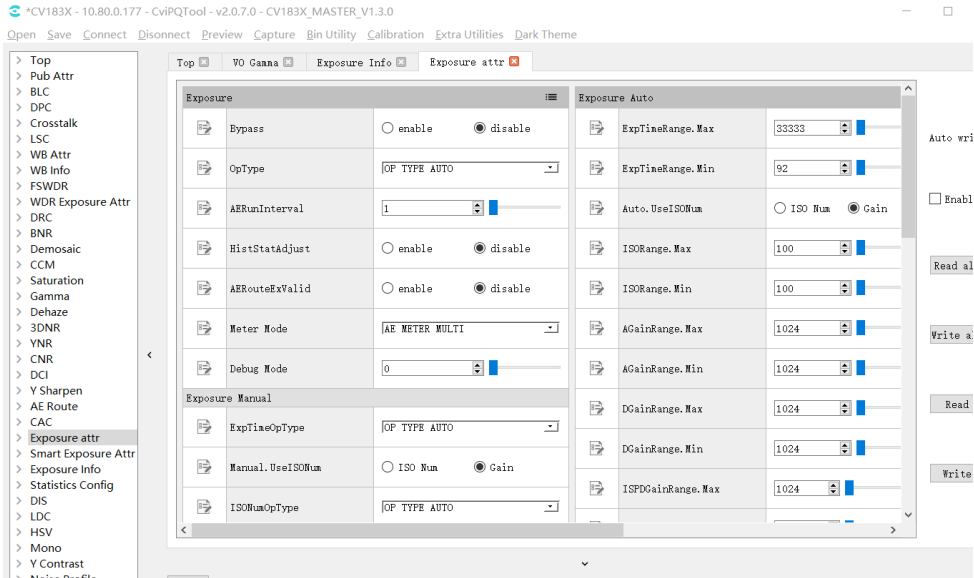


Fig. 3.10: Exposure parameter setting page

## 3.1.6 White Balance Data Status and Parameter Setting

### 3.1.6.1 White Balance Data Status

WB info page can display the current white balance data status, including gain, color temperature, etc. As shown in Fig. 3.11.

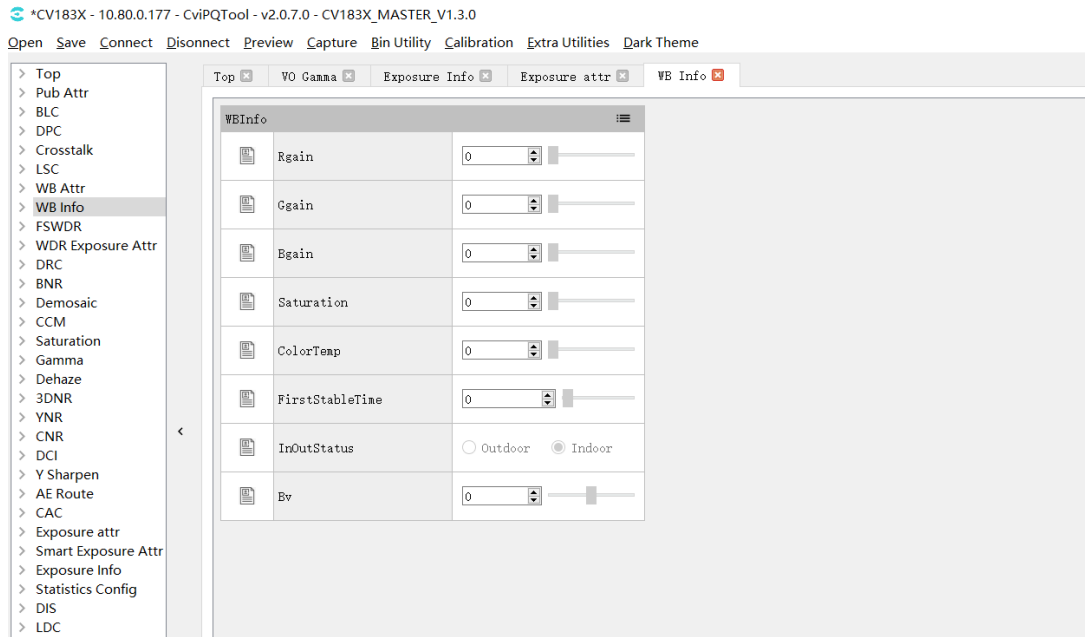


Fig. 3.11: White balance data status page

### 3.1.6.2 White Balance Parameter Setting

WB attr page can be set white balance related parameters, including gain, color temperature, etc. As shown in Fig. 3.12.

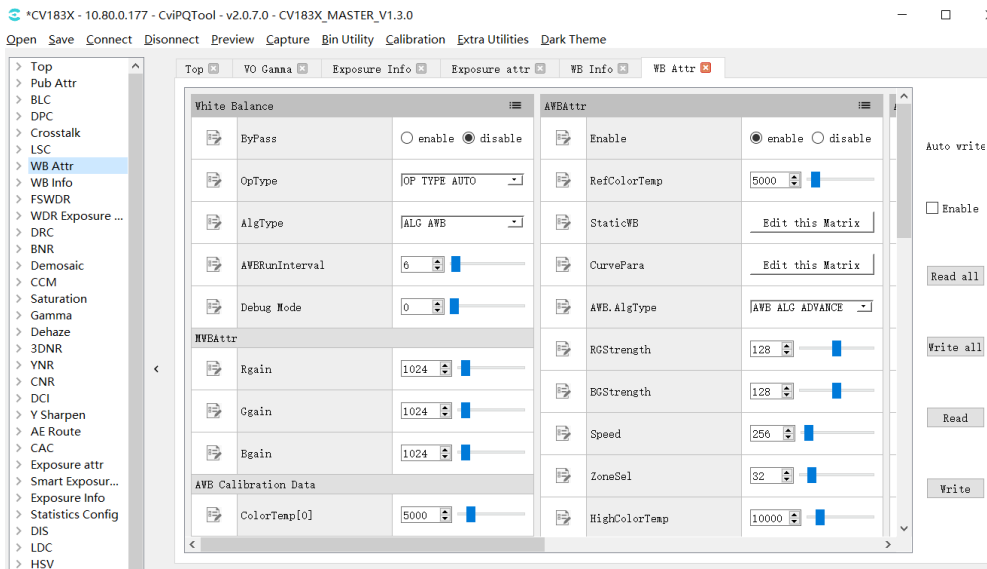


Fig. 3.12: White balance parameter setting page

## 3.1.7 WDR Exposure Parameter Setting and Local-Tone Switch

For detailed description of WDR and DRC parameters, please refer to section WDR and section DRC from “ISP Tuning Guide” .

### 3.1.7.1 WDR Exposure Parameter Setting

WDR Exposure Attr page can be set wide dynamic exposure related parameters, including exposure ratio, etc. As shown in Fig. 3.13.

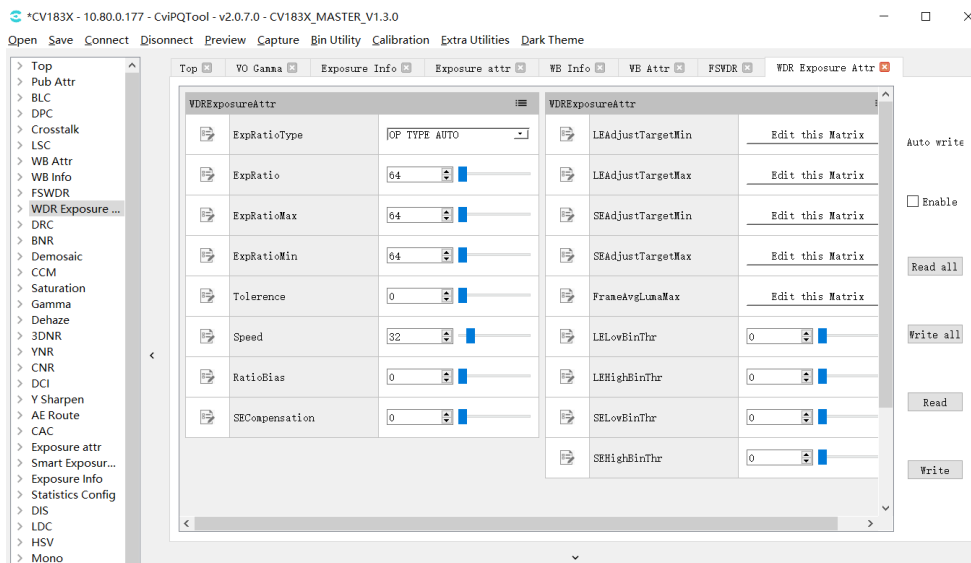


Fig. 3.13: WDR exposure parameter setting page

### 3.1.7.2 Local-tone Switch

Local-tone can be turned on or off on the DRC page. When the [LocalToneEn] parameter is set to enable, it means that the Local-tone is turned on. When the [LocalToneEn] parameter is set to disable, it means that the Local-tone is turned off, as shown in Fig. 3.14.

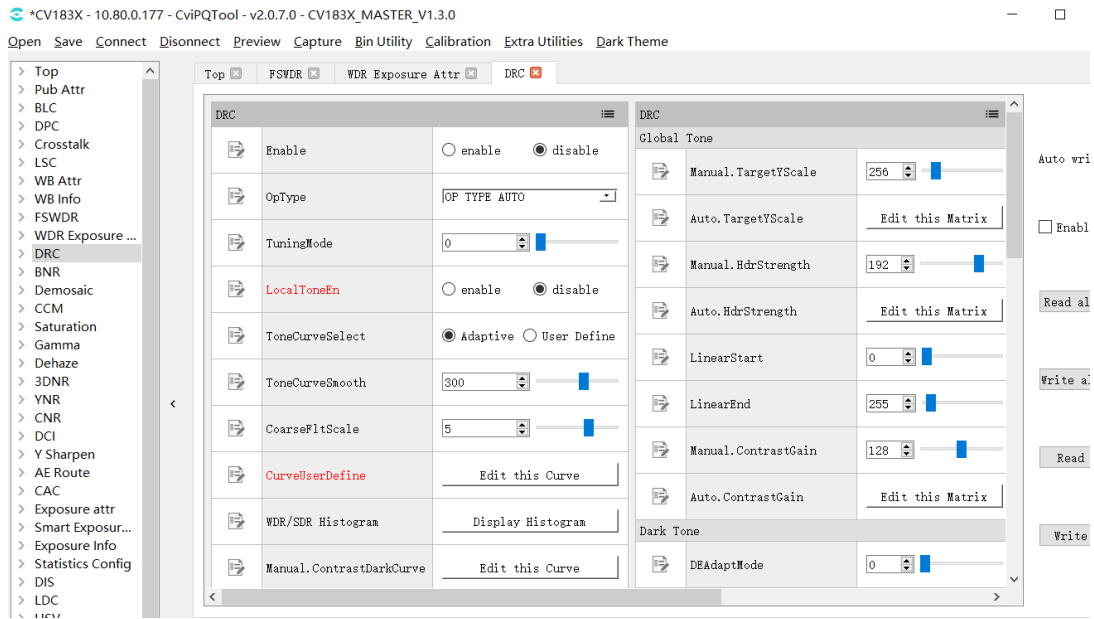


Fig. 3.14: DRC Local-tone switch

### 3.1.8 3DNR and YNR Noise Reduction Tuning

The tuning parameters related to noise reduction can be found in 3DNR and YNR pages. For detailed description of noise reduction parameters, please refer to section 3DNR and section YNR in “ISP Tuning Guide” .

