Product Manual
G1 Gimbal Camera

Ver. 20220919

Introduction to Gimbal Camera

AMOVLAB G1 Gimbal Camera:

The OV OS12D40 color image sensor used in the 3-axis stabilized gimbal camera module is a high-performance CMOS 1/2.49" image sensor that delivers 11.3 megapixel (4512×2512) image signals at up to 60fps. Pixel dots are 1.404 μm×1.404 μm. It supports 11M pixel HD image shooting, and supports up to 4K@60fps (difference) and 4K@30fps video recording. Smart gimbal camera = gimbal + camera + AI chip + human - computer interaction software + deep learning

G1 Gimbal Camera

1. Introduction:

AMOVLAB G1 Gimbal Camera: The OV OS12D40 color image sensor used in the 3-axis stabilized gimbal camera module is a high-performance CMOS 1/2.49" image sensor that delivers 11.3 megapixel (4512×2512) image signals at up to 60fps. Pixel dots are 1.404 μm×1.404 μm. It supports up to 4K@30fps video recording. Smart gimbal camera = gimbal + camera + AI chip + human-computer interaction software + deep learning.

Application scenarios:

- Target acquisition
- Target recognition
- Target tracking
- Target attack
- Missile searching
- Data collection
- ......

The components included in the product are as follows:

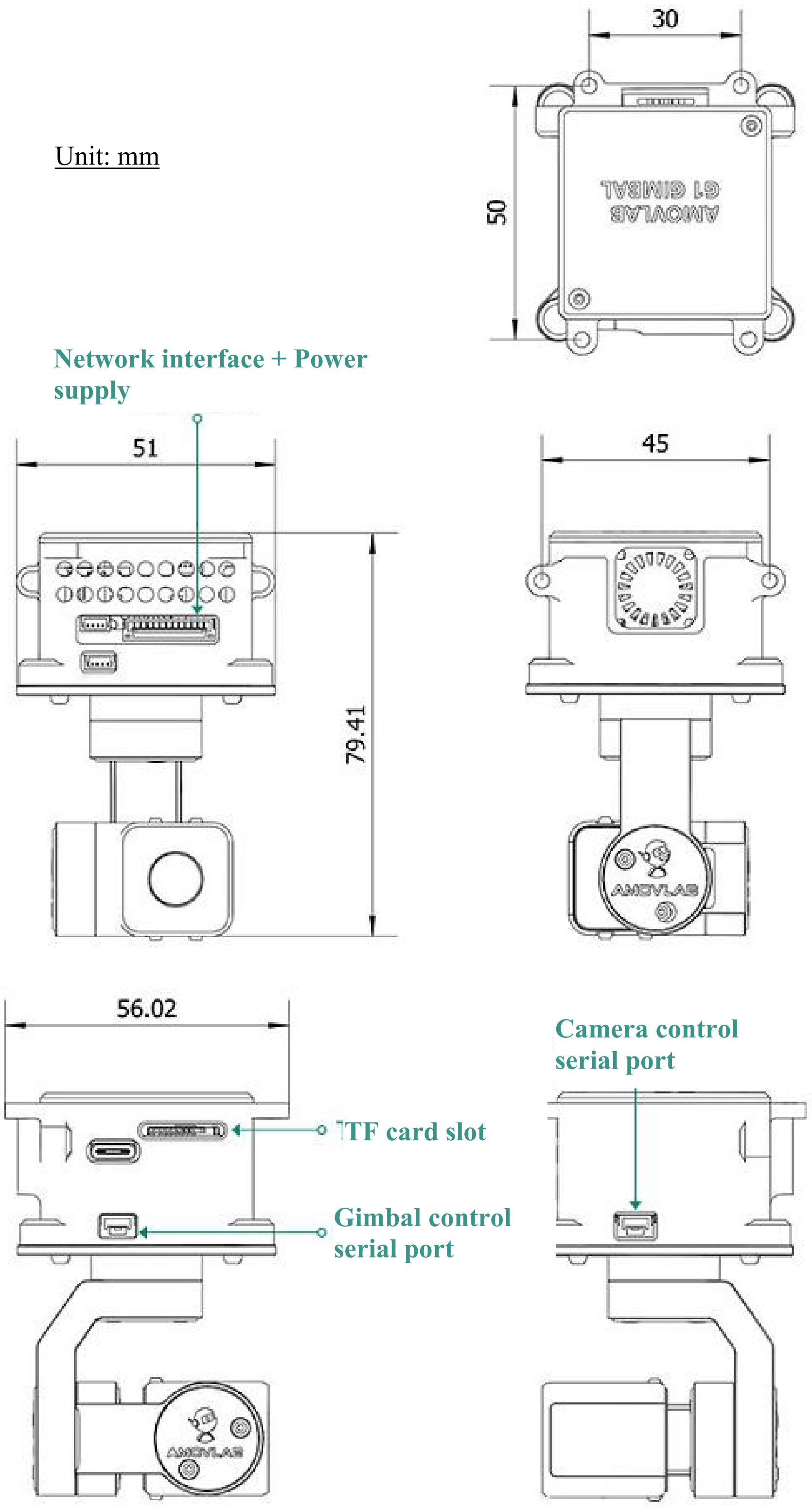
- Gimbal camera
- Three-in-one cable (network cable, power cord, and camera control line)
- Serial port control line
- Gimbal adjustable support

3. Appearance:

AMOVLAB
G1 GIMBAL

4. Dimensions and weight:

Unit: mm


30
50
AMOVLAB G1 GIMBAL
Network interface + Power supply
51
79.41
45
AMOVLAB
56.02
TF card slot
Gimbal control serial port
Camera control serial port
AMOVLAB

Weight: 104g

3D model download address: G1 吊舱 3D 模型 (amovlab.com)

- The gimbal camera supports upright (on the UAV) and inverted (on the vehicle or robot dog) installation mode, with the default mode of upright. To switch to inverted installation mode, please contact the customer service in advance.

5. Camera:

Image sensor	OV OS12D40
Sensor type	CMOS
Effective pixels	11.3M
Sensor size	1/2.49”
Pixel dot size	1.4×1.4（μm）
Video format	H.264
Video resolution	4K@24/25/30/fps 2.7K@24/25/30/48/50/60fps 1080P@24/25/30/48/50/60/120fps 720P@24f/25/30/48/50/60/120/240fps
Support RTSP video stream	with the minimum bandwidth of 500 Kb, and maximum bandwidth of 10 Mb

6. Lens:

Focal Length	15.43mm
Aperture (F/NO.)	2.0
Diagonal Field of View FOV(D)	143° (y’=3.625mm)
Vertical Field of View FOV(V)	69° (y’=1.763mm)
Horizontal Field of View FOV(H)	125° (y’=3.167mm)
TV Distortion	<-33%
Relative Illumination	>66%
Operation Temperature	-20~+60℃

Note: The lens does not support optical zoom.

