Dual-Camera Jetson Orin Prototype (GC02M1+GC50B2) - User Guide

Technical Doc

David Song

6/3/2025

1. Introduction

This document provides an overview of our **GC02M1 + GC50B2 dual-camera solution** built on the NVIDIA Jetson Orin NX platform. The prototype integrates a 50 MP RGB **main camera** (GC50B2) with a secondary **multispectral** sensor (GC02M1) to deliver enhanced imaging capabilities. The multispectral camera captures additional spectral data across nine channels, enabling advanced features such as improved auto white balance and color accuracy beyond the standard RGB spectrum.

In the following sections, we outline the hardware specifications of the Jetson Orin-based prototype and the dual cameras, detail the software environment, and guide the user through the demonstration application's operation. This guide is intended for a technically literate audience evaluating our camera module solution, with an emphasis on clear organization and professional presentation.

2. Version Control

2.1. Document Revision History

Version	Date	Author/Edtor	Key Changes
1.0	2025-04-22	David Song	Initial draft.
1.1	2025-06-16	Mengxuan Yu	Translation; Language polish.

2.2. Software Release History

Version	Date	Key Changes
1.0	2025-04-22	
1.1	2025-06-16	Add Zoom, Burst Capture, more display mode; Simplify manual focus and exposure control; optimize Auto Focus; improve GUI; add one-tap desktop launcher.

3. Hardware Overview

3.1. Platform

NVIDIA Jetson Orin NX 8GB Module: This prototype uses the Jetson Orin NX 8GB system-on-module as the computing platform. Key hardware specifications of the Orin NX module are:

Sub-system	Specification (Jetson Orin NX 8 GB)
AI performance	Up to **70 TOPS** INT8
GPU	1024-core **NVIDIA Ampere** GPU with **32 Tensor Cores**
GPU max clock	765 MHz

Sub-system	Specification (Jetson Orin NX 8 GB)
CPU	6 × Arm Cortex-A78AE v8.2 64-bit (1.5 MB L2 + 4 MB L3)
CPU max clock	2.0 GHz
Deep-learning accelerator	1 × NVDLA v2.0 (max 1.2 GHz)
System memory	8 GB LPDDR5 (128-bit, 102.4 GB/s)
On-module storage	16 GB eMMC 5.1
Video encode	1 × 4K 60 fps (H.265) • 3 × 4K 30 fps • 6 × 1080p 60 fps • 12 × 1080p 30 fps
Video decode	1 × 8K 30 fps (H.265) • 2 × 4K 60 fps • 4 × 4K 30 fps • 9 × 1080p 60 fps • 18 × 1080p 30 fps
Camera interfaces	1 × 8-lane MIPI CSI-2 (≃ 33.4 Gbps)
PCI Express	1 × 4-lane PCIe Gen 4 (RC / EP selectable)
USB	3 × USB 3.2 Gen 2 (10 Gbps) + 3 × USB 2.0
Networking	1 × GbE MAC
Display outputs	1 × eDP 1.4a / DP 1.4a / HDMI 2.1 (up to 4K 60 fps)
Additional I/O	3 × UART, 2 × SPI, 4 × I^2S, 4 × I^2C, CAN, DSI/DSPI, PWM, GPIO
Power envelope	Configurable **10 W – 20 W**
Form-factor	69.6 mm × 45 mm, 260-pin SO-DIMM
Max GPU Grid Size:	x: 2147483647 y: 65535 z: 65535

3.2. CMOS Image Sensor Info

3.2.1. Main Camera

Item	Specification
Sensor	GC50B2
Active Pixel Numbers	8192(H) x 6144(V)
Resolution	15 fps @ 50MP 60 fps @ Bin4K
Unit Pixel size	1.0μm(H) x 1.0μm(V)

3.2.2. multispectral image sensor

Item	Specification
Sensor	GC02M1
Active Pixel Numbers	1600(H) x 1200(V)
Resolution	30fps @ UXGA & HD 720P 60fps @ SVGA
Unit Pixel size	1.75μm(H) x 1.75μm(V)

4. Software/OS Overview

NVCC：

nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2024 NVIDIA Corporation
Built on Tue_Oct_29_23:53:06_PDT_2024
Cuda compilation tools, release 12.6, V12.6.85
Build cuda_12.6.r12.6/compiler.35059454_0

Cmake：cmake version 3.22.1

Ubuntu：22.04

5. Software Operation Guide

5.1. Prerequisites

5.1.1. CUDA Toolkit

CUDA capable GPU (Jetson Orin, compute capability 8.7)

CUDA Toolkit matching your system

5.1.2. C++ Development Environment

C++17 compatible compiler

CMake (version 3.10 or higher)

5.1.3. Python Environment

Python 3

Python development headers

pybind11 package

5.1.4. Computer Vision

OpenCV library and development files

5.1.5. AI/ML Dependencies

ONNX Runtime (version 1.21.0)

ONNX Runtime development headers

5.1.6. System Libraries

pthread (threading library)