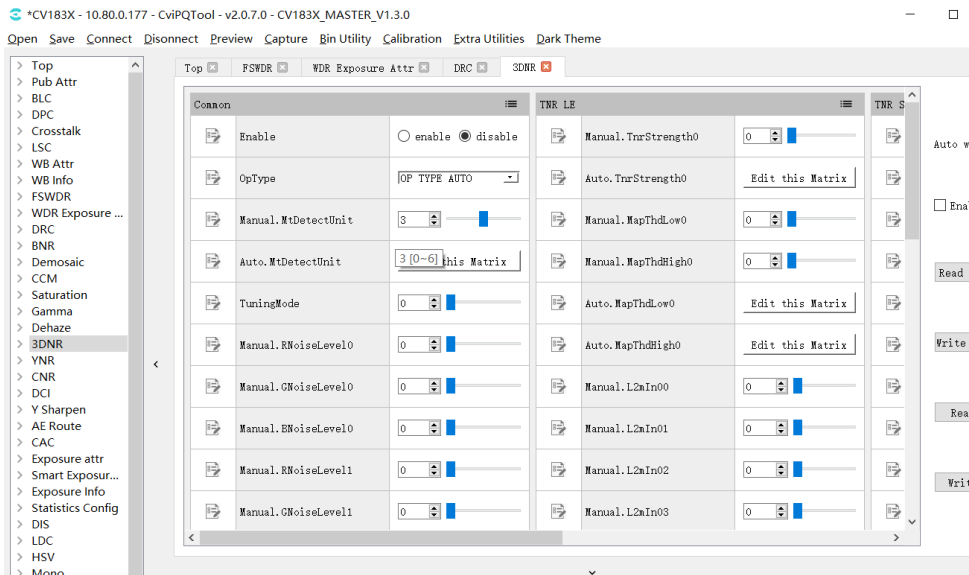


Fig. 3.15: 3DNR page

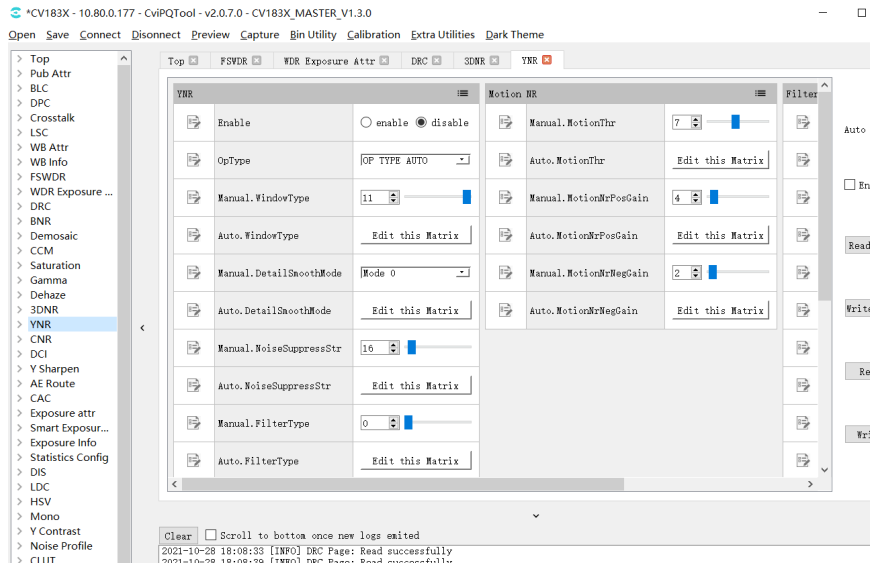


Fig. 3.16: YNR page

### 3.1.9 Y Sharpen Sharpening and Edge Enhancement Tuning

Y Sharpen page contains parameters related to sharpening and edge enhancement. Please refer to section Sharpen of “ISP Tuning Guide” for detailed description of sharpening and edge enhancement parameters.

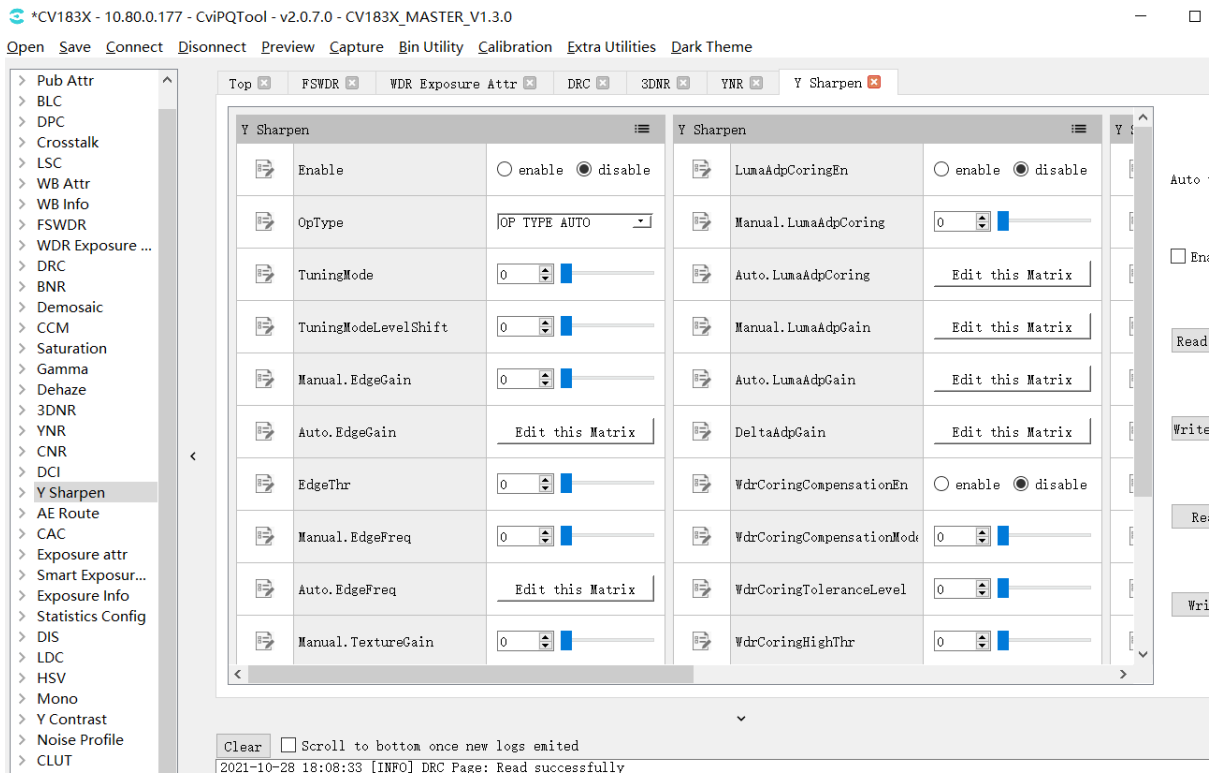


Fig. 3.17: Y Sharpen page

### 3.1.10 Top Page

The Top page includes ViPipe and ViChn parameters. If there are two sensors on the board end, Vipipe can be used to switch the tuning parameters of different sensors. In addition, Vichn can be used to switch the image of the sensor to be displayed on the screen.

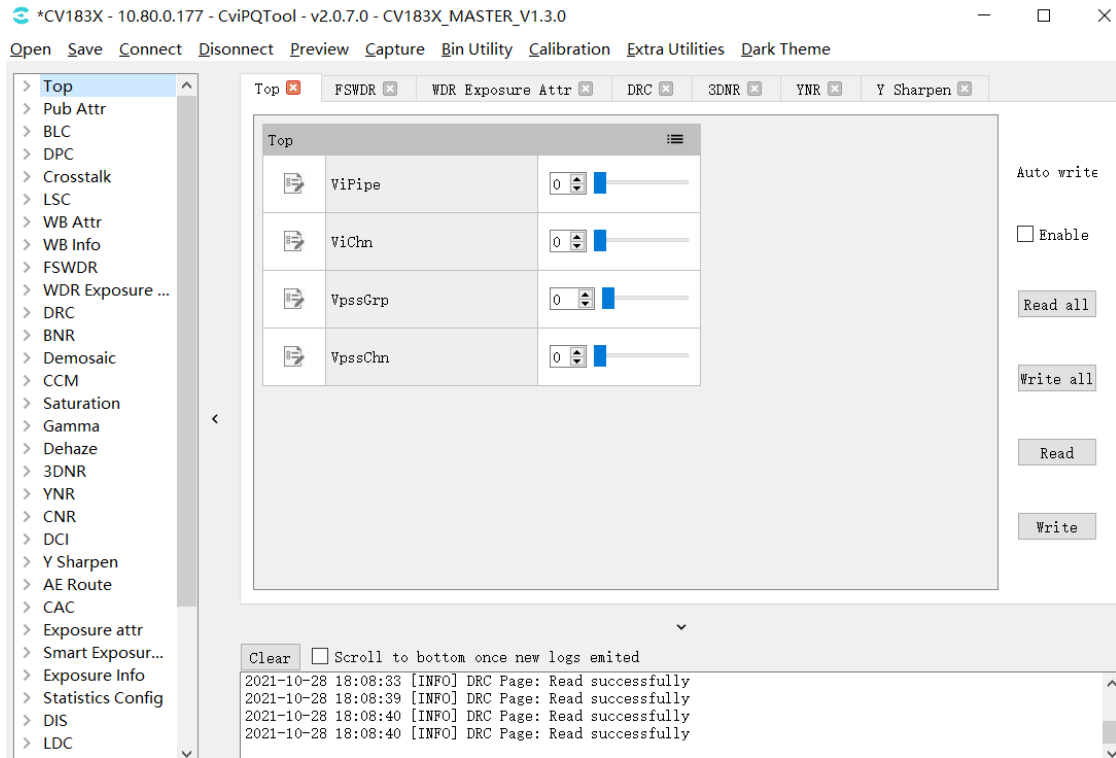


Fig. 3.18: Top page

### 3.1.11 VPSS Adjustment Page

The VPSS Adjustment page is used to tune vpss-related parameters, and the following points should be noted in the tuning process.

1. During tuning with pqtool, the scene id and vpss grp id are one-to-one, that is, the parameters of scene0 will be written to grp 0;
2. When App is actually used, it should call CVI\_VPSS\_SetGrpParamfromBin after each load pq bin to set the binding relationship between scene and vpss grp id, and the vpss parameters in pq bin will take effect at the same time. Otherwise, the vpss parameters tuned with pqtool will not take effect after load pq bin.

Note: the reason for this design is that it is not possible to predict which vpss grp id the app will actually use during tuning of the pq parameters, so the tuning is done one by one, and the mapping relationship needs to be specified according to the actual scenario when the app uses the pq bin generated by tuning.

### 3.1.12 LDC Page

The LDC page is used for tuning VI and VPSS LDC module related parameters, and the following points should be noted in the usage process:

1. Read all and Write all do not include read/write operations on the LDC page. Therefore, if the LDC page is set to read or write, a separate operation (click the Read or Write button) must be performed on this page to make it effective.
2. Currently, the LDC operation is performed at the board end and its computation is large and time consuming. It takes about 2 minutes to complete a computation operation and the PQtool needs to wait in the meantime before the next effective operation can be performed on the PQtool.

## 3.2 ISP Calibration Tool

Please refer to “ISP Tuning Guide” to complete ISP calibration.

## 3.3 Advanced Functions

### 3.3.1 Export and Import of Parameters

The parameters of the tool can be imported and exported in the form of configuration files on the PC side, and the tool can also solidify the parameters to the board side.

- Please refer to *section 2.4.3.1 Save Tuning Data Files* and *section 2.4.3.2 Open Tuning Data File* for the operation of importing and exporting parameters to configuration file on PC.
- To solidify the parameters to the board end, or to export and back up the parameters from the board end to the PC local file, please use the BIN utility tool for operation.

### 3.3.2 Communication Log

After connecting with the board, all data interaction with the board will be recorded and displayed in the communication log window, as shown in [Fig. 3.19](#) . When interacting with the board, the log contains the following information:

- Communication time;
- Communication mode and parameters;
- Communication contents (such as tuning items currently read or written);
- If communication error occurs, error message will be displayed.

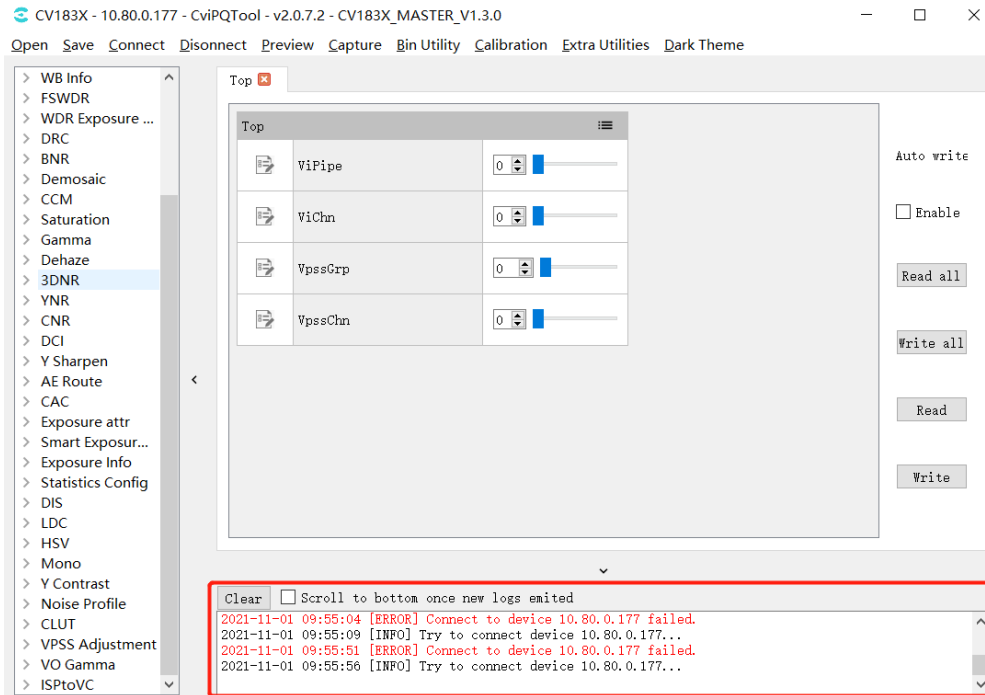


Fig. 3.19: Commuication log window

## 3.4 Instructions for Other Auxiliary Tools

### 3.4.1 Instructions for Use of Capture Tool

The image data (YUV and Raw) at the board end can be captured and saved as a local file.