7. Gimbal:

Axis	Mechanical 3-axis

Axis	Mechanical 3-axis
Controllable angle	Pitch: +90°~-30° Roll: ±45° Yaw: ±60°
Maximum controllable speed	180° /S
Angle jitter	±0.005°
SDK supports angle and angular speed control	Linux SDK 和 ROS SDK
SDK supports IMU and encoder angle feedback	Linux SDK 和 ROS SDK

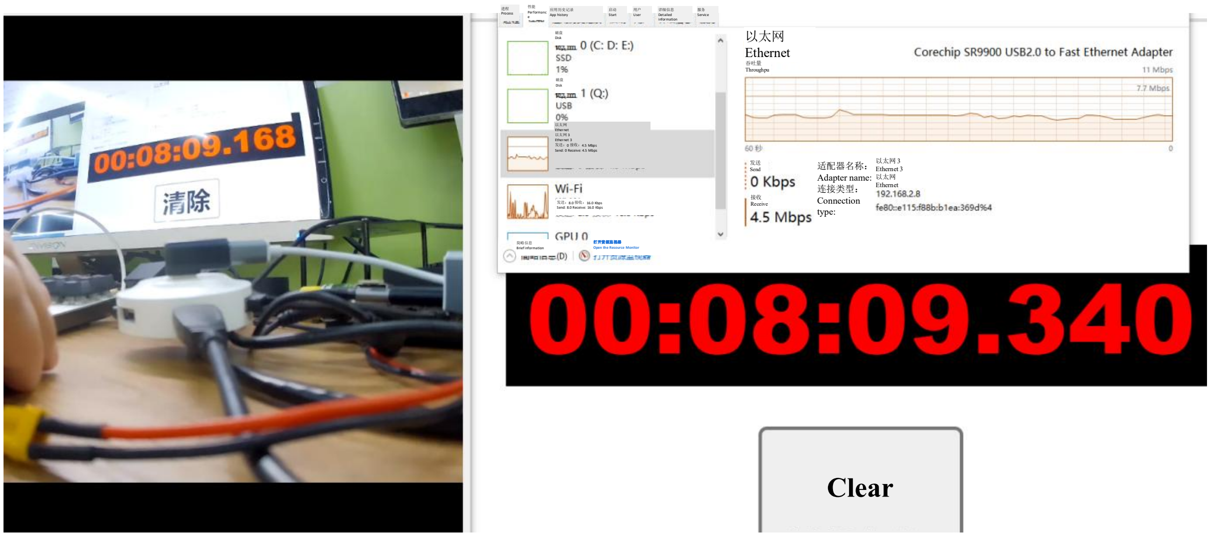
8. SDK:

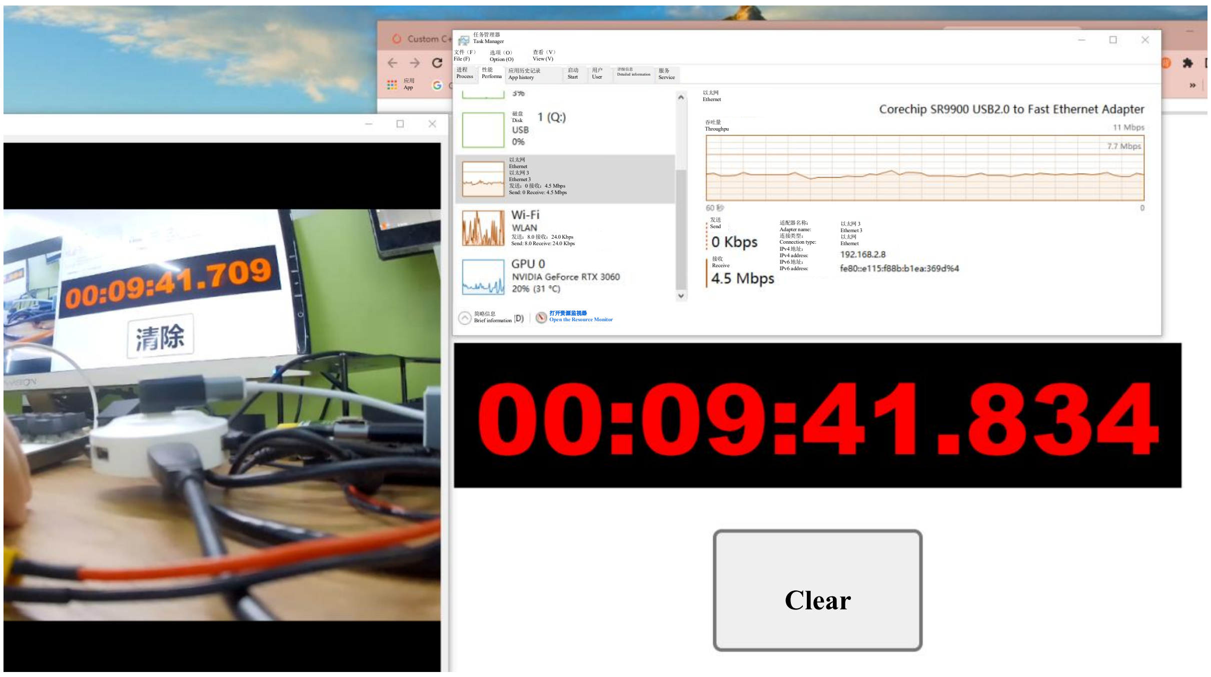
- Linux SDK (GCC 7.5.0) Gimbal Camera C++ SDK: https://gitee.com/amovlab/gimbal-sdk.git
- Gimbal Camera ROS SDK (GCC 7.5.0):https://gitee.com/amovlab/gimbal-sdk-ros.git
-

9. Tools:

- Name: Amov GimbalStudio (Ubuntu version and Window version)
- Download address: AmovGimbalStudio

10. Video stream delay test:





- Test conditions: Use hardware decoding and use the gimbal camera to capture the timer on the monitor. Then the image captured by the gimbal camera will be displayed on the same monitor. In the above screenshot, the left side is the image captured by the gimbal camera, and the right side is the image of the timer. The video stream delay can be calculated by comparing the time difference.
- Delay range: It can be seen that video delay (camera to monitor) is between 150-200 ms at a resolution of 720P and a bit rate of 4.7Mbps

