

# Use MIPI camera

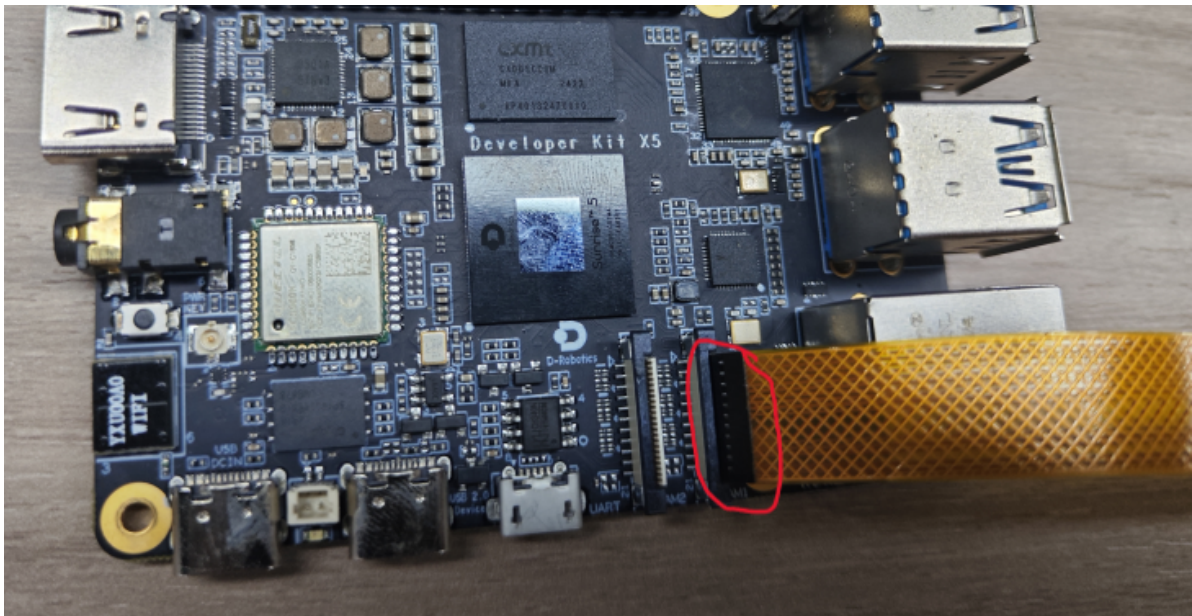
## Use MIPI camera

1. Environment preparation
2. Run mode
3. Expected results

The development board is installed with the `mipi_camera.py` program to test the data path of the MIPI camera. This example will collect the image data of the MIPI camera in real time, then run the target detection algorithm, and finally fuse the image data and the detection results and output them through the HDMI interface.

## 1. Environment preparation

- Connect the power supply
- Connect the MIPI camera module to the MIPI CSI interface of the development board. Note that the black side is facing the HDMI port.



- Connect the development board and the monitor via an HDMI cable
- Log in to RDK X5 via MobaXterm or other ssh tools

## 2. Run mode

Execute the program according to the following command

Turn off the desktop display

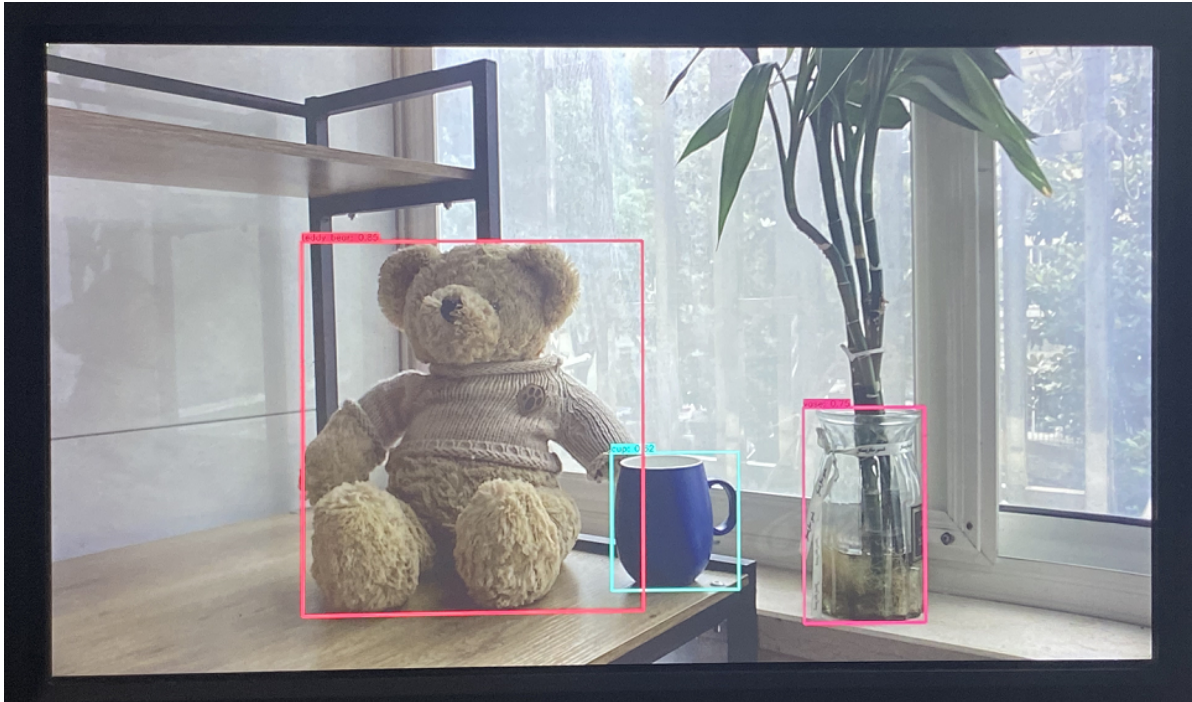
```
sudo systemctl stop lightdm
```

Running MIPI Examples

```
sunrise@ubuntu:~$ cd /app/pydev_demo/03_mipi_camera_sample/  
sunrise@ubuntu:/app/pydev_demo/03_mipi_camera_sample$ sudo python3  
./mipi_camera.py
```

### 3. Expected results

After the program is executed, the display will show the camera image and the results of the target detection algorithm (target type, confidence) in real time, as shown below:



Note: When using the 02 and 03 examples of /app/pydev\_demo of RDK X5, there is no screen display, only a black window.

- (1) First, you need to perform apt upgrade on the premise that apt update has the sweet potato apt source, and upgrade all packages starting with hobot\* to the latest version.
- (2) The significance of this example is to directly output nv12 data from HDMI hardware through BT1120 instead of displaying it on the xfce desktop. The gray window is not produced by cv2.imshow.
- (3) When running the Desktop system, if you run an example that requires an HDMI preview screen, you need to first execute `sudo systemctl stop lightdm` to close the graphical interface, otherwise an error will be reported and HDMI will not display the expected image.