#### 3.4.1.1 Tool Interface

Click the Capture button on the tuning tool menu bar to open the capture tool interface, as shown in Fig. 3.20.



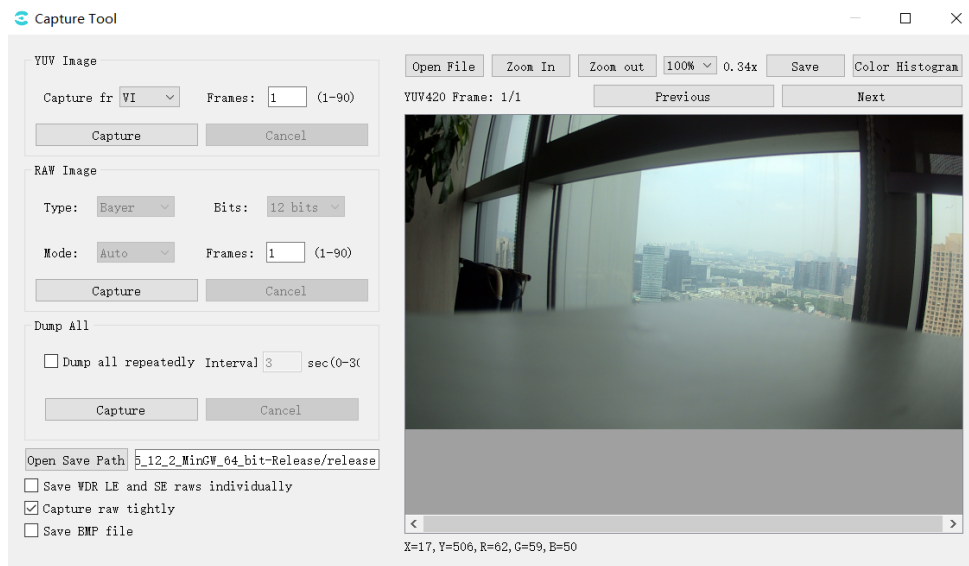


Fig. 3.20: Capture tool interface

Save WDR LE and SE raws individually: when checked, LE raw and SE raw are stored separately; when unchecked, the two raw are stored together.

Capture raw tightly: the way of capturing raw is different, faster when it is checked and slower when it is unchecked.

Save BMP file: when checked a bmp image will be saved, the default is saved in jpg.

At the bottom of the image window, the x,y coordinates and r,g,b values of the point are displayed and change with mouse movement.

Dump all repeatedly: when checked, the dump all will be repeated at the Interval setting time (s).

### 3.4.1.2 Capture YUV Image Data

Steps to capture YUV data:

1. In the “YUV image” group box, select the source location of YUV data to capture, that is, select VI / VPSS from the Capture from drop-down menu;
2. Input the number of frames to capture in the Frames text input box;
3. Click the Capture button, if the capture is successful, YUV data will be automatically stored in the path position displayed in “Open Save Path” , and YUV data will be displayed in the right image display area, which can be viewed frame by frame by clicking Previous / Next button;

—Done

### 3.4.1.3 Capture Raw Image Data










Steps to capture Raw data:

1. Select the bits to capture Raw data in the “Raw Image” group box, that is, select 8bits / 10bits / 12bits / 16bits in the Bits drop-down menu;
2. Select the current mode as Linear, WDR or Auto from the Mode drop-down menu;
3. Input the number of frames to capture in the Frames text input box;
4. Click the “Capture” button, if the capture is successful, Raw and Raw Info data will be automatically stored in the path position displayed in “Open Save Path”, and Raw data will be displayed in the image display area on the right, which can be viewed frame by frame by clicking “Previous / Next” button.

—Done

### 3.4.1.4 Dump All

Dump All can capture YUV, RAW images and log information, and capture files as follows:


-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_.jpg
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_.json
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_.raw
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_.txt
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_.yuv
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_-ae.bin
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_-aelog.txt
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_-awb.wbin
-  3840X1080\_RGGB\_WDR\_20211101111620\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=1\_ISO=100\_-awblog.txt

Capture steps:

1. Set up the yuv capture information in the YUV Image.
2. Set up the raw capture information in the RAW Image.
3. Click the Capture button in Dump All, the tool will first capture and save the raw image and file, then capture and save the yuv image and file, the right image display area will show the raw image, which can be viewed frame by frame by clicking the Previous/Next button.

### 3.4.1.5 File Name Format Description of YUV Image and RAW Image

The following is the example of file name format of YUV image :

 1920X1080\_RGGB\_Linear\_20211101105807\_00\_-color=3\_-bits=12\_-frame=1\_-hdr=0\_ISO=110\_.yuv

- 1920X1080 means 1920 in width and 1080 in height
- color=3 means BayerID=3 (0: BGGR, 1: GBRG, 2: GRBG, 3: RGGB)

- bits=12 indicates that the significant bits of the image are 12 bits
- frame=1 means that this file contains one YUV image
- 20211101105807 indicates the save time in the format of "yyyymmddhhmmss"

The following is the example of file name of RAW image:

```
3840X1080_RGGB_WDR_20211101111620_00_-color=3_-bits=12_-frame=1_-hdr=1_ISO=100_.raw
```

- 3840X1080 means 3840 in width and 1080 in height
- RGGB means BayerID
- WDR is represented as a wide dynamic image
- color=3 means BayerID=3 (0: BGGR, 1: GBRG, 2: GRBG, 3: RGGB)
- bits=12 indicates that the effective bits of the image are 12 bits
- frame=1 means that this file contains one RAW image
- hdr=1 represents wide dynamic image
- 20211101111620 indicates the save time in the format of "yyyymmddhhmmss"

**The Raw here is decompressed and cropped to: size = width \* height \* 2 \* frame;**

#### 3.4.1.6 File Format and Content Description of Raw Info File

An example of the RAW Info file name format is as follows:

```
3840X1080_RGGB_WDR_20211101111620_00_-color=3_-bits=12_-frame=1_-hdr=1_ISO=100_.txt
```

The file name format of RAW Info is exactly the same as that of RAW image, except for the text file with the extension of TXT.

The following is an example of RAW Info file content:

```
ISO = 100
Color Temp. = 4482
ISP DGain = 1024
Long Exposure = 12630
Short Exposure = 3157
Exposure Ratio = 4
reg_wbg_rgain = 1834
reg_wbg_bgain = 2166
reg_ccm_00 = 1736
reg_ccm_01 = -589
reg_ccm_02 = -123
reg_ccm_10 = -649
reg_ccm_11 = 2016
reg_ccm_12 = -343
reg_ccm_20 = -32
reg_ccm_21 = -742
reg_ccm_22 = 1798
reg_blc_offset_r = 241
reg_blc_offset_gr = 241
reg_blc_offset_gb = 241
reg_blc_offset_b = 241
reg_blc_gain_r = 1089
reg_blc_gain_gr = 1089
reg_blc_gain_gb = 1089
reg_blc_gain_b = 1089
```

## 3.4.2 Video Player Instructions (VLC)

### 3.4.2.1 Steps to use VLC to play video stream on board

**Step 1:** Download the VLC player. You can download the install-free windows VLC player from the following address : (<https://get.videolan.org/vlc/3.0.10/win32/vlc-3.0.10-win32.zip>)

For other versions of VLC for other platforms can be downloaded from the VLC website.

**Step 2:** Open the tool, set VLC parameters, and save and restart VLC.



Fig. 3.21: VLC setting interface

**Step 3:** Use network cable to connect development board and computer. The development board and computer can be directly connected by network cable or connected by router.



Fig. 3.22: Schematic diagram of network port connection

**Step 4:** On the board, enter the file system through the serial port and start the CviISPTool program

```
[ 4.148202] end jpu_init result = 0x0
[ 4.539741] cvi_vc_drv_init result = 0x0
[ 4.895003] sh (130): drop_caches: 3
Starting app...
Starting crond: OK

login[281]: root login on 'console'
[root@cvitek]~# ifconfig eth0 192.168.1.177
[root@cvitek]~# cd /
[root@cvitek]/# mount /dev/mmcblk0p1 /mnt/sd/
cd /mnt/sd/
[root@cvitek]/# cd /mnt/sd/
[root@cvitek]/mnt/sd# cd install/
./CviIsptool.sh 128M
[root@cvitek]/mnt/sd/install# ./CviIsptool.sh 128M
MMF Version:3f8eba872-musl_riscv64
[6]level = 3
[6]isEnabledSetPQBin = 0
[6]SDR_PQBinName = /mnt/data/bin/cvi_sdr_bin
[6]WDR_PQBinName = /mnt/data/bin/cvi_wdr_bin
```

The computer connected with development board needs to be set to the same IP network segment and after mutual ping, VLC can be started to play on the computer connected with development board. The following are some parameters of play setting. After setting, click the play button to see the Sensor screen.

