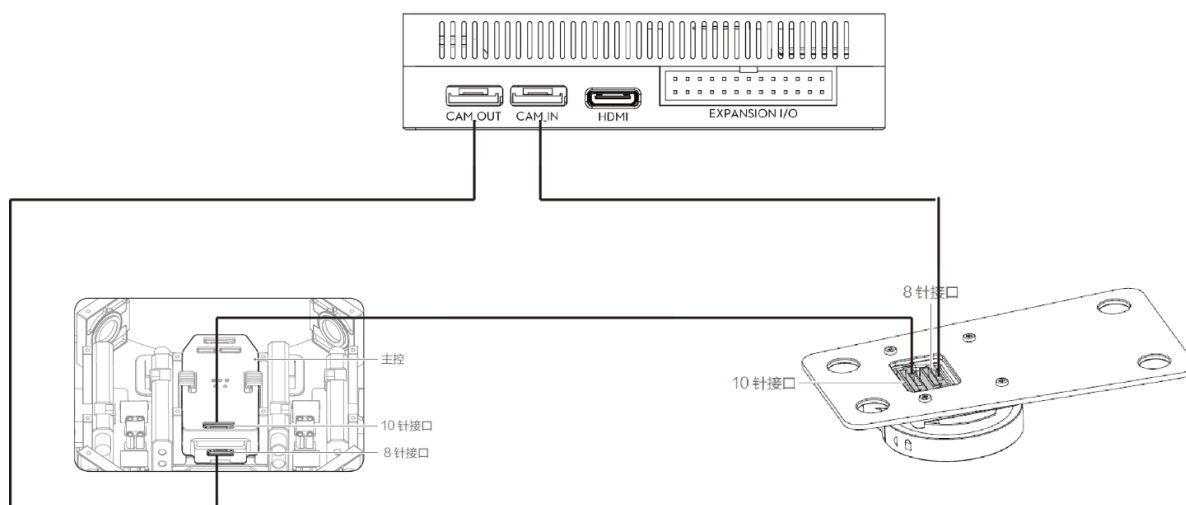


使用 Manifold 获取 M100 上的 X3 摄像头数据

本文将介绍如何在妙算开发板上使用库 libdcam.so 文件，获取从 M100 上的 Zenmuse X3 相机数据以实现实时播放和码流数据的透传等功能。

一、硬件连接

在挂载 X3 相机的云台底端有两个接口，分别为 10-pins 和 8-pins。首先把相机云台上的 8-pins 接口通过 Manifold 附带的 8pin 接口线，连接到 Manifold 的 CAM_IN 接口；然后用另外一根附带的 8pin 接口线把 Manifold 的 CAM_OUT 接口连接到飞控的 8-pins 信号接口上。另外云台底端的 10pin 接口直接连接到飞控上的 10pin 接口。



参考视频“Manifold Tutorials - Mounting Manifold onto Matrice 100”<https://developer.dji.com/manifold/videos/> 以了解详细的连接方法。

二、使用 libdcam.so

进入妙算默认操作系统，通过访问路径 (/usr/lib/libdcam.so) 找到 libcam.so 库文件。Libcam.so 库文件的程序范例文件存放于该目录 /home/ubuntu/demo/manifold_cam 下。

1、编译

(1) 进入”manifold_cam”目录，使用 make 命令进行编译，编译通过后，生成可执行文件 test。

```
$ cd /home/ubuntu/manifold_cam
$ make
```

(2) 如果要删除可执行文件 test 及中间文件，则执行命令：

```
$ make clean
```

2、运行

编译生成的 test 文件为可执行文件。执行该文件可实现如下几项功能：视频的实时播

放；码流数据的获取；码流数据的透传。

执行如下命令可测试相机是否正常工作：

```
$ sudo ./test [-dgt]
```

其中，-d、-g、-t 分别为可选参数，可配置一项或多项。**需要注意的是，-d 和-g 选项不可同时配置。**具体含义如下：

-d: 表示实时解码并播放视频流

-g: 表示将解码后的视频帧以 NV12 格式保存

-t: 表示将获取的视频码流透传到飞控

注意：参数缺省或配置参数有误时，系统打印 usage 使用信息，提示用户正确配置参数。

当需要终止程序时，键盘上同时按下 **Ctrl+c**，系统发出终止信号，程序可正常退出。

4、libdcam.so 接口函数

(1) int manifold_cam_init(int mode);

功能

初始化函数，用于系统的初始化。

参数

mode: 模式设置。可设置为 DISPLAY_MODE、GETBUFFER_MODE、TRANSFER_MODE，对应于例程中的-d、-g、-t 选项。

返回值

0: 表示初始化设置成功

-1: 表示初始化设置失败

(2) int manifold_cam_exit();

功能

退出函数，用于检测系统内部是否已正常退出。

返回值

0: 表示系统未退出或未完全退出

1: 表示系统已正常退出

(3) int manifold_cam_read(unsigned char *buffer, unsigned int *nframe, unsigned int block);

功能

用于获取解码后的视频帧数据。

参数

buffer: 存放解码后一帧的数据，数据格式为 NV12

nframe: 获取的第 N 帧

block: 可设置为阻塞方式 CAM_BLOCK，或者非阻塞方式 CAM_NON_BLOCK

返回值

在阻塞方式 CAM_BLOCK 下，

>0: 表示获取一帧数据的大小

<0: 表示捕获到 Ctrl+c 中止信号，停止获取数据；或者其它异常

在非阻塞方式 CAM_NON_BLOCK 下，

>0: 表示获取一帧数据的大小

=0: 表示最新一帧数据未更新

<0: 表示捕获到 Ctrl+c 中止信号，停止获取数据；或者其它异常

本示例已经上传到 **github**，网址如下。请通过 **git** 命令及时更新。

https://github.com/dji-sdk/manifold_cam.git