11. Recommended data link (video&data transmission):

- Homer data link Manual
- MINI Homer data link Manual

12. Recommended onboard computer (computer image-processing boards)

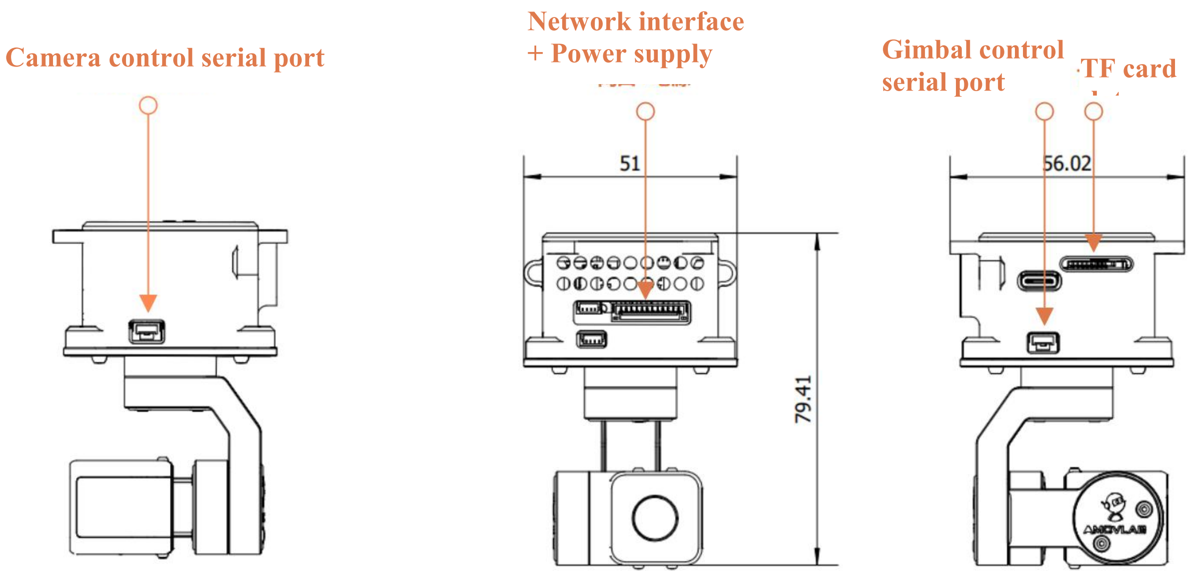
- All Spark onboard computer Manual

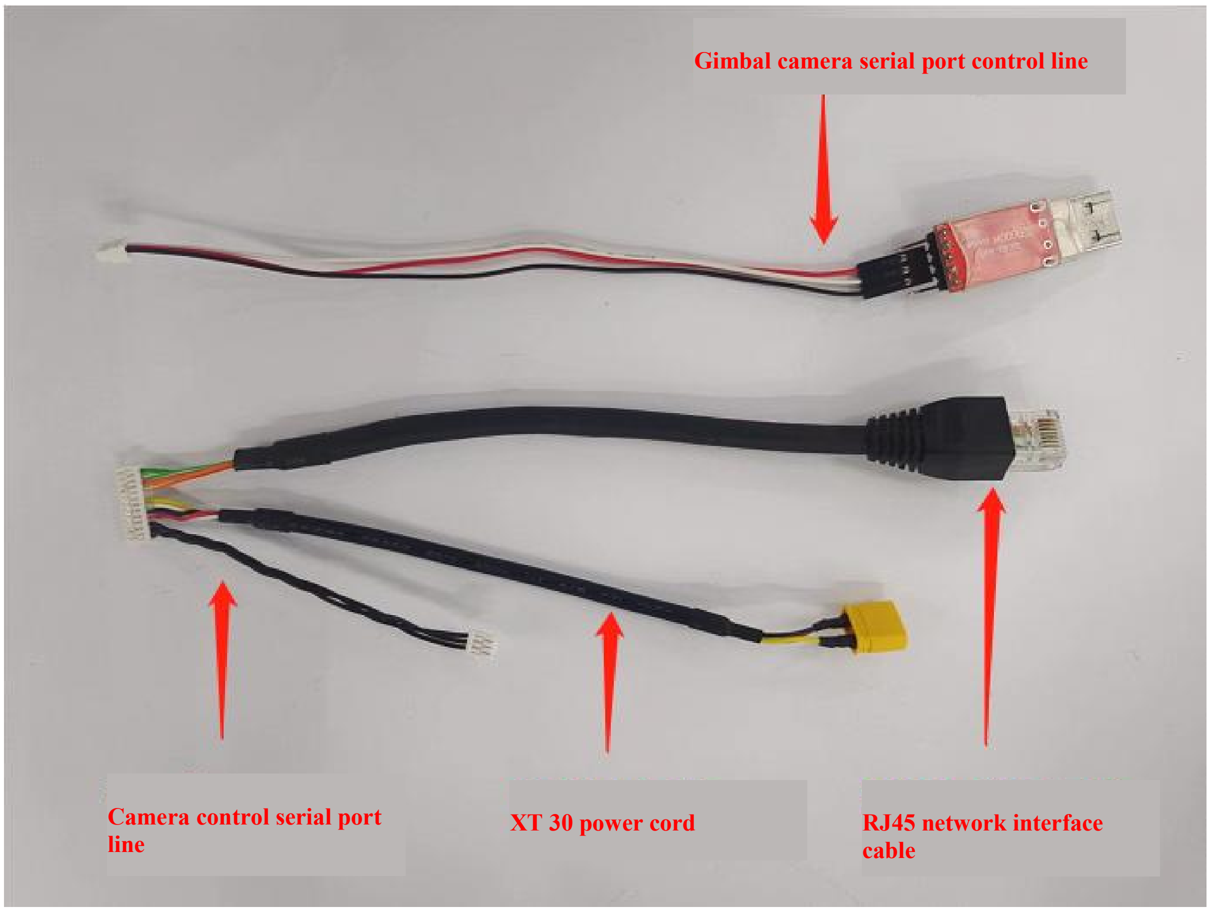
QUICK START OF AMOVLAB G1 GIMBAL CAMERA

QUICK START:

- Power supply: Gimbal camera supply voltage of 12$_{16.8}$v$_{\text{by}}$ 3s or 4s battery in general
- Video stream IP address: rtsp://192.168.2.64

Interface description





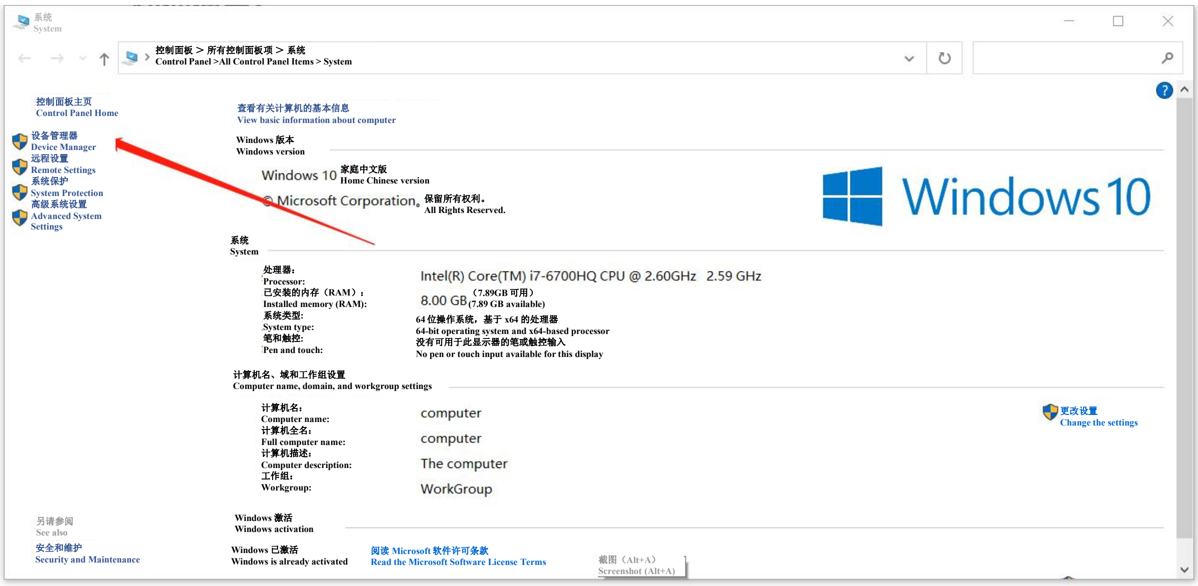
Gimbal camera serial port control line
Camera control serial port line
XT 30 power cord
RJ45 network interface cable

Gimbal Camera Start

- Install the gimbal camera on a stand or aircraft
- Connect the gimbal camera to the power supply, and quietly wait for self-test of the gimbal camera after hearing the sound of "tick tick tick-tick"(axis verification order: roll axis, pitch axis and yaw axis, and the self-test takes about 20s). Do not touch the gimbal camera during self-test, and do not let it in a shaking state, or the self-test cannot be completed.
- After self-test of the gimbal camera, the camera will remain level with the ground and have a stabilization effect.

Connect to the gimbal camera

Serial port connection on Windows



系统
System

控制面板 > 所有控制面板项 > 系统
Control Panel >All Control Panel Items > System

控制面板主页
Control Panel Home

设备管理器
Device Manager
远程设置
Remote Settings
系统保护
System Protection
高级系统设置
Advanced System Settings

查看有关计算机的基本信息
View basic information about computer

Windows 版本
Windows version

Windows 10 家庭中文版
Home Chinese version

© Microsoft Corporation。保留所有权利。
All Rights Reserved.