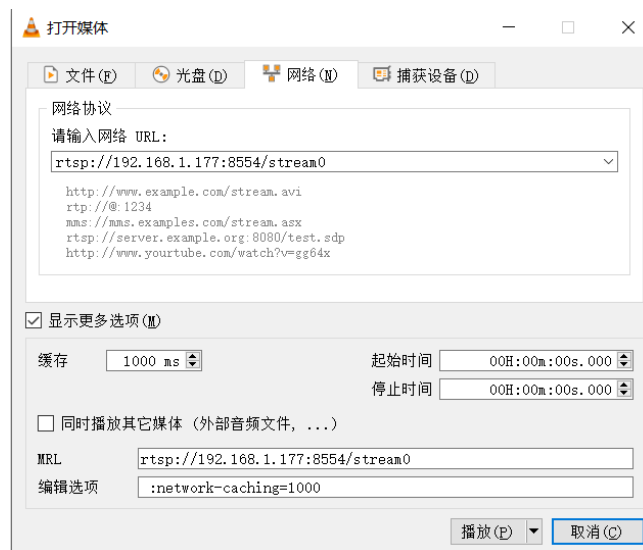


Fig. 3.23: VLC playback parameters setting

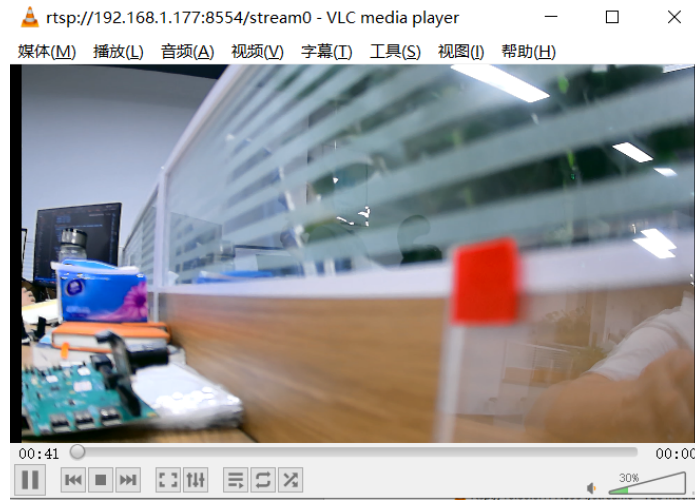
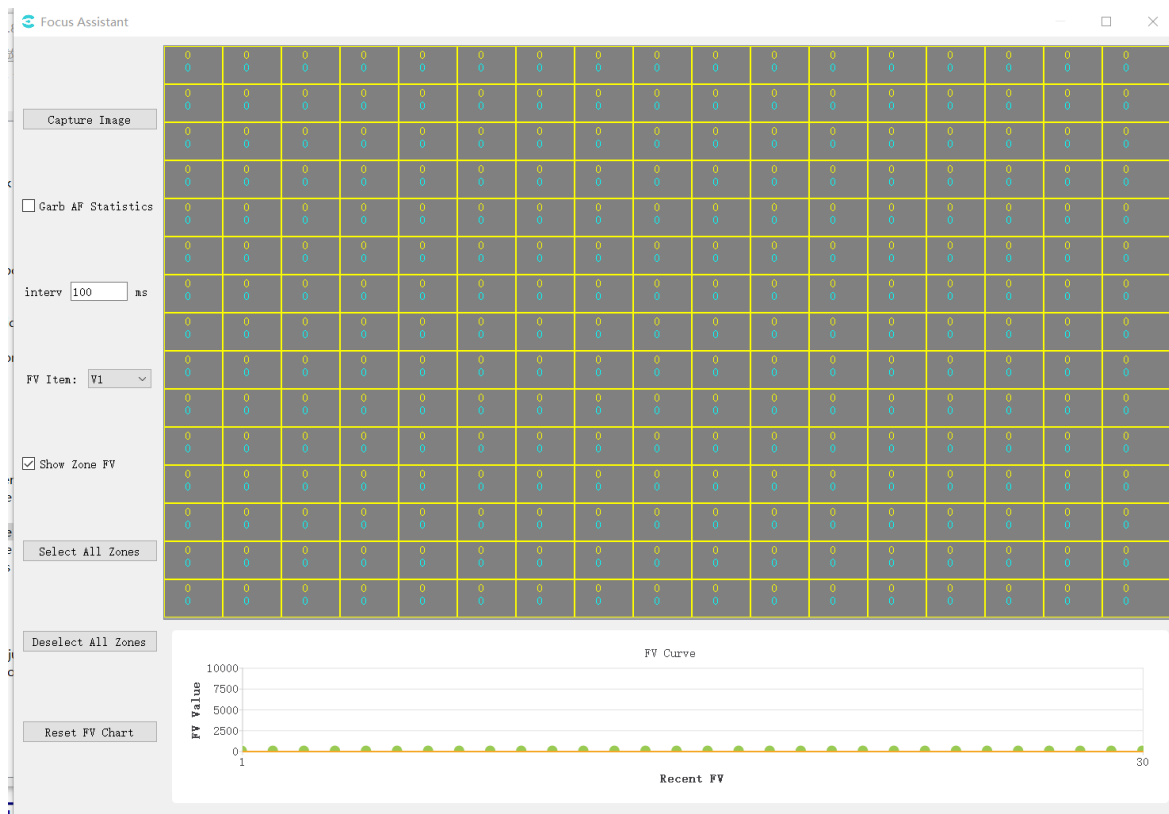


Fig. 3.24: VLC playing video stream

### 3.4.3 Instructions for Auxiliary Focusing Tool

#### 3.4.3.1 Tool Interface

On the tool home page, select Extra Utilities->Focus Assistant, and the auxiliary focusing interface is shown in the figure below:



The tool interface is mainly divided into three parts:



(1)Operation area: for various tool operations

(2)Image and interval display area: display the captured image, interval distribution and FV value of each interval (the Yellow value is the current FV value of the interval, and the blue value is the maximum historical FV value of the interval)

(3)FV change curve: display the change curve of global FV value. The point whose abscissa is 1 is the point represented by the nearest FV value. Each time FV value is refreshed, the previous value will be shifted to the right.

### 3.4.3.2 Use the Auxiliary Focusing Tool to Focus

Steps to focus with the auxiliary focusing tool are as follows:

Step1. Adjust the focal length of the lens to the most blurred state

Step2. Determine the AF interval configuration status of the current device: check grab AF statistics, and remove the check when the number of intervals in the image display area changes.

Step3. Select FV group: select V1 (vertical direction) and H1 (horizontal direction) in FV Item drop-down box.

Step4. Click in the image and interval display area to determine which intervals to focus on. A green circle will be displayed in the upper left corner of the region of interest. You can click “Select All Zones” on the left side of the tool to select all zones, and “Deselect All Zones” to cancel the selection of all zones for easy operation.

Step5. Check “Grab AF Statistics” again. At this time, the image display area and the lower graph begin to refresh continuously.

Step6. Adjust the focal length of the lens. At this time, the blue line of FV curve maximum value increases steadily.

Step7. When observed that the point of position 1 on the FV value curve begins to decline steadily, the position of the maximum blue line should not change (it can be considered that the blue line FV is the theoretical optimal focusing FV). At this time, focus in the opposite direction until the point on position 1 is infinitely close to the blue line position.

### 3.4.3.3 Focusing Auxiliary Operation

If the user needs to determine which region of the on-demand image belongs to in the focusing process, click “CaptureImage” to capture an image. After the image is displayed in the image area, it can be confirmed.

When “Grab AF statistics” is checked on the tool, the tool will obtain AF statistics from the board end at a fixed time interval. The user can modify the Interval value on the interface to change the interval. The change range is 100-1000 ms.

If the user does not need to view the current value and peak value of FV value between partitions, the show zone FV check box on the left can be unchecked. Remove the check and check again, the interval FV value will be displayed again.

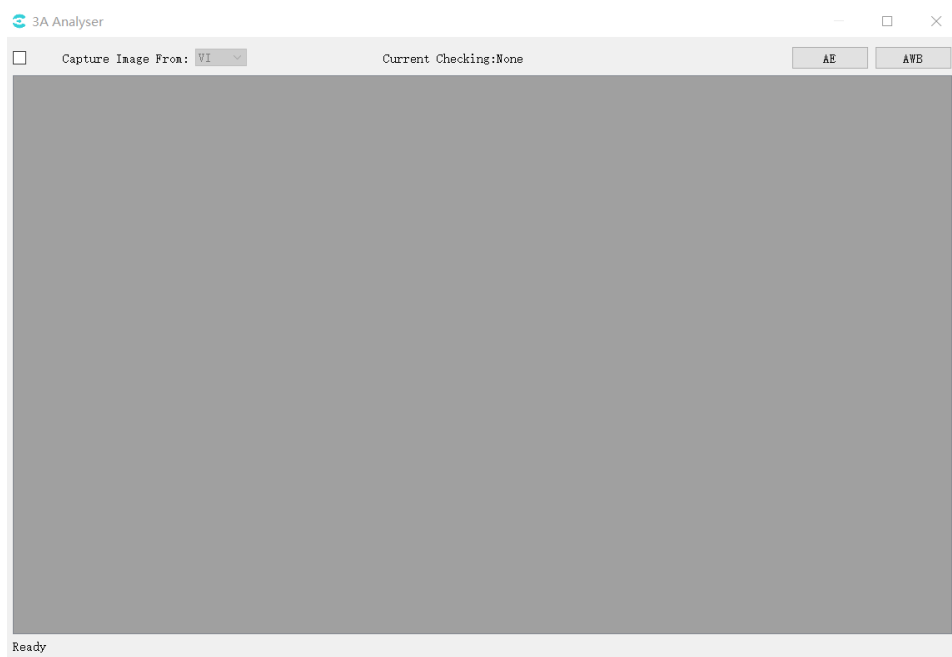
If the user changes the interval configuration, the theoretical FV maximum will change. Users are advised to click the “Reset FV Chart” button at this time. After clicking, the historical

global FV value in the lower right chart will be cleared, and the blue line of the maximum FV value will return to zero.

## 3.4.4 3A Analyser Tool

### 3.4.4.1 Tool Interface

In the tool home page, select Extra Utilities -> 3A Analyser in the toolbar, the 3A Analyser interface is as follows:

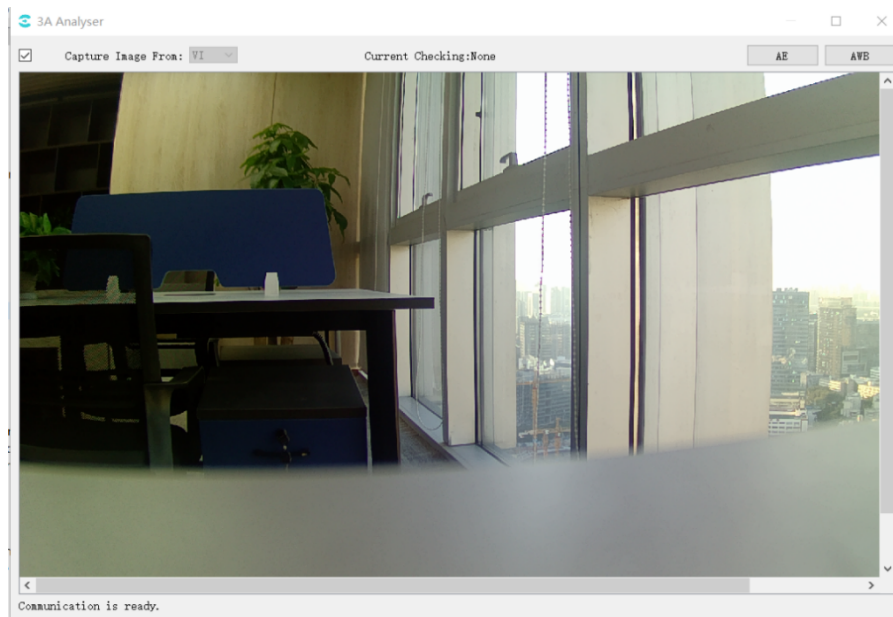


### 3.4.4.2 Obtain Images and Statistics

1. CviPQTool connects to the board side.
2. On the main interface of the 3A Analysis tool, in the drop-down box after Capture Image from, select the image data source (currently the tool only supports acquiring image data from VI).
3. Check the checkbox before Capture Image from. The tool will then automatically capture the data from the board side until the user unchecks the Capture Image from checkbox.

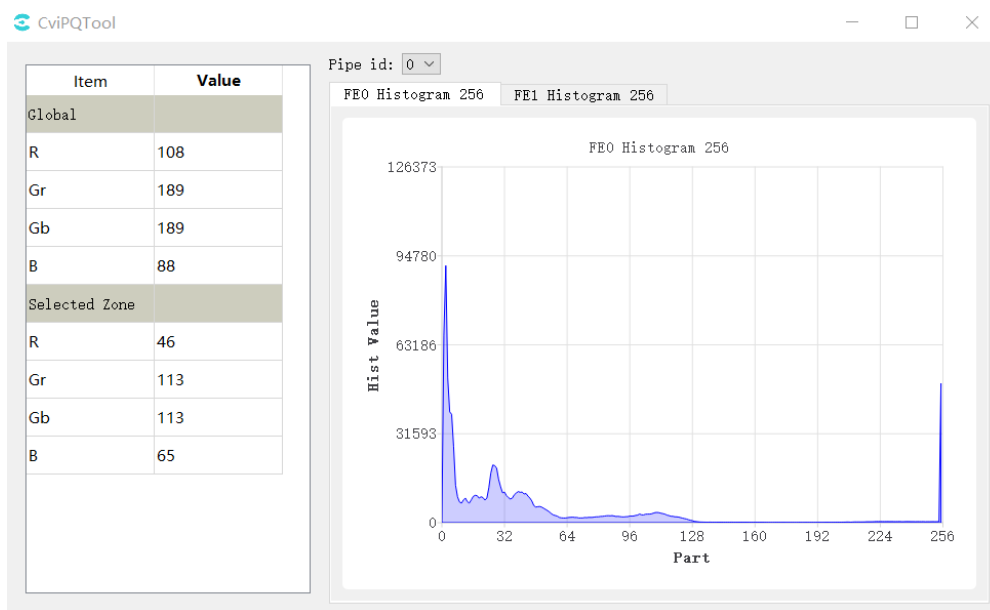
After successfully capturing data from the board side, the tool will display the image effect in the gray area below as shown in the figure below:



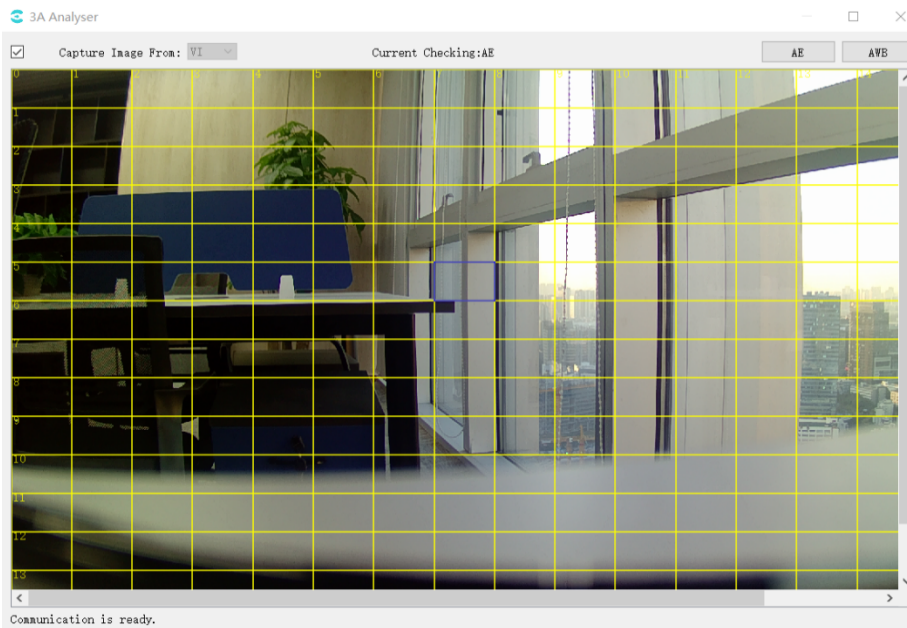


### 3.4.4.3 View AE Statistics

Click the AE button in the upper right corner of the 3A Analyser tool interface to open the AE statistics window as shown below:



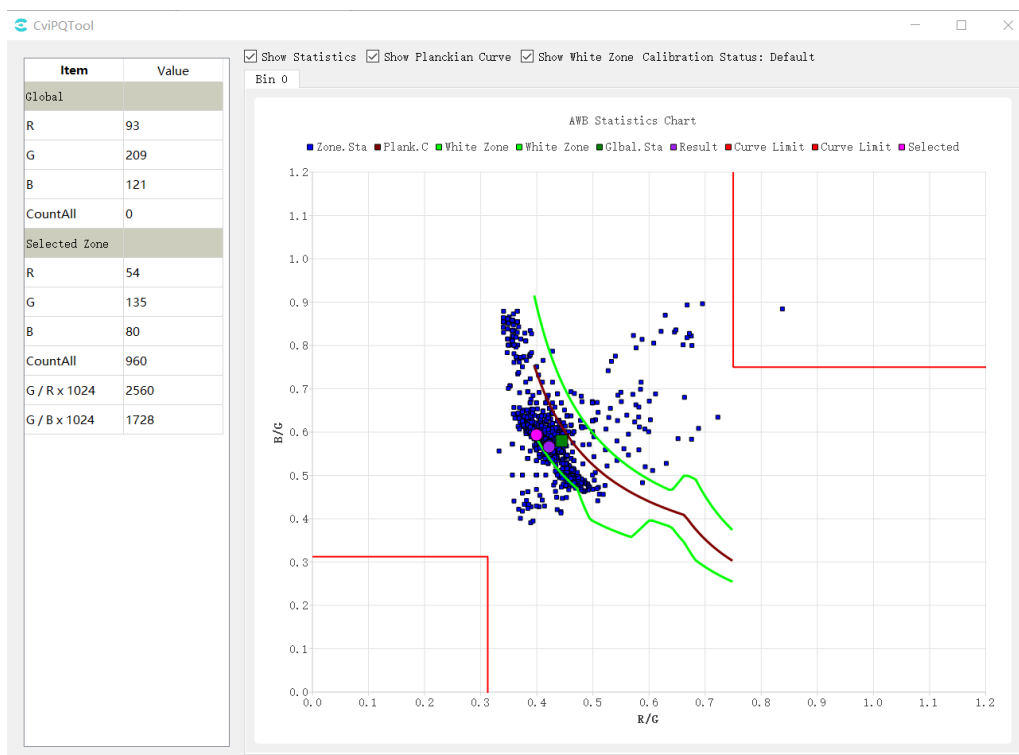
At the same time, the image area of the main interface of 3A Analyser will be divided into intervals by yellow lines, and users can use the mouse The user can use the left mouse button to select the interval, as shown in the figure below.



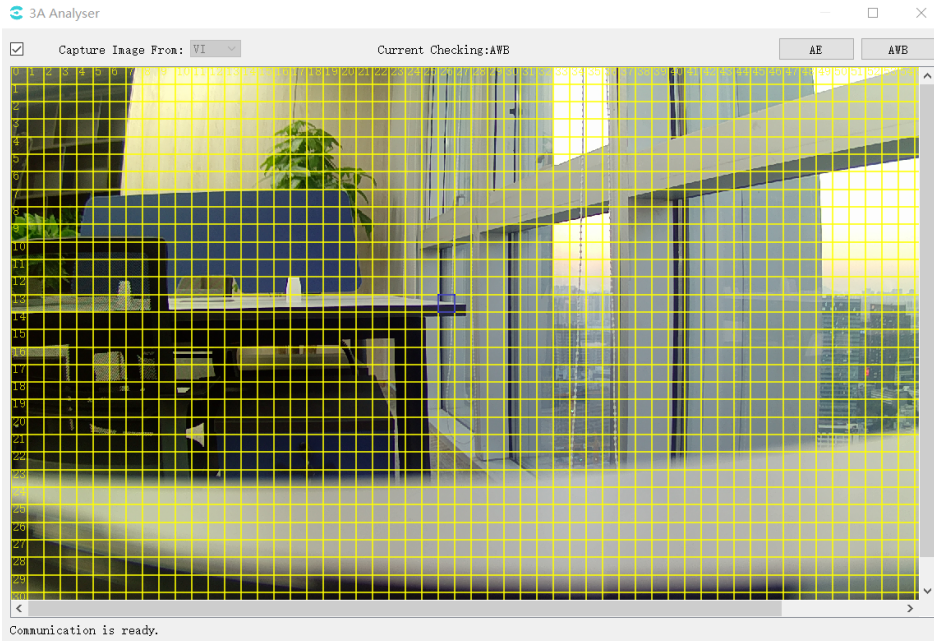
Statistical information: The average value of the components of Global and Selected Zone can be viewed. When switching the selection zone in the main screen of 3A Analyser, the statistical information data of the selected zone will change.

### 3.4.4.4 View AWB Statistics

Click the AWB button in the upper right corner of the 3A Analyser interface to open the AWB statistics window as shown below.



At the same time, the image area of the main interface of 3A Analyser is also divided into intervals by yellow lines. The user can use the left mouse button to select the interval, as shown in the following figure.



The following are shown on the statistical chart.

Red-brown curve: Planck curve (plank.c).

Green point: the point on the Planck coordinate system corresponding to the global statistics.

Purple point: the point corresponding to the result of AWB calculation on the Planck coordinate system.

Blue point: the point on the Planck coordinate system corresponding to the statistical information of the subinterval.

Red point: the point on the Planck coordinate system corresponding to the selected zone of the image selected by the mouse.

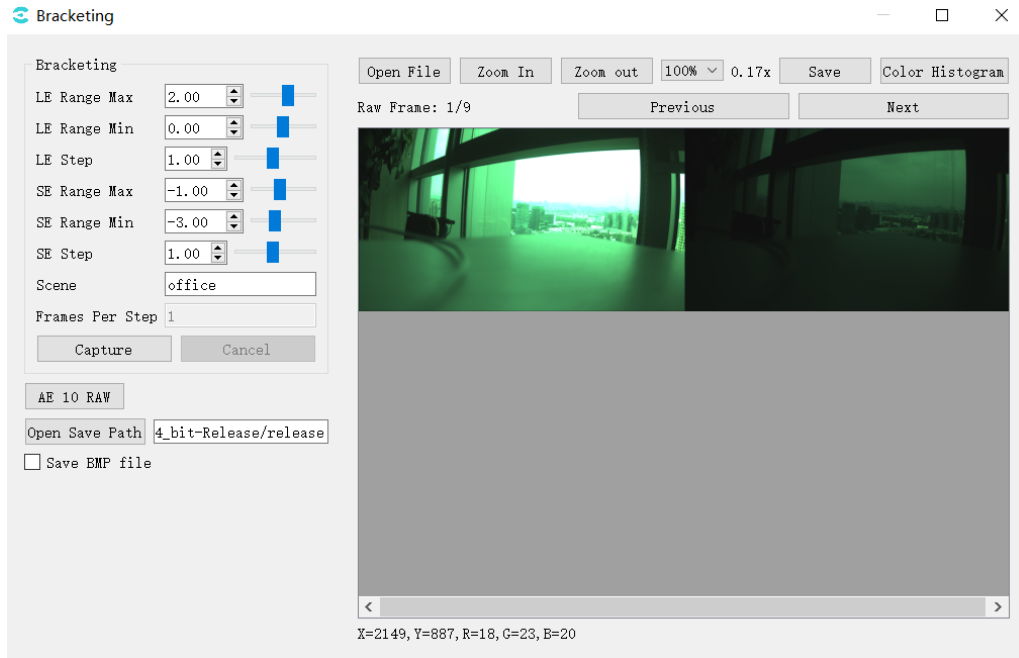
Green curve: base white zone interval.

The area enclosed by the red lines at the top right and bottom left: the intervals excluded from the white zone to exclude purple and green interference, respectively, are related to CurveLLimit and CurveRLimit in the AWB parameters.

Show Statistics, ShowPlanckian Curve, and ShowWhite Zone above the image control the display and hiding of statistics, color temperature curves, and white zone intervals on the coordinate map, respectively.

### 3.4.5 Bracketing

Select Extra Utilities -> Bracketing in the toolbar and the Bracketing window is shown below.



LE/SE Max: Set long and short exposure maximum

LE/SE Min: Set long and short exposure minimum

LE/SE Step: Step length

Capture: Capture the raw of different exposure combinations set by LE/SE

AE 10 RAW: Capture 10 raw of a specific exposure

Save BMP file: Save BMP images, default save as JPG

Open Save Path: File saving directory

Calculation of the number of captured pictures.

$$\text{LeNum} = (\text{LeMax} - \text{LeMin}) / \text{LeStep} + 1;$$

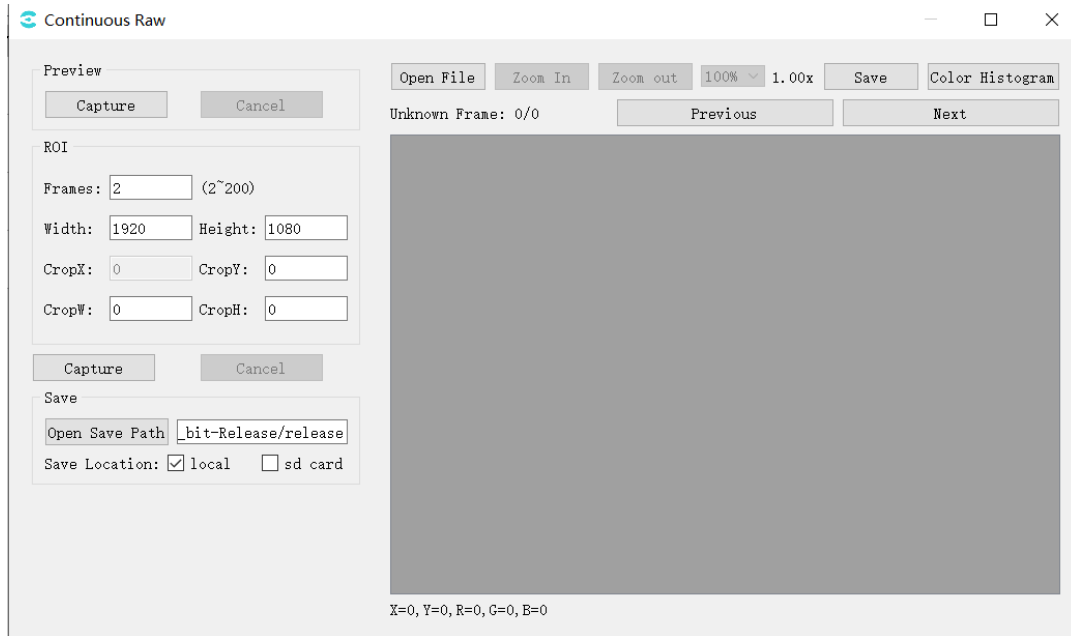
$$\text{SeNum} = (\text{SeMax} - \text{SeMin}) / \text{SeStep} + 1;$$

$$\text{wdr: TotalNum} = \text{LeNum} * \text{SeNum};$$

$$\text{linear: TotalNum} = \text{LeNum}$$

### 3.4.6 Continuous Raw Tool

Select Extra Utilities -> Continuous Raw in the toolbar and the Continuous Raw tool is shown below .