5.2. Project file structure

DualCam_Demo /

```
├──── hgui3.py              #PySide6 GUI Blueprint
├──── install_shortcut.sh   #Script for shortcut install
├──── main.py               #executable
├──── update_check.sh       #Script for Update check
├──── icon/                 #Icon
|    └────logo_color.ico
├──── lib/                  #library folder
|    ├──── aec_stats.so
|    ├──── af_control.so
|    ├──── af_stats.so
|    ├──── af_tenegrad.so
|    ├──── awb.so
|    ├──── ccm.so
|    ├──── demosaic.so
|    ├──── dn.so
|    ├──── dpc.so
|    ├──── ee_dn_gamma.so
|    ├──── extend.so
|    ├──── isp_engine.so
|    ├──── isp_engine_py.so
|    ├──── isp_pipeline.so
|    ├──── libonnxruntime.so
|    ├──── libonnxruntime.so.1
|    ├──── libonnxruntime.so.1.21.0
|    ├──── libv4l2_helper.so
|    ├──── lsc.so
|    ├──── MSI_aec_stats.so
|    ├──── MSI_awb.so
|    ├──── MSI_crop.so
|    ├──── MSI_dpc.so
|    ├──── MSI_lsc.so
|    ├──── MSI_render_utils.so
```

```
│   ├── ob.so
│   └── render_utils.so
└── models/          #ONNX model folder
    ├── multispec_net_0516.onnx
    ├── multispec_net_led_0427.onnx
    ├── random_forest_color_xy_model.onnx
    ├── random_forest_model_xy_0.onnx
    ├── random_forest_model_xy_1.onnx
    ├── random_forest_model_xy_10.onnx
    ├── random_forest_model_xy_2.onnx
    ├── random_forest_model_xy_3.onnx
    ├── random_forest_model_xy_4.onnx
    ├── random_forest_model_xy_5.onnx
    ├── random_forest_model_xy_6.onnx
    ├── random_forest_model_xy_7.onnx
    ├── random_forest_model_xy_8.onnx
    └── random_forest_model_xy_9.onnx
```

5.3. Launching the Application

5.3.1. Launch via Terminal:

Open a terminal window and run the following commandsl:

cd Path/To/DualCam_Demo/
LD_LIBRARY_PATH=./lib python main.py

5.3.2. Launch via Desktop Shortcut:

- Initial Installation (first-time only): Open a terminal and execute:

  cd Path/To/DualCam_Demo/
  chmod +x ./install_shortcut.sh
  ./install_shortcut.sh

- Launching the Application:
  Search the **Camera Module** icon in the application menu (the 9 dot icon at the bottom left corner). You may also add it to your Favorites by dragging the icon. Click the icon once to launch the application.

5.4. Update Check:

To check for updates, open a terminal and execute:

cd Path/To/DualCam_Demo/

chmod +x ./update_check.sh

./update_check.sh

- If there's new version available, the script will prompt: *Proceed with HARD RESET to remote? (all local changes will be lost) (y/n)*,

Enter 'y' to process update, 'n' to cancel.

- If you are already on the latest version but have local modifications, The script will prompt:

  Available actions:

  1. Keep local changes (default)

  2. Discard all local changes

  3. View change differences

Enter 1 to skip the update.