系统
System

处理器: Processor:	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.59 GHz
已安装的内存（RAM）： Installed memory (RAM):	8.00 GB（7.89GB 可用） (7.89 GB available)
系统类型: System type:	64 位操作系统，基于 x64 的处理器 64-bit operating system and x64-based processor
笔和触控: Pen and touch:	没有可用于此显示器的笔或触控输入 No pen or touch input available for this display

计算机名、域和工作组设置
Computer name, domain, and workgroup settings

计算机名: Computer name:	computer
计算机全名: Full computer name:	computer
计算机描述: Computer description:	The computer
工作组: Workgroup:	WorkGroup

更改设置
Change the settings

另请参阅
See also
安全和维护
Security and Maintenance

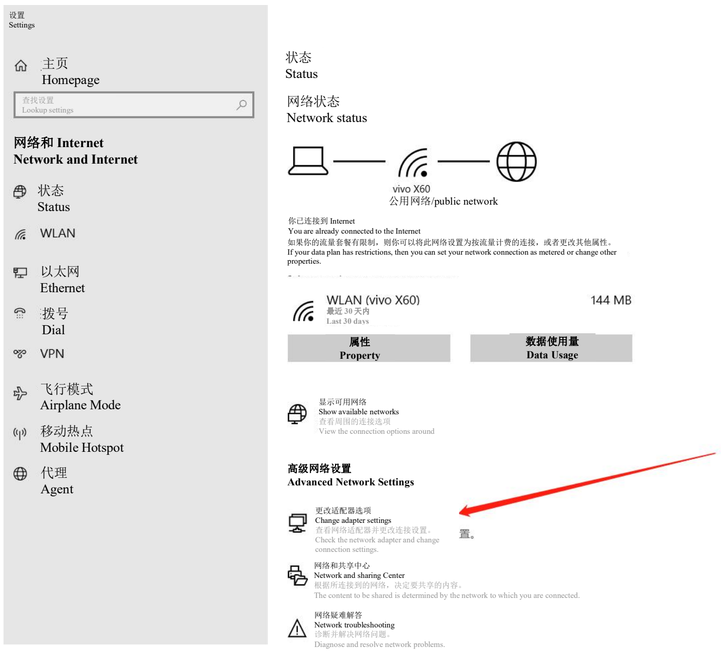
Windows 激活
Windows activation

Windows 已激活
Windows is already activated

阅读 Microsoft 软件许可条款
Read the Microsoft Software License Terms

- Plug the USB-TTL module into the USB port of the computer, and open the Device Manager → the serial port number can be seen.
- In case of lack of drives of the computer, please download from https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers and install them

Network port connection on Windows


设置
Settings

主页
Homepage

查找设置
Lookup settings

网络和 Internet
Network and Internet

状态
Status

WLAN

以太网
Ethernet

拨号
Dial

VPN

飞行模式
Airplane Mode

移动热点
Mobile Hotspot

代理
Agent

状态
Status

网络状态
Network status

vivo X60
公用网络/public network

你已连接到 Internet
You are already connected to the Internet
如果你的流量套餐有限制，则你可以将此网络设置为按流量计费的连接，或者更改其他属性。
If your data plan has restrictions, then you can set your network connection as metered or change other properties.

WLAN (vivo X60) 144 MB
最近 30 天内
Last 30 days

属性
Property

数据使用量
Data Usage

显示可用网络
Show available networks
查看周围的连接选项
View the connection options around

高级网络设置
Advanced Network Settings

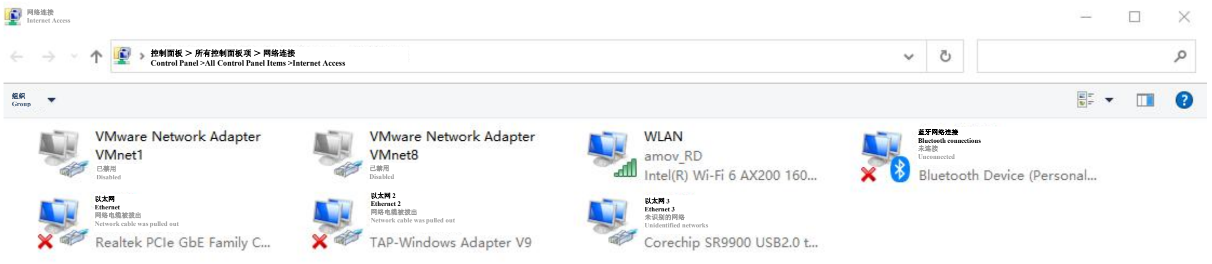
更改适配器选项
Change adapter settings
查看网络适配器并更改连接设置。
Check the network adapter and change connection settings.

置。

网络和共享中心
Network and sharing Center
根据所连接到的网络，决定要共享的内容。
The content to be shared is determined by the network to which you are connected.

网络疑难解答
Network troubleshooting
诊断并解决网络问题。
Diagnose and resolve network problems.

- Open Settings → click Ethernet → click Change Adapter Settings → find the corresponding Network


网络连接
Internet Access

控制面板 > 所有控制面板项 > 网络连接
Control Panel >All Control Panel Items >Internet Access

组织
Group

VMware Network Adapter VMnet1
已禁用
Disabled

VMware Network Adapter VMnet8
已禁用
Disabled

WLAN
amov_RD
Intel(R) Wi-Fi 6 AX200 160...

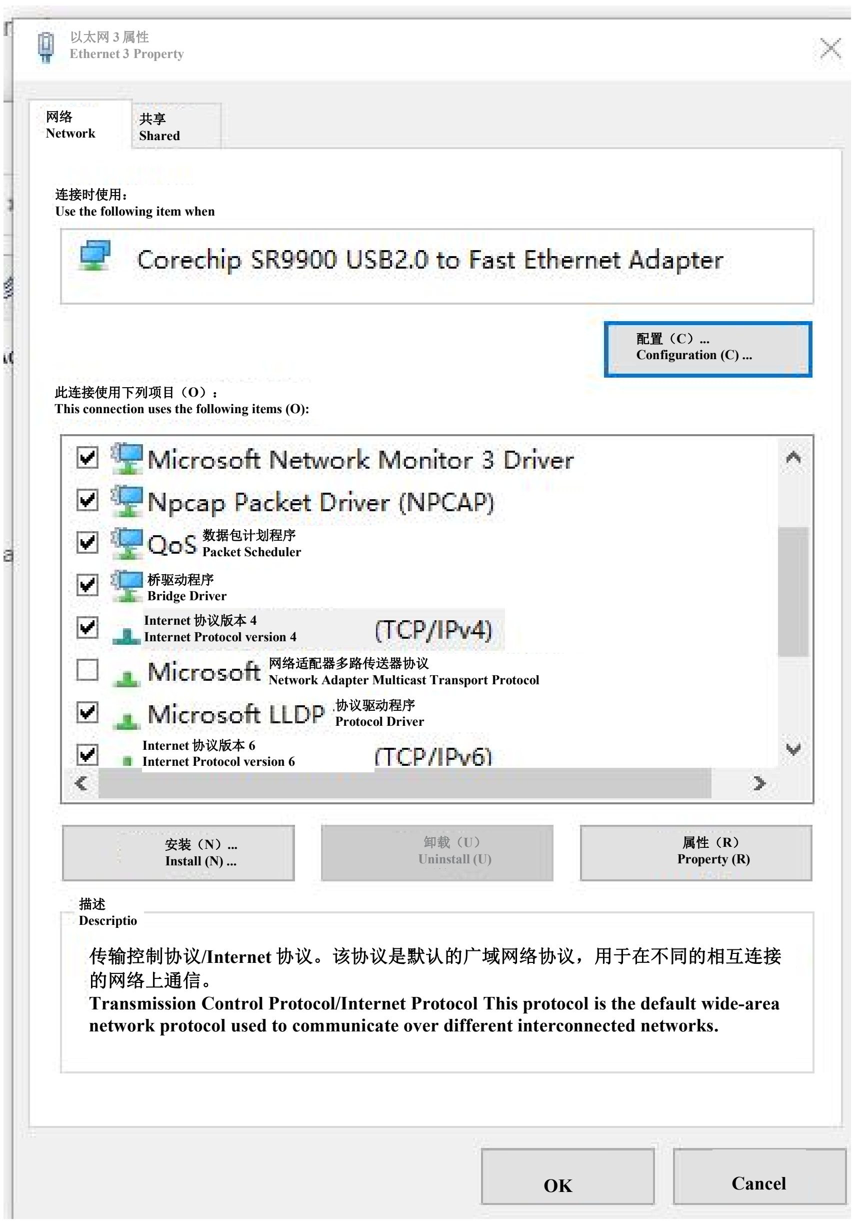
蓝牙网络连接
Bluetooth connections
未连接
Unconnected
Bluetooth Device (Personal...