Preview: Preview the current screen

Capture: Click on “capture” to capture the yuv and display it in the right window;

ROI: Capture raw settings

Frames: Capture raw number of frames;

Width/Height: The width and height of the current screen;

CropX/CropY/CropW/CropH: Set the size and position of capturing raw;

Capture: Start capturing raw;

Save: Save settings

Open Save Path: Select the save path;

Save Location: Select save location, select local, raw file will be saved on pc side; select sd card, raw file will be saved on board side;

The capturing files are shown [Fig. 3.25](#) :

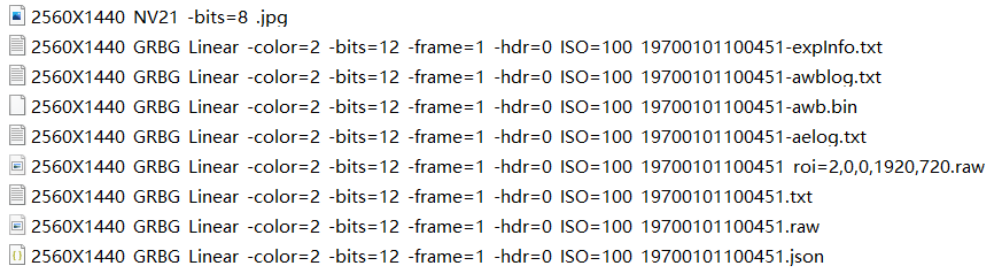


Fig. 3.25: The capturing files

The raw image here is compressed and the size is calculated as follows:

1. Full size raw

$$\text{Size} = (\text{width} * 6 / 8 + 15) / 16 * \text{height} * \text{frame};$$

**If Width > 2304, the compression mode is tile, Width needs to be added by 8.**

2. roi raw

$$\text{Size} = (\text{CropW} * 6 / 8 + 15) / 16 * \text{CropH} * \text{frame};$$

**If Width > 2304, CropW > 1536 and the compression mode is tile, CropW needs to be added by 8.**

## 3.5 Detail Description of Parameters

The following is the description of API function reference of SDK corresponding to each module parameter on the tool interface, as shown in Table 3-2~ Table3-42.

Table 3.2: API corresponding to TOP settings

Function Module	Corresponding SDK API
Mode Handle	N/A
Bypass Setting	CVI_ISP_SetModuleControl CVI_ISP_GetModuleControl
Fmw State	CVI_ISP_SetFMWState CVI_ISP_GetFMWState
CTRL Param	CVI_ISP_SetCtrlParam CVI_ISP_GetCtrlParam
Repeat	CVI_ISP_SetPipeRepeatMode CVI_ISP_GetPipeRepeatMode
MOD Param	CVI_ISP_SetModParam CVI_ISP_GetModParam

Table 3.3: API corresponding to PubAttr

Function Module	Corresponding SDK API
PubAttr	CVI_ISP_SetPubAttr CVI_ISP_GetPubAttr

Table 3.4: API corresponding to ISPInfo

Function Module	Corresponding SDK API
ISPInfo	CVI_ISP_QueryInnerStateInfo

Table 3.5: API corresponding to ExposureAttr

Function Module	Corresponding SDK API
ExposureAttr	CVI_ISP_SetExposureAttr CVI_ISP_GetExposureAttr

Table 3.6: API corresponding to WDRExposureAttr

Function Module	Corresponding SDK API
WDRExposureAttr	CVI_ISP_SetWDRExposureAttr CVI_ISP_GetWDRExposureAttr

Table 3.7: API corresponding to ExposureInfo

Function Module	Corresponding SDK API
ExposureInfo	CVI_ISP_QueryExposureInfo

Table 3.8: API corresponding to AERoute

Function Module	Corresponding SDK API
AERoute	CVI_ISP_SetAERouteAttr CVI_ISP_GetAERouteAttr CVI_ISP_GetAERouteAttrEx CVI_ISP_SetAERouteAttrEx

Table 3.9: API corresponding to WBAttr

Function Module	Corresponding SDK API
White Balance	CVI_ISP_SetWBAttr CVI_ISP_GetWBAttr
AWBAttr	CVI_ISP_SetWBAttr CVI_ISP_GetWBAttr
MWBAttr	CVI_ISP_SetWBAttr CVI_ISP_GetWBAttr
AWB Calibration Data	CVI_ISP_GetWBCalibration CVI_ISP_SetWBCalibration
AWBAttrEx	CVI_ISP_GetAWBAttrEx CVI_ISP_SetAWBAttrEx

Table 3.10: API corresponding to WBInfo

Function Module	Corresponding SDK API
WBInfo	CVI_ISP_QueryWBInfo

Table 3.11: API corresponding to CCM

Function Module	Corresponding SDK API
CCM	CVI_ISP_SetCCMAttr CVI_ISP_GetCCMAttr
ColorToneAttr	CVI_ISP_GetColorToneAttr CVI_ISP_SetColorToneAttr

Table 3.12: API corresponding to CLUT

Function Module	Corresponding SDK API
CLUT Attr	CVI_MPI_ISP_SetCLutAttr CVI_MPI_ISP_GetCLutAttr

Table 3.13: API corresponding to RadialShading

Function Module	Corresponding SDK API
RadialShading Attr	CVI_MPI_ISP_SetRadialShadingAttr CVI_MPI_ISP_GetRadialShadingAttr
RadialShading Gain Lut	CVI_MPI_ISP_SetRadialShadingGainLutAttr CVI_MPI_ISP_GetRadialShadingGainLutAttr

Table 3.14: API corresponding to Demosaic

Function Module	Corresponding SDK API
Demosaic	CVI_MPI_ISP_SetDemosaicAttr CVI_MPI_ISP_GetDemosaicAttr CVI_MPI_ISP_SetDemosaicDemoireAttr CVI_MPI_ISP_GetDemosaicDemoireAttr CVI_MPI_ISP_SetDemosaicFilterAttr CVI_MPI_ISP_GetDemosaicFilterAttr CVI_MPI_ISP_SetDemosaicEEAttr CVI_MPI_ISP_GetDemosaicEEAttr

Table 3.15: API corresponding to AntiFalse

Function Module	Corresponding SDK API
AntiFalse	CVI_ISP_SetAntiFalseColorAttr CVI_ISP_SetAntiFalseColorAttr



Table 3.16: API corresponding to 3DNR

Function Module	Corresponding SDK API
3DNR	CVI_ISP_GetTNRAttr CVI_ISP_SetTNRAttr CVI_ISP_GetTNRNoiseModelAttr CVI_ISP_SetTNRNoiseModelAttr CVI_ISP_GetTNRlumaMotionAttr CVI_ISP_SetTNRlumaMotionAttr CVI_ISP_GetTNRGhostAttr CVI_ISP_SetTNRGhostAttr CVI_ISP_GetTNRMtPrtAttr CVI_ISP_SetTNRMtPrtAttr

Table 3.17: API corresponding to DRC

Function Module	Corresponding SDK API
DRC	CVI_ISP_SetDRCAttr CVI_ISP_GetDRCAttr

Table 3.18: API corresponding to FSWDR

Function Module	Corresponding SDK API
FSWDR	CVI_ISP_SetFSWDRAttr CVI_ISP_GetFSWDRAttr

Table 3.19: API corresponding to Crosstalk

Function Module	Corresponding SDK API
Crosstalk	CVI_ISP_GetCrosstalkAttr CVI_ISP_SetCrosstalkAttr

Table 3.20: API corresponding to Dehaze

Function Module	Corresponding SDK API
Dehaze	CVI_ISP_SetDehazeAttr CVI_ISP_GetDehazeAttr

Table 3.21: API corresponding to Shading

Function Module	Corresponding SDK API
Shading Attr	CVI_ISP_SetMeshShadingAttr CVI_ISP_GetMeshShadingAttr
Shading Lut Attr	CVI_ISP_SetMeshShadingGainLutAttr CVI_ISP_GetMeshShadingGainLutAttr

Table 3.22: API corresponding to DPC

Function Module	Corresponding SDK API
Dynamic Attribute	CVI_ISP_GetDPDynamicAttr CVI_ISP_SetDPDynamicAttr
Static Attribute	CVI_ISP_GetDPStaticAttr CVI_ISP_SetDPStaticAttr

Table 3.23: API corresponding to CAC

Function Module	Corresponding SDK API
CAC	CVI_ISP_SetCACAttr CVI_ISP_GetCACAttr

Table 3.24: API corresponding to CNR

Function Module	Corresponding SDK API
CNR Attr	CVI_ISP_SetCNRAttr CVI_ISP_GetCNRAttr

Table 3.25: API corresponding to DCI

Function Module	Corresponding SDK API
DCI	CVI_ISP_SetDCIAttr CVI_ISP_GetDCIAttr

Table 3.26: API corresponding to Gamma

Function Module	Corresponding SDK API
Gamma	CVI_ISP_SetGammaAttr CVI_ISP_GetGammaAttr

Table 3.27: API corresponding to DRC

Function Module	Corresponding SDK API
DRC	CVI_MPI_ISP_SetDRCAttr CVI_MPI_ISP_GetDRCAttr

Table 3.28: API corresponding to FSWDR

Function Module	Corresponding SDK API
FSWDR	CVI_MPI_ISP_SetFSWDRAttr CVI_MPI_ISP_GetFSWDRAttr

Table 3.29: API corresponding to HSV

Function Module	Corresponding SDK API
HSV Attr	CVI_ISP_GetHSVAttr CVI_ISP_SetHSVAttr

Table 3.30: API corresponding to LUT 3D

Function Module	Corresponding SDK API
LUT 3D	TBD

Table 3.31: API corresponding to YNR

Function Module	Corresponding SDK API
YNR	CVI_ISP_SetYNRAAttr CVI_ISP_GetYNRAAttr CVI_ISP_SetYNRMotionNRAttr CVI_ISP_GetYNRMotionNRAttr CVI_ISP_SetYNRFilterAttr CVI_ISP_GetYNRFilterAttr

Table 3.32: API corresponding to Y Sharpen

Function Module	Corresponding SDK API
Sharpen	CVI_ISP_GetSharpenAttr CVI_ISP_SetSharpenAttr

Table 3.33: API corresponding to BLC

Function Module	Corresponding SDK API
Black Level	CVI_ISP_GetBlackLevelAttr CVI_ISP_SetBlackLevelAttr

Table 3.34: API corresponding to BNR

Function Module	Corresponding SDK API
BNR	CVI_ISP_SetNRAttr CVI_ISP_GetNRAttr
RLSC	CVI_ISP_SetRLSCAttr CVI_ISP_GetRLSCAttr
Filter	CVI_ISP_SetNRFilterAttr CVI_ISP_GetNRFilterAttr

Table 3.35: API corresponding to Saturation

Function Module	Corresponding SDK API
Saturation	CVI_ISP_GetSaturationAttr CVI_ISP_SetSaturationAttr

Table 3.36: API corresponding to Statistics Config

Function Module	Corresponding SDK API
AE/AF/AWB Config	CVI_ISP_GetStatisticsConfig CVI_ISP_SetStatisticsConfig

Table 3.37: API corresponding to DIS

Function Module	Corresponding SDK API
DIS Attr	CVI_ISP_GetDisAttr CVI_ISP_GetDisAttr
DIS Config	CVI_ISP_GetDisConfig CVI_ISP_SetDisConfig

Table 3.38: API corresponding to LDC

Function Module	Corresponding SDK API
VI LDC	CVI_VI_GetChnLDCAttr CVI_VI_SetChnLDCAttr
VPSS LDC	CVI_VPSS_GetChnLDCAttr CVI_VPSS_SetChnLDCAttr

Table 3.39: API corresponding to Mono

Function Module	Corresponding SDK API
Mono	CVI_ISP_GetMonoAttr CVI_ISP_SetMonoAttr

Table 3.40: API corresponding to Y Contrast

Function Module	Corresponding SDK API
YContrast Attr	CVI_ISP_GetYContrastAttr CVI_ISP_SetYContrastAttr

Table 3.41: API corresponding to Vpss Adjustment

Function Module	Corresponding SDK API
VPSS Adjustment	CVI_VPSS_GetGrpProcAmp CVI_VPSS_SetGrpProcAmp

Table 3.42: API corresponding to Noise Profile

Function Module	Corresponding SDK API
NoiseProfile	CVI_ISP_GetNoiseProfileAttr CVI_ISP_SetNoiseProfileAttr

# 4 Tool Application Reference

## 4.1 How to Import and Export Tool Parameters?

### 4.1.1 Import and Export Parameters Using Tool

CviPQ Tool supports the import and export of configuration files in JSON format on the PC side, and supports the solidification of tool parameters to the board side.