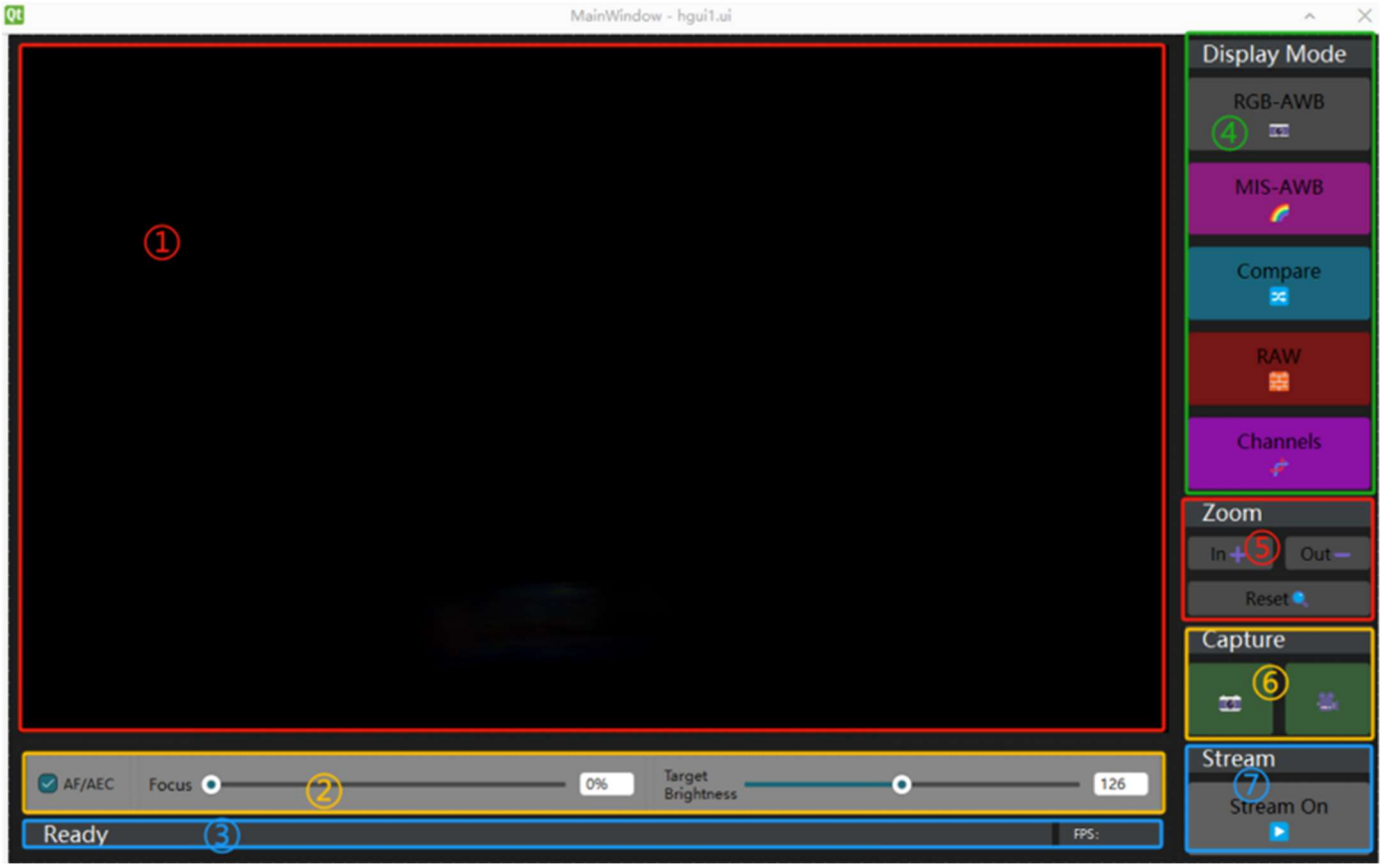
Enter 2 to discard all local modifications and synchronize with the remote repository.

Enter 3 to review the local changes before deciding.

- If the local repository is already up to date and unmodified, the script will confirm: *Repository is fully synchronized with origin/main.*

6. GUI Introduction

6.1. GUI Screen Shoot:



6.2. GUI Component Overview

Note The numbers ①–⑦ correspond to the call-outs in the screenshot.

① **Display Area**: Shows the live video feed from the selected camera. All zooming, panning and comparison actions happen here.

② **AF/AEC Pannel**:

a) When the AF/AEC CheckBox is checked:
- *Focus* slider is disabled (autofocus is active).
- *Target Brightness* slider / text box sets the Auto-Exposure target (range **0–255**).

b) When the AF/AEC CheckBox is unchecked:
- *Focus* slider becomes active (range **0 %–100 %**); far focus on the left, macro on the right.
- *EV* slider replaces *Target Brightness* (range **EV –2 to EV 6**).
  EV 0 ≙ 33 ms exposure & 1× gain.
  Final brightness ≈ (brightness at EV 0) × 2^(EV value).

c) All values can be typed directly into the text boxes.

③ **Info Bar**:

a) Left side: operation results and error messages.
b) Right side: real-time FPS when streaming is active.

④ **Display mode selector**: We provided 5 display modes to demonstrate our Sensor.

a) **RGB-AWB**: Main camera GC50B2 (3072 × 4096) with **standard** auto white balance.
b) **MIS_AWB**: GC50B2 using AWB gains calculated from MIS data. Hovering over the Display Area shows a local CCT grid.
c) **Compare**: Left half: standard AWB; right half: MIS-AWB for side-by-side comparison.
d) **RAW**: Raw Bayer image from multispectral camera GC02M1 (1200 × 1600).
e) **Channels**: GC02M1 with its nine spectral channels displayed separately.

⑤ **Zoom Toolbox**:

a) **Zoom In / Out**: click anywhere in the Display Area to zoom around the cursor.
b) **Reset**: re-center and fit the image to the window.

⑥ **Capture Button**: Active only while streaming.

a) Clicking 📷 (left button) saves the following to ./out:
- Raw image from GC50B2 (main camera)
- Raw image from GC02M1 (multispectral camera)
- GC50B2 raw image after Lens-Shading Correction (before AWB)
- Fully processed JPEG from GC50B2
- • Applies standard or MIS AWB according to the current/last display mode.
- Rendered JPEG from GC02M1 with its nine channels separated
- A .json file containing sensor settings for both cameras at capture time

b) Clicking 📹 (right button) to enable **Burst Capture Mode**. A dialog window will appear prompting you to enter the number of RAW images to capture. Once confirmed, the main camera will begin capturing unprocessed RAW frames at the currently configured frame rate. The capture process will automatically stop once the specified number of frames has been acquired.

⑦ **Stream Control**:

a) **Stream On**: starts both camera streams; the button label changes to **Stop & Quit**.

b) **Stop & Quit**: stops streaming and exits the application.