以太网
Ethernet
网络电缆被拔出
Network cable was pulled out
Realtek PCIe GbE Family C...

以太网 2
Ethernet 2
网络电缆被拔出
Network cable was pulled out
TAP-Windows Adapter V9

以太网 3
Ethernet 3
未识别的网络
Unidentified networks
Corechip SR9900 USB2.0 t...

- Open Network Property-> click Internet Protocol version 4


以太网 3 属性
Ethernet 3 Property

网络 Network | 共享 Shared

连接时使用：
Use the following item when

Corechip SR9900 USB2.0 to Fast Ethernet Adapter

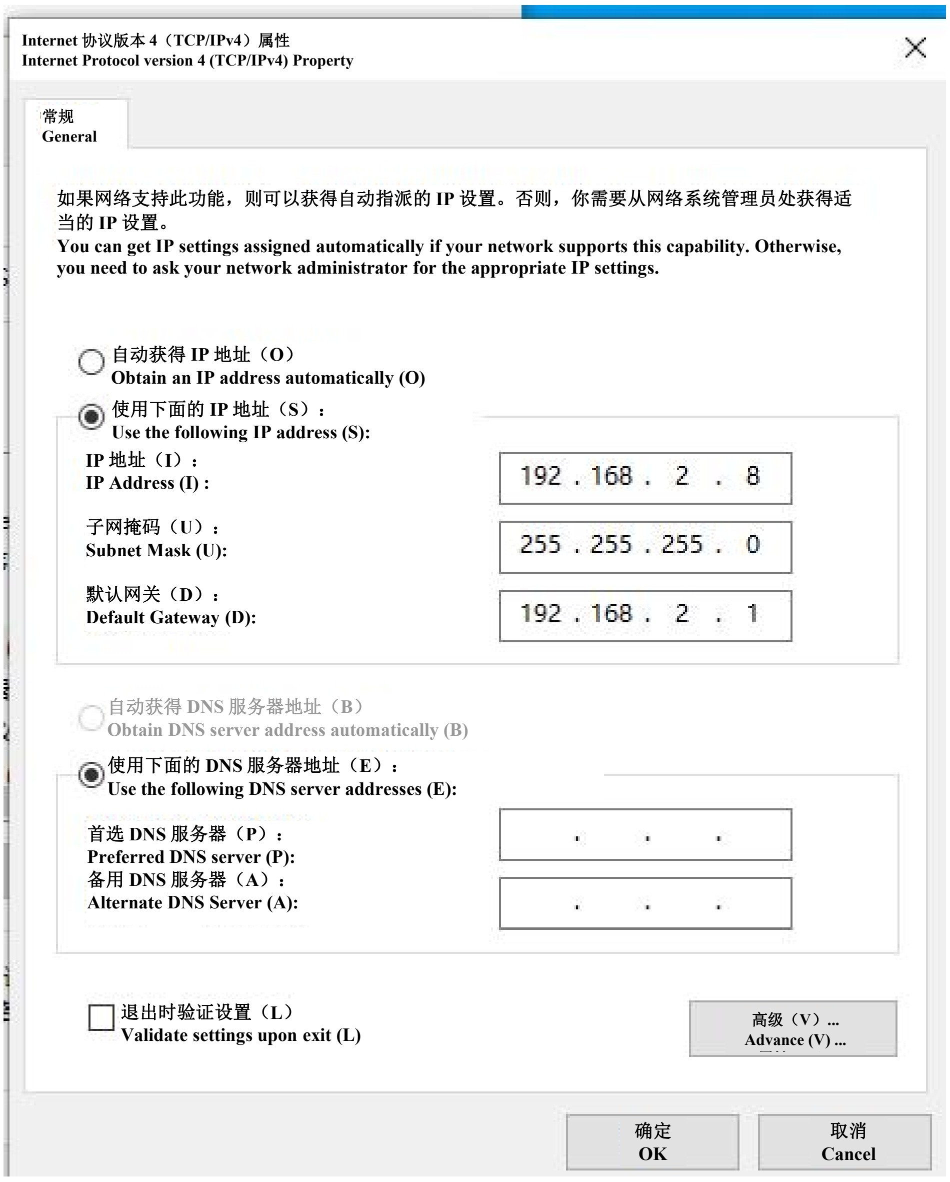
此连接使用下列项目（O）：
This connection uses the following items (O):

- [x] Microsoft Network Monitor 3 Driver
- [x] Npcap Packet Driver (NPCAP)
- [x] QoS 数据包计划程序 Packet Scheduler
- [x] 桥驱动程序 Bridge Driver
- [x] Internet 协议版本 4 Internet Protocol version 4 (TCP/IPv4)
- [ ] Microsoft 网络适配器多路传送器协议 Network Adapter Multicast Transport Protocol
- [x] Microsoft LLDP 协议驱动程序 Protocol Driver
- [x] Internet 协议版本 6 Internet Protocol version 6 (TCP/IPv6)

描述
Descriptio

传输控制协议/Internet 协议。该协议是默认的广域网络协议，用于在不同的相互连接的网络上通信。
Transmission Control Protocol/Internet Protocol This protocol is the default wide-area network protocol used to communicate over different interconnected networks.

- Check Use the Following IP Address → enter IP Address, Subnet Mask and Default Gateway as shown in the following figure, and click OK



Internet 协议版本 4（TCP/IPv4）属性
Internet Protocol version 4 (TCP/IPv4) Property

常规
General

如果网络支持此功能，则可以获得自动指派的 IP 设置。否则，你需要从网络系统管理员处获得适当的 IP 设置。
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

自动获得 IP 地址（O）
Obtain an IP address automatically (O)

[✓ 使用下面的 IP 地址（S）：
Use the following IP address (S):]

IP 地址（I）： IP Address (I) :	192 . 168 . 2 . 8
子网掩码（U）： Subnet Mask (U):	255 . 255 . 255 . 0
默认网关（D）： Default Gateway (D):	192 . 168 . 2 . 1