If the user wants to export or import the parameter file of CviPQ tool, please refer to the description in *chapter 2.4.3.1* and *2.4.3.2*.

If the user wants to import parameters to the board or export parameters from the board to the PC for backup, use the “Bin Utility” tool. Specific operation instructions are as follows:

- Import and export bin file of board end parameters

When connected to the board, click Bin Utility in the menu bar to open the binary utility window, as shown in Figure 4-1

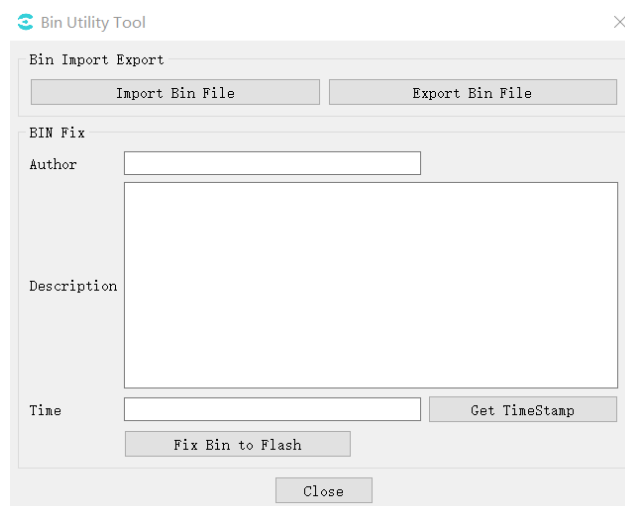


Fig. 4.1: Binary Utility

There are two buttons in Bin Import Export group. Click the buttons respectively to complete the following operations:

- Import parameter bin file to the board side: click “Import Bin File” and select the correct parameter bin file to import in the pop-up open file dialog box. When the tool finishes sending the file to the board side, the parameter will take effect immediately.
- Export board side parameter bin file: click “Export Bin File” and select a save path in the pop-up save file dialog box. The tool will save the board side current parameters in the specified path.
- Solidify parameter configuration to Flash

After filling in the author, description and time information in the Bin Fix group, you can click “Fix Bin to Flash” to send instructions to the board end to write the parameter information of the current board end into Flash.

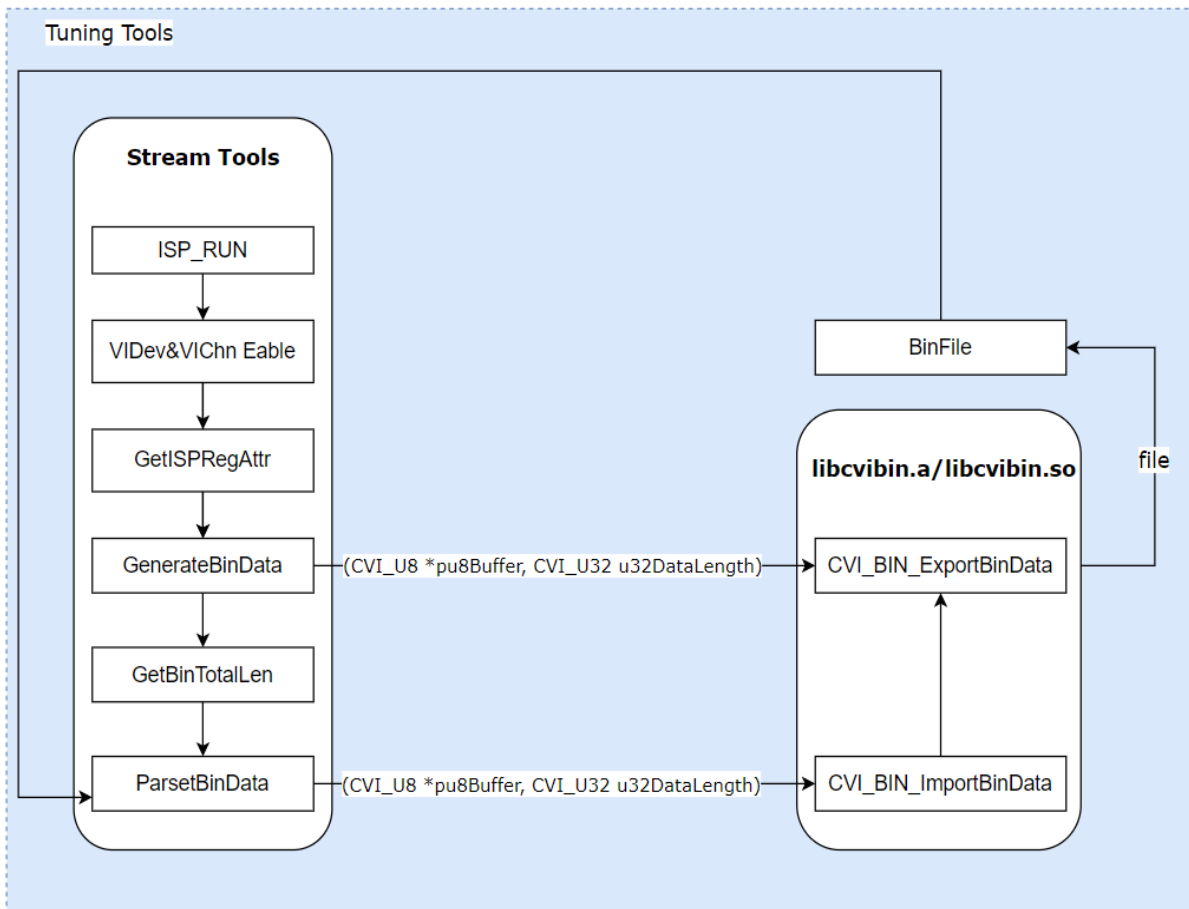
## 4.1.2 Import and export image quality parameters using the library

The board-side tool provides import and export of parameter library files, and the following two methods can be selected according to your needs.

**Note:** The following interfaces must be called after the `CVI_ISP_Init` interface is called.

1. Obtain and save all blocks bin data (ISP x, VPSS, VO, etc.).
  - `CVI_BIN_GetBinTotalLen`: Get the total length of the bin data;
  - `CVI_BIN_ExportBinData`: Export all blocks bin data;
  - `CVI_BIN_ImportBinData`: Import all blocks bin data;
2. Obtain and save single block bin data (ISP x, VPSS, VO, etc.).
  - `CVI_BIN_SaveParamToBin`: Export single block bin data;
  - `CVI_BIN_LoadParamFromBin`: Import single block bin data;
  - `CVI_BIN_LoadParamFromBinEx`: Import single block bin (with input buf length) data;

Please refer to the recommended use process as shown below.



Other related interfaces API:

- *CVI\_BIN\_SetBinName*: Set the path and file name where the PQBin is stored;
- *CVI\_BIN\_GetBinName*: Get the path and filename where the PQBin is stored;
- *CVI\_BIN\_GetBinExtraAttr*: Get bin header data information;

### 4.1.3 API Reference

#### 4.1.3.1 CVI\_BIN\_GetBinTotalLen

**【Description】**

Get the total length of bin data.

**【Syntax】**

```
CVI_U32 CVI_BIN_GetBinTotalLen(void);
```

**【Parameter】**

Parameter	Description	Input/Output
None	None	None

**【Return Value】**

Return Value	Description
CVI_U32	The total length of bin file

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

**【Note】**

This function must be called before calling the `CVI_BIN_ExportBinData` interface.

**【Example】**

```
CVI_U32 u32BinLen = 0;
u32BinLen = CVI_BIN_GetBinTotalLen();
```

**【Related subject】**

- *CVI\_BIN\_ExportBinData*

**4.1.3.2 CVI\_BIN\_ExportBinData****【Description】**

Export all blocks bin data.

**【Syntax】**

```
CVI_S32 CVI_BIN_ExportBinData(CVI_U8 *pu8Buffer, CVI_U32 u32DataLength);
```

**【Parameter】**

Return Value	Description	Input/Output
<code>pu8Buffer</code>	Save buffer of pqbin data	Input
<code>u32DataLength</code>	The total length of pqbin data	Input

**【Return Value】**

Return Value	Description
0	Success
Not 0	Failure, see <i>error code 1</i> for its value

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

**【Note】**



This function must be called after calling the CVI\_BIN\_GetBinTotalLen function to get the size of the data, otherwise memory problems may occur.

**【Example】**

```

CVI_S32 ret = CVI_SUCCESS;
CVI_BIN_HEADER header;
CVI_U32 u32BinLen = 0, u32TempLen = 0;
CVI_U8 *pBuffer;

memset(&header, 0, sizeof(CVI_BIN_HEADER));
u32BinLen = CVI_BIN_GetBinTotalLen();
pBuffer = (CVI_U8 *)malloc(u32BinLen);
if (pBuffer == NULL) {
    ISP_DAEMON_DEBUG(LOG_ALERT, "malloc err!\n");
    return CVI_FAILURE;
}
header.extraInfo = *binExtra;
memcpy(pBuffer, &header, sizeof(CVI_BIN_HEADER));
ret = CVI_BIN_ExportBinData(pBuffer, u32BinLen);
if (ret != CVI_SUCCESS) {
    ISP_DAEMON_DEBUG_EX(LOG_ALERT, "CVI_BIN_ExportBinData err(%#x)!\n", ret);
} else {
    u32TempLen = fwrite(pBuffer, 1, u32BinLen, fp);
    if (u32TempLen != u32BinLen) {
        ISP_DAEMON_DEBUG(LOG_ALERT, "writeIspRegToBin fail\n");
        ret = CVI_FAILURE;
    }
}
if (pBuffer != NULL) {
    free(pBuffer);
}

ISP_DAEMON_UNUSED(numDevice);
return ret;

```

**【Related subject】**

None.

### 4.1.3.3 CVI\_BIN\_ImportBinData

**【Description】**

Import all blocks bin data.

**【Syntax】**

```

CVI_S32 CVI_BIN_ImportBinData(CVI_U8 *pu8Buffer, CVI_U32 u32DataLength);

```

**【Parameter】**

Parameter	Description	Input/Output
pu8Buffer	Memory space of Bin	Input
u32DataLength	Size of Bin	Input

**【Return Value】**

Return Value	Description
0	Success
Not 0	Failure, see <i>error code 2</i> for its value

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so` `libcvi_bin_isp.so`

**【Note】**

None.

**【Example】**

```

CVI_S32 ret = CVI_SUCCESS;
FILE *fp = NULL;
CVI_U8 *buf = NULL;
CVI_CHAR binName[BIN_FILE_LENGTH] = { 0 };
CVI_U32 u32TempLen = 0, u32FileSize = 0;

ret = CVI_BIN_GetBinName(binName);
if (ret != CVI_SUCCESS) {
    CVI_TRACE_SYS(CVI_DBG_WARN, "GetBinName(%s) fail\n", binName);
}

fp = fopen((const CVI_CHAR *)binName, "rb");
if (fp == NULL) {
    CVI_TRACE_SYS(CVI_DBG_WARN, "Can't find bin(%s)\n", binName);
    ret = CVI_FAILURE;
    goto ERROR_HANDLER;
} else {
    CVI_TRACE_SYS(CVI_DBG_WARN, "Bin exist (%s)\n", binName);
}

getFileSize(fp, &u32FileSize);
buf = (CVI_U8 *)malloc(u32FileSize);
if (buf == NULL) {
    ret = CVI_FAILURE;
    CVI_TRACE_SYS(CVI_DBG_WARN, "Allocate memory fail\n");
    goto ERROR_HANDLER;
}

u32TempLen = fread(buf, u32FileSize, 1, fp);
if (u32TempLen <= 0) {
    CVI_TRACE_SYS(CVI_DBG_WARN, "read data to buff fail!\n");
}

```

(continues on next page)

(continued from previous page)

```

    ret = CVI_FAILURE;
    goto ERROR_HANDLER;
}

ret = CVI_BIN_ImportBinData(buf, (CVI_U32)u32FileSize);
if (ret != CVI_SUCCESS) {
    CVI_TRACE_SYS(CVI_DBG_WARN, "CVI_BIN_ImportBinData error! value:(0x%x)\n",
    ↪ret);
    goto ERROR_HANDLER;
}

ERROR_HANDLER:
if (fp != NULL) {
    fclose(fp);
}
if (buf != NULL) {
    free(buf);
}

```

**【Related subject】**

None

**4.1.3.4 CVI\_BIN\_SaveParamToBin****【Description】**

Export single block bin data

**【Syntax】**

```
CVI_S32 CVI_BIN_SaveParamToBin(FILE *fp, CVI_BIN_EXTRA_S *extraInfo);
```

**【Parameter】**

Parameter	Description	Input/Output
fp	PQBin file pointer	Input
extraInfo	PQBin information and description	Input

**【Return Value】**

Return Value	Description
0	Success
Not 0	Failure, see <i>error code 1</i> for its value

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

**【Note】**

None

**【Example】**

```

CVI_BIN_EXTRA_S BinExtra = {
    "User",
    "test-default",
    "2021-04-19",
};

FILE *fd = fopen(TEST_BIN, "wb");

if (fd == NULL) {
    ISP_DEBUG(LOG_ERR, "Open file failed\n");
}

CVI_BIN_SaveParamToBin(fd, &BinExtra);
fclose(fd);

```

**【Related subject】**

None

**4.1.3.5 CVI\_BIN\_LoadParamFromBin****【Description】**

Import single block bin data.