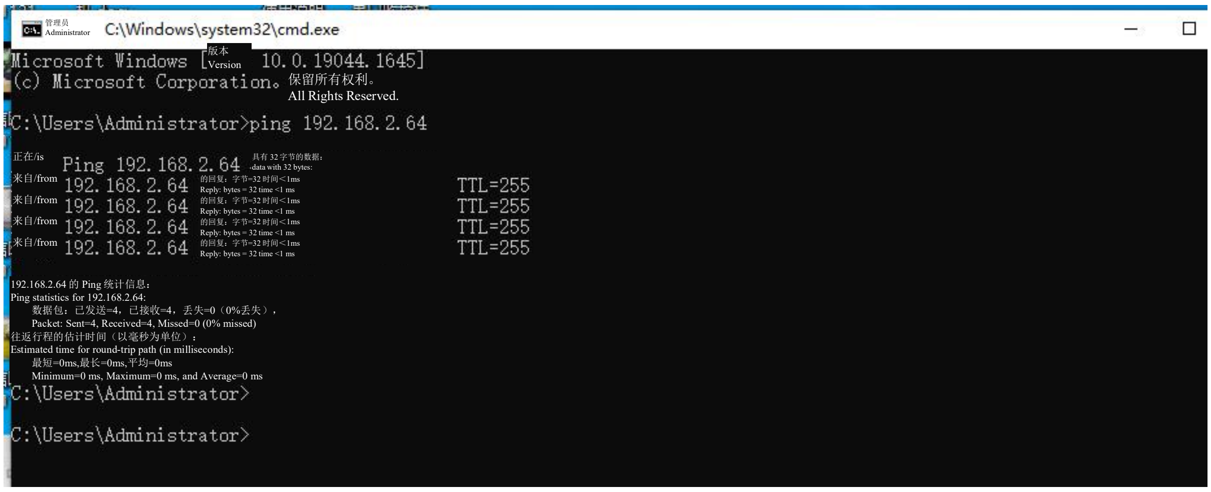
自动获得 DNS 服务器地址（B）
Obtain DNS server address automatically (B)

[✓ 使用下面的 DNS 服务器地址（E）：
Use the following DNS server addresses (E):]

首选 DNS 服务器（P）： Preferred DNS server (P):	. . .
备用 DNS 服务器（A）： Alternate DNS Server (A):	. . .

退出时验证设置（L）
Validate settings upon exit (L)

- Go back to the homepage of the PC, use the shortcut "WIN+R" to open the operating procedure, enter cmd to open the terminal, and enter ping 192.168.2.64, to ensure that the gimbal camera is successfully connected.



```
Microsoft Windows [版本 Version 10.0.19044.1645]
(c) Microsoft Corporation。保留所有权利。
All Rights Reserved.

C:\Users\Administrator>ping 192.168.2.64

正在/is Ping 192.168.2.64 具有 32 字节的数据: data with 32 bytes:
来自/from 192.168.2.64 的回复: 字节=32 时间<1ms Reply: bytes = 32 time <1 ms   TTL=255
来自/from 192.168.2.64 的回复: 字节=32 时间<1ms Reply: bytes = 32 time <1 ms   TTL=255
来自/from 192.168.2.64 的回复: 字节=32 时间<1ms Reply: bytes = 32 time <1 ms   TTL=255
来自/from 192.168.2.64 的回复: 字节=32 时间<1ms Reply: bytes = 32 time <1 ms   TTL=255

192.168.2.64 的 Ping 统计信息:
Ping statistics for 192.168.2.64:
    数据包: 已发送=4, 已接收=4, 丢失=0 (0%丢失),
    Packet: Sent=4, Received=4, Missed=0 (0% missed)
往返行程的估计时间(以毫秒为单位):
Estimated time for round-trip path (in milliseconds):
    最短=0ms,最长=0ms,平均=0ms
    Minimum=0 ms, Maximum=0 ms, and Average=0 ms
C:\Users\Administrator>

C:\Users\Administrator>
```

- Next, open the AmovGimbalStudio software

Modify the video stream IP address:

- **Be sure to use a U3 high-speed TF card (SanDisk and Samsung are recommended)**
- The main role of the gimbal camera configuration file: configuring the streaming address
- Configuration file download address: 吊舱配置文件 (amovlab.com)
- Modify the gimbal camera IP address: open the configuration file → the following code can be seen at the bottom of the file → just modify the net_Ip and net_Gateway.

```
;In wifi STA mode or Ethernet mode,if select static IP ,Configure the staic IP and gateway of the network,
net_Ip=192.168.2.64
net_Gateway=192.168.2.1
```

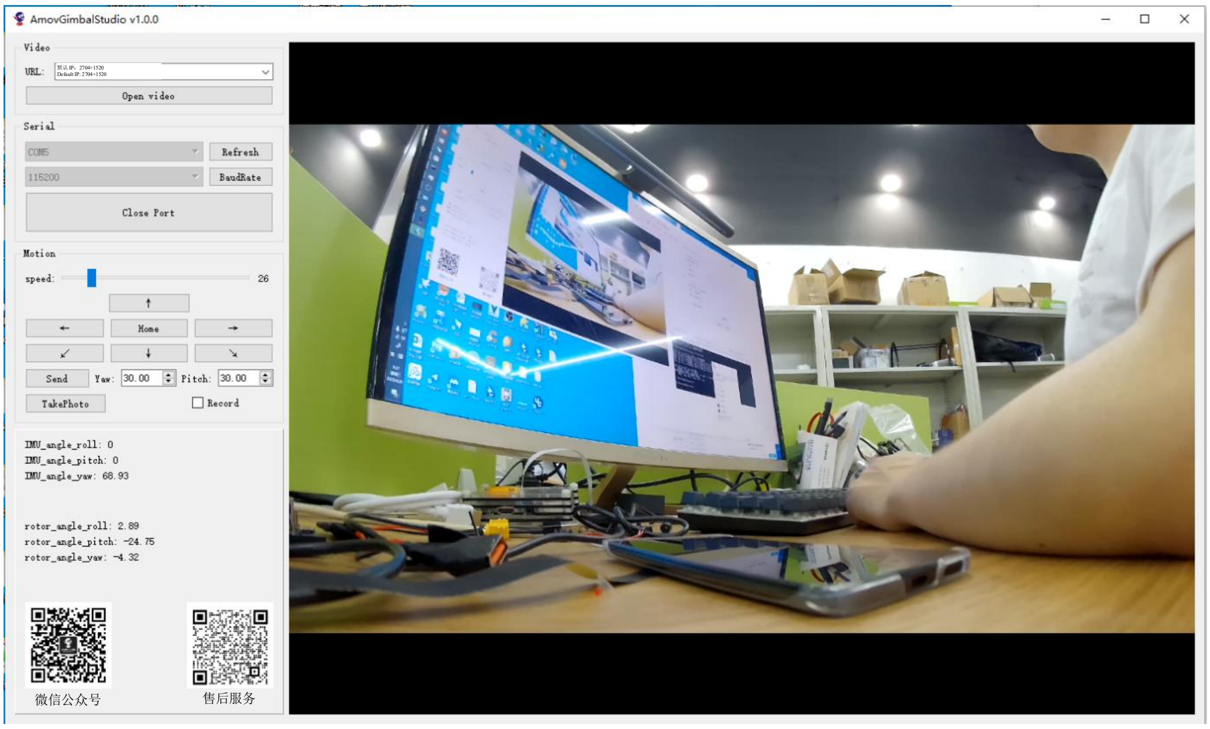
- Modify image orientation:

```
;Video Image Rotation
;Available values:   [0]:  Normal
;                    [1]:  Vertical
;                    [2]:  Level     / (screen upside down)
;Default value: 0
Rotation=0
```

Amov Gimbal Studio

1. Introduction

It is the control software for AMOVLAB Gimbal Camera, which supports Windows and Ubuntu systems.



URL: 默认IP：2704×1520 Default IP: 2704×1520

COM5

115200

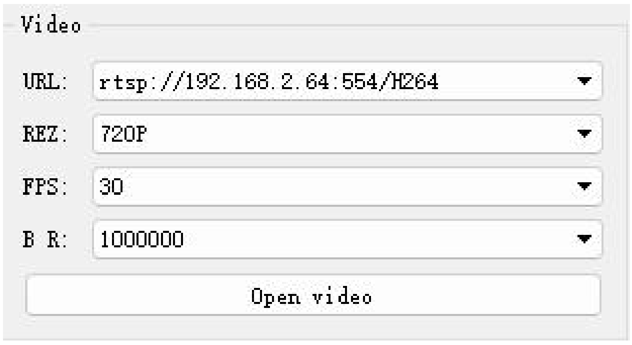
speed: 26

Yaw: 30.00 Pitch: 30.00

IMU_angle_roll: 0
IMU_angle_pitch: 0
IMU_angle_yaw: 68.93

rotor_angle_roll: 2.89
rotor_angle_pitch: -24.75
rotor_angle_yaw: -4.32

2. IMAGE ACQUISITION

- The following default address exists in the URL drop-down box: rtsp://192.168.2.64:554/H264
- Video stream address parameters interpretation: REZ: Resolution; FPS: Frames Per Second; BR: Bit Rate;
- Click Open video button to read the image data of the gimbal camera after selecting Complete.
- Tip: It is recommended that the software be run in plain English path, otherwise it may not run successfully.
- 

URL: rtsp://192.168.2.64:554/H264

REZ: 720P

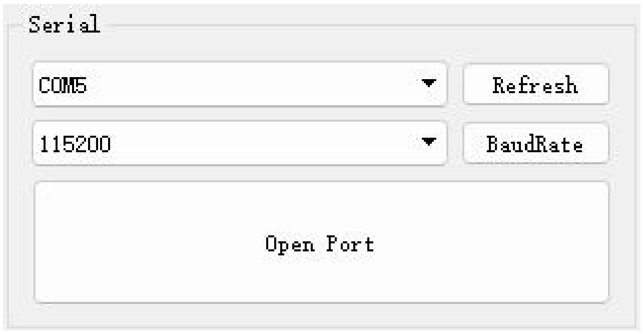
FPS: 30

B R: 1000000

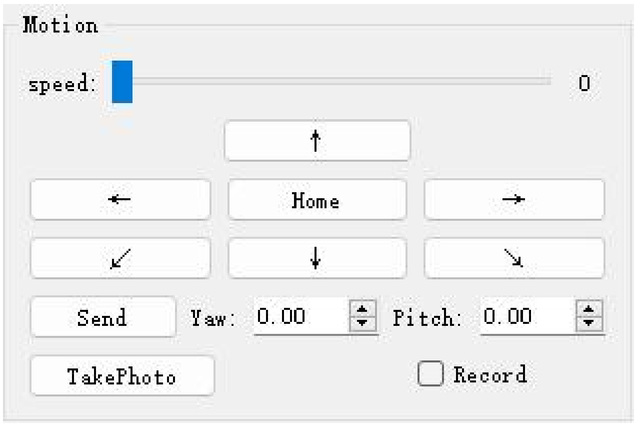
3. CONTROL THE GIMBAL CAMERA

1. Open the serial port

- Click Refresh to update the serial port and select the Gimbal Camera serial port
- Select the baud rate of 115200

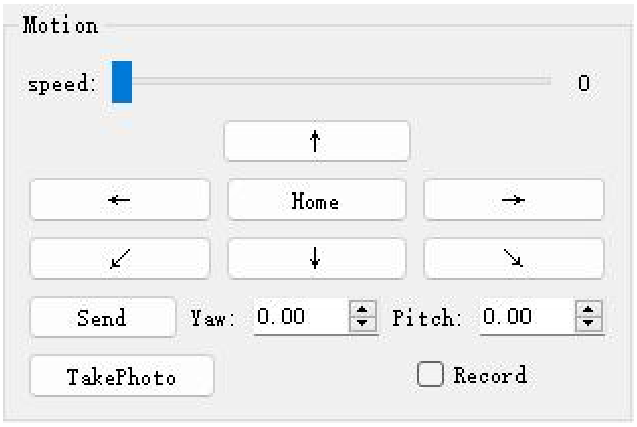
- Click Open Port button to open the serial port
- 

2. Control the gimbal camera by speed

- Slide Speed button to adjust the gimbal camera speed (the speed setting cannot be 0 when controlling the angle)
- Click the up, down, left and right icon buttons to control the cimbal camera
- When clicked the Home button, the gimbal camera returns to center position.
- 

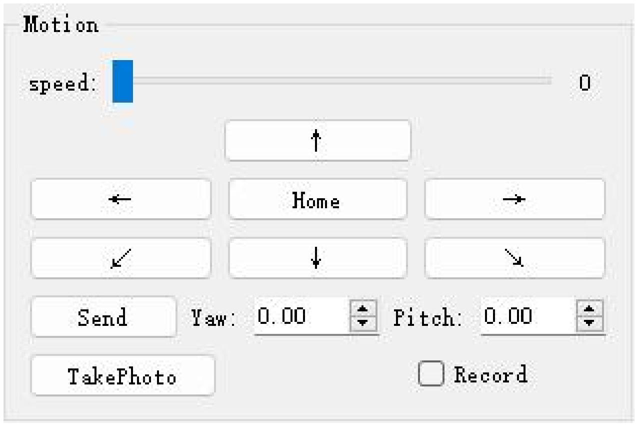
3. Control the cimbal camera by angle

- Slide Speed button to adjust the gimbal camera speed (the speed setting cannot be 0 when controlling the angle)
- Type an angle in the Yaw input box
- Type an angle in the Pitch input box
- And then click Send button

- 

4. Photo and record functions

- Click TakePhoto button to take a photo
- Check Record box to record (with a "tick" sound), and uncheck Record box to stop recording (with a "tick-tick" sound)
- Photos and videos are saved in TF card, and **be sure to use a U3 high-speed TF card (SanDisk and Samsung are recommended)**
- The photo function is only supported in low-resolution and low bitrate RTSP video stream



4. Download address

- **Windows version (AmovGimbalStudio.exe)**
- **Ubuntu x86 version (AmovGimbalStudio.AppImage)**
- **Ubuntu arm64 version (AmovGimbalStudio_arm64.AppImage)**

Download address

Note: Ubuntu version requires the Gstreamer to be installed for dependency (if it is already installed, you don't need to install it again):

sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgstreamer-plugins-bad1.0-dev gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly gstreamer1.0-libav gstreamer1.0-doc gstreamer1.0-tools gstreamer1.0-x gstreamer1.0-alsa gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-qt5 gstreamer1.0-pulseaudio

The dependency may cannot be installed due to network reasons: please download the file manually for installation: Download address

Manual installation method: sudo dpkg -i xxx.deb

AMOVLAB G1 Gimbal Camera SDK (Linx C++)

Introduction

- The AMOVLAB G1 Gimbal Camera is an optical gimbal camera with high performance and low cost
- Smart gimbal camera = gimbal + camera + AI chip + human - computer interaction software + deep learning
- Linux SDK (GCC 7.5.0)

Download SDK:

Clone repository

```
git clone https://gitee.com/amovlab/gimbal-sdk.git
```

Run the sample:

1. Access the repository and use the following command to compile the SDK

```
mkdir build
cd build
cmake ..
make
```

2. `GetGimbalState`, `GimbalSpeedControl`, `GimbalAngleAndSpeedControl` and other executable files can be seen in the build folder

3. Connect the serial port, and make sure there is already a serial port `ls /dev/ttyUSB*` with the command ls /dev/ttyUSB*

4. Methods for running the sample:

- Obtain status data of the Gimbal Camera, such as IMU angle and encoder angle

```
./GetGimbalState -S /dev/ttyUSB0 -b 115200
```

- Control the rotation speed of the Gimbal Camera. The following example is to control with roll: 10°/s, pitch: 10°/s, and yaw: 10°/s

```
 ./GimbalAngleRateControl -S /dev/ttyUSB0 -b 115200 -r 10 -p 10 -y 10
```