**【Syntax】**

```

CVI_S32 CVI_BIN_LoadParamFromBin(enum CVI_BIN_SECTION_ID id, CVI_U8 *buf);

```

**【Parameter】**

Parameter	Description	Input/Output
id	Module ID	Input
buf	Data contents of PQBin	Input

**【Return Value】**

Return Value	Description
0	Success
Not 0	Failure, see <i>error code 2</i> for its value

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

**【Note】**

PQBin must be read from the data to the cache memory space before calling this function. This API function will be replaced by CVI\_BIN\_LoadParamFromBinEx and it is recommended to use it instead.

**【Example】**

```
CVI_S32 result = CVI_FAILURE;
CVI_U32 size = 0;
FILE *fp;

ISP_DEBUG(LOG_DEBUG, "try load default value from bin %s\n", TEST_BIN);

fp = fopen(TEST_BIN, "rb");
if (fp == NULL) {
    ISP_DEBUG(LOG_WARNING, "Cant find bin(%s)\n", TEST_BIN);
} else {
    ISP_DEBUG(LOG_INFO, "Bin exist (%s)\n", TEST_BIN);
}

fseek(fp, 0L, SEEK_END);
size = ftell(fp);
rewind(fp);

if (size > 0) {
    // allocate buffer
    CVI_U8 *binBuffer = malloc(size);

    fread(binBuffer, size, 1, fp);
    for (CVI_U32 idx = CVI_BIN_ID_MIN ; idx < CVI_BIN_ID_MAX ; idx++) {
        result = CVI_BIN_LoadParamFromBin((enum CVI_BIN_SECTION_ID)idx, binBuffer);
    }

    // free buffer
    free(binBuffer);

    if (result == CVI_SUCCESS) {
        ISP_DEBUG(LOG_DEBUG, "load default value from bin %s\n", TEST_BIN);
    }
}
```

**【Related subject】**

*CVI\_BIN\_LoadParamFromBinEx*

#### 4.1.3.6 CVI\_BIN\_LoadParamFromBinEx

##### 【Description】

The same as *CVI\_BIN\_LoadParamFromBin*.

##### 【Syntax】

```
CVI_S32 CVI_BIN_LoadParamFromBin(enum CVI_BIN_SECTION_ID id, CVI_U8 *buf, CVI_U32 u32DataLength);
```

##### 【Parameter】

Parameter	Description	Input/Output
id	Module ID	Input
buf	Data contents of PQBin	Input
u32DataLength	Size of buf	Input

##### 【Return Value】

Return Value	Description
0	Success
Not 0	Failure, see <i>error code 2</i> for its value

##### 【Requirement】

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

##### 【Note】

The same as *CVI\_BIN\_LoadParamFromBin*.

##### 【Example】

Please refer to *CVI\_BIN\_LoadParamFromBin*, note that: the size of buf needs to be imported.

#### 4.1.3.7 CVI\_BIN\_SetBinName

##### 【Description】

Set the path and file name where PQBin is stored.

##### 【Syntax】

```
CVI_S32 CVI_BIN_SetBinName(WDR_MODE_E wdrMode, const CVI_CHAR *binName);
```

##### 【Parameter】

Parameter	Description	Input/Output
wdrMode	Sensor wdr mode	Input
binName	Set the path and file name of the pqbin corresponding to the wdr mode Ex. “/mnt/data/cvi_sdr_bin”	Input

**【Return Value】**

Return Value	Description
0	Success
Not 0	Failure, see error code

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

**【Note】**

None

**【Example】**

None

**【Related subject】**

*CVI\_BIN\_GetBinName*

**4.1.3.8 CVI\_BIN\_GetBinName****【Description】**

Get the path and filename where the PQBin is stored.

**【Syntax】**

```
CVI_S32 CVI_BIN_GetBinName(CVI_CHAR *binName);
```

**【Parameter】**

Parameter	Description	Input/Output
binName	Path and file name of pqbin	Output

**【Return Value】**

Return Value	Description
0	Success
Not 0	Failure, see error code

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

**【Note】**

None

**【Example】**

None

**【Related subject】**

*CVI\_BIN\_SetBinName*

**4.1.3.9 CVI\_BIN\_GetBinExtraAttr****【Description】**

Get bin header data information.

**【Syntax】**

```
CVI_S32 CVI_BIN_GetBinExtraAttr(FILE *fp, CVI_BIN_EXTRA_S *extraInfo)
```

**【Parameter】**

Parameter	Description	Input/Output
<code>fp</code>	PQBin file pointer	Input
<code>extraInfo</code>	Memory space of bin header information	Input

**【Return Value】**

Return Value	Description
0	Success

**【Requirement】**

- Header file: `cvi_bin.h`
- Library file: `libcvi_bin.so`, `libcvi_bin_isp.so`

**【Note】**

None.

**【Example】**

```
CVI_BIN_EXTRA_S BinExtraAttr = {};
FILE *fd = fopen("/mnt/data/bin/cvi_sdr_bin", "rb");
CVI_BIN_GetBinExtraAttr(fd, &BinExtraAttr);
```

**【Related subject】**

None



**4.1.3.10 Error code 1**

Interface Return Value	Meaning
0xCB000001	The entry pointer is null
0xCB000003	No memory space allocated
0xCB00000B	json compression failure
0xCB00000D	Insufficient space in the input buffer
0xCB00000F	Failure to update parameters to PQbin file
0xCB000012	Failure to create json handle
0xCB000013	The block id entered is invalid
0xCB000014	Failure to read parameter from file

**4.1.3.11 Error code 2**

Interface Return Value	Meaning
0xCB000001	The entry pointer is null
0xCB000003	No memory space allocated
0xCB000006	The size of the input buf is 0
0xCB000008	There is an exception in the data in PQbin
0xCB00000C	json decompression failure
0xCB00000F	Failure to update parameters to PQbin file
0xCB000010	Unable to find bin file
0xCB000011	The json parameter in the current PQbin file is invalid
0xCB000012	Failure to create json handle
0xCB000013	The block id entered is invalid
0xCB000014	Failure to read parameter from file
0xCB000015	The currently used PQbin file is invalid
0xCB000016	The number of sensors exceeds the number specified in the PQbin file

# 5 FAQ

## 5.1 Why Does the Tool Indicate a Version Mismatch When Opening the Debug Data File?

When the tool opens the debug data file, it will check the version information in the file, and an example of the version information is shown below.

```
"JSON Info": [  
  {  
    "key": "SoC",  
    "value": "CV183X"  
  },  
  {  
    "key": "CviPQ Tool Version",  
    "value": "v1.8.0.0"  
  },  
  {  
    "key": "Parameters Version",  
    "value": "cv183x v1.0.0"  
  },  
  {  
    "key": "DateTime",  
    "value": "20210421214211"  
  }  
],
```

Fig. 5.1: Version information in the debug data file

If the version information within the debug data file does not match the tool version, there are three prompt conditions:

1. If the file does not contain version information, the following message will be prompted:

**[Suggested handling method]** Read the default data of the current SDK version, and then open the debug data file, so that it can contain the default data of the difference parameters, and also keep the debug data of the file.

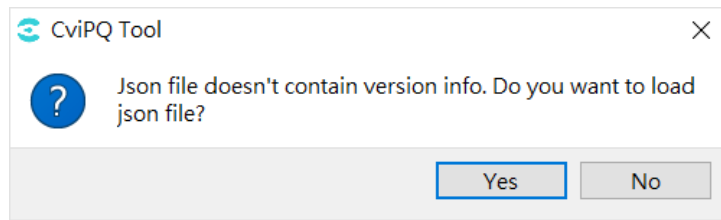


Fig. 5.2: Prompt message of no version information

2. If the SoC information inside the file does not match the debug tool, the following message will be prompted.

**[Suggested handling method] Check whether the file and the SoC to which the debug tool works are the same.**

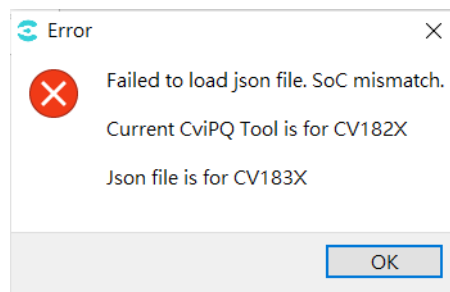


Fig. 5.3: SoC mismatch prompt message

3. If the parameter version information within the file does not match, the following message will be prompted.

**[Suggested handling method] Read the default data of the current SDK version, and then open the debug data file, so that it can contain the default data of the difference parameters, and also keep the debug data of the file.**

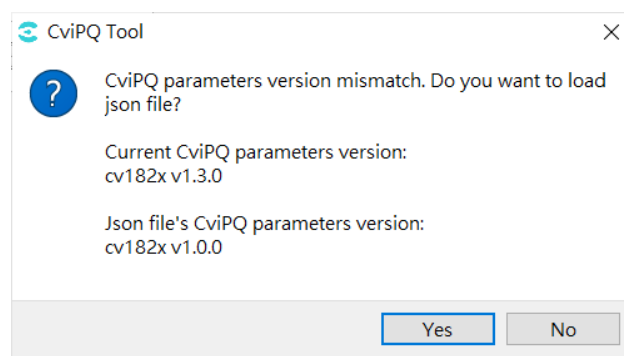


Fig. 5.4: Prompt message for parameter version mismatch

## 5.2 Why Does the Tool Prompt SoC Mismatch When Connecting To the Board?

When the tool connects to the board, it will check if the board's SoC matches the tool. If it does not match, the following error message will be displayed and the connection will be interrupted.

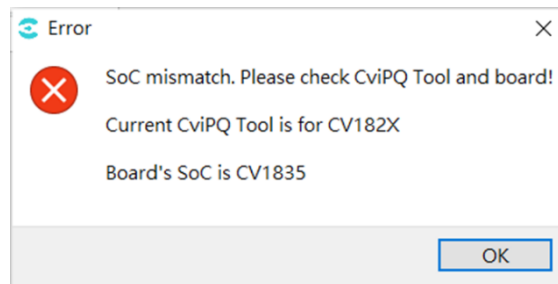


Fig. 5.5: SoC mismatch prompt message

[Suggested handling method] Confirm whether the SoC shown on the top left of the tool is the same as the board.

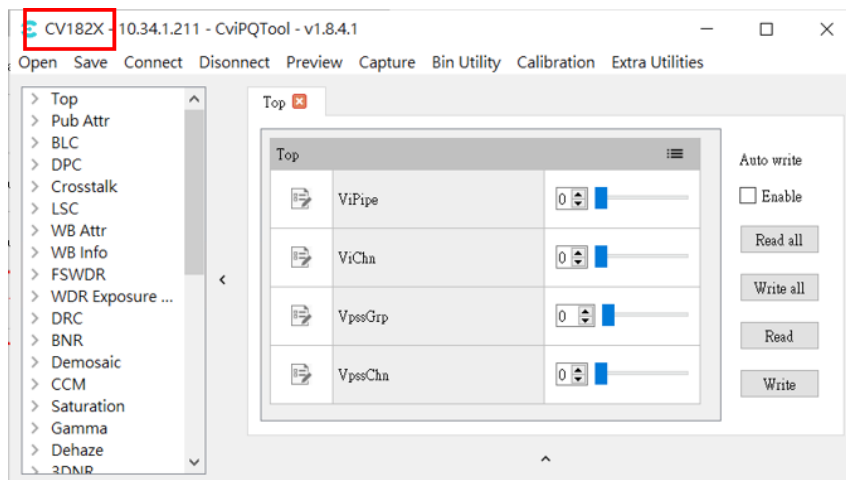


Fig. 5.6: Tool SoC information