- Control the rotation angle of the Gimbal Camera and set the rotation speed to 10°/s. The following example is to control by turning twit roll: 10°, pitch: 10°, and yaw: 10°

```
./GimbalAngleControl -S /dev/ttyUSB0 -b 115200 -r 10 -p 10 -y 10 -rate 10
```

- The recorded video will be saved to the TF card

```
./CameraControl -a 1 #Record and stop recording
./CameraControl -a 2 #Take photos
```

Contact us

- AMOVLAB official website: https://www.amovlab.com/
- AMOVLAB forum: https://bbs.amovlab.com/

Acquire Image (Linux)

1. Example of using OpenCV to acquire an RTSP video stream:

Use JETSON NX to acquire RTSP video stream (hardware decoding):

Environmental requirements:

1、Jetson series hardware

2、OpenCV (have been compiled with gstreamer)

Gstreamer

3、C++ environment

rtsp_capture.cpp

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc.hpp>
int main()
{
    std::string pipline_str = "rtspsrc
location=rtsp://192.168.2.64:/H264?W=1920&H=1080&FPS=30&BR=4900000 latency=100 \
        caps='application/x-rtp,media=(string)video,clock-rate=(int)90000,encoding-name=\
        (string)H264,width=1920,height=1080,framerate=30/1' !\
        rtph264depay ! h264parse ! omxh264dec ! nvvidconv ! \
        video/x-raw, width=(int)1280, height=(int)720, format=(string)BGRx ! \
        videoconvert ! appsink sync=false";

    cv::VideoCapture capture;
    cv::Mat frame;
    capture.open(pipline_str);
    if (!capture.isOpened())
    {
        std::cout << "Can not open web camera !" << std::endl;
        return -1;
    }

    while (1)
    {
        capture.read(frame);
        if (frame.empty())
        {
            break;
        }
        cv::imshow("video", frame);
        cv::waitKey(20);
    }
    capture.release();
    return 0;
}
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc.hpp>

int main()
{

    std::string pipline_str = "rtspsrc 
location=rtsp://192.168.2.64:/H264?W=1920&H=1080&FPS=30&BR=4900000 latency=100 \
            caps='application/x-rtp,media=(string)video,clock-rate=(int)90000,encoding-name=\
            (string)H264,width=1920,height=1080,framerate=30/1' !\
            rtph264depay ! h264parse ! omxh264dec ! nvvidconv ! \
            video/x-raw, width=(int)1280, height=(int)720, format=(string)BGRx ! \
            videoconvert ! appsink sync=false";
    cv::VideoCapture capture;
    cv::Mat frame;
    capture.open(pipline_str);

    if (!capture.isOpened())
    {
        std::cout << "Can not open web camera !" << std::endl;
        return -1;
    }

    while (1)
    {
        capture.read(frame);
        if (frame.empty())
        {
            break;
        }
        cv::imshow("video", frame);
        cv::waitKey(20);
    }
    capture.release();
    return 0;
}
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 2.6)

project(rtsp_capture)

find_package(OpenCV REQUIRED)
add_definitions(-std=c++11)

include_directories(
    ${OpenCV_INCLUDE_DIRS}
)

add_executable(rtsp_capture src/rtsp_capture.cpp)
target_link_libraries(rtsp_capture  ${OpenCV_LIBS})
cmake_minimum_required(VERSION 2.6)

project(rtsp_capture)

find_package(OpenCV REQUIRED)
add_definitions(-std=c++11)

include_directories(
   ${OpenCV_INCLUDE_DIRS}
)

add_executable(rtsp_capture src/rtsp_capture.cpp)
target_link_libraries(rtsp_capture  ${OpenCV_LIBS})
```

Acquire 720P video stream:

```
std::string pipline_str="rtsp://192.168.2.64:554/H264?W=1280&H=720&BR=10000000&FPS=30";
```

Acquire 1080P video stream:

```
std::string pipline_str="rtsp://192.168.2.64:554/H264?W=1920&H=1080&BR=10000000&FPS=30"
```

Acquire 2.7K video stream:

```
std::string pipline_str="rtsp://192.168.2.64:554/H264?W=2704&H=1520&BR=10000000&FPS=30"
```

Acquire video stream below 720p:

- Modify nvvidconv ! video/x-raw, width=(int)640, height=(int)360, format=(string)BGRx

```
   std::string pipline_str = "rtspsrc
location=rtsp://192.168.2.64:/H264?W=1920&H=1080&FPS=30&BR=4900000 latency=100 \
         caps='application/x-rtp,media=(string)video,clock-rate=(int)90000,encoding-name=\
         (string)H264,width=1920,height=1080,framerate=30/1' !\
         rtph264depay ! h264parse ! omxh264dec ! nvvidconv ! \
         video/x-raw, width=(int)640, height=(int)360, format=(string)BGRx ! \
         videoconvert ! appsink sync=false";
std::string pipline_str = "rtspsrc
location=rtsp://192.168.2.64:/H264?W=1920&H=1080&FPS=30&BR=4900000 latency=100 \
         caps='application/x-rtp,media=(string)video,clock-rate=(int)90000,encoding-name=\
         (string)H264,width=1920,height=1080,framerate=30/1' !\
         rtph264depay ! h264parse ! omxh264dec ! nvvidconv ! \
         video/x-raw, width=(int)640, height=(int)360, format=(string)BGRx ! \
         videoconvert ! appsink sync=false";
```

2. Frequently asked questions on image acquisition:

Why can't the gimbal camera acquire the image?

1、Is it possible to PING the gimbal camera? If it can open, it means the network communication is normal

2、Is it possible to open a video stream using AmovGimbalStudio? If it can open, it means the Gstreamer environment is normal

3、Is it possible to open a video stream using OpenCV? If it cannot be opened, it means that the with gstreamer option was not checked when OpenCV was compiled, and it is recommended to recompile OpenCV.

Why does my Jetson platform delay higher?

1、It is possible that NVENC, the hardware decoder of Jetson, was not activated.

2、Enter jtop command and see if NVENC is called.

3、If it is displayed as OFF, you need to use the longer pipline_str above

Why is the image delay particularly high when the DEMO runs with my own code?

1、I have used hardware decoding, but there is a significant delay when using this code with my own code

2、It is suggested to open a thread for the function of acquiring the image of the gimbal camera and use multithreading to solve this problem

AMOVLAB G1 Gimbal Camera ROS SDK

Introduction

1、The AMOVLAB G1 Gimbal Camera is an optical gimbal camera with high performance and low cost

2、Smart gimbal camera = gimbal + camera + AI chip + human - computer interaction software + deep learning

ROS SDK(GCC 7.5.0)

DOWNLOAD SDK:

Clone repository

git clone https://gitee.com/amovlab/gimbal-sdk-ros.git

ENVIRONMENT

We have completed the test in the following system environments:

1、Hardware platform: Allspark (core board: Jetson NX)

2、Operating system: Ubuntu 18.04

3、C++ version: C++11

4、ROS version: melodic

5、OpenCV：3.3.1

BUILD AMOV GIMBAL SDK

Follow the commands below to create and compile the SDK if you don't have a workspace.

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
git clone https://gitee.com/amovlab/gimbal-sdk-ros.git
cd gimbal-sdk-ros
git submodule update --init  #update the submodule
cd ../..
catkin_make/ compile SDK
```

RUN THE SAMPLE

1、Access catkin_ws directory first

2、Add the ROS SDK feature package to the environment variables via the source devel/setup.bash command

3、Run the sample:

1. Run the gimbal camera node (control the gimbal camera via the /amov_gimbal_ros/gimbal_control topic, and acquire gimbal camera status information via the /amov_gimbal_ros/gimbal_state topic)

roslaunch amov_gimbal_sdk_ros gimbal_G1.launch

- Control the camera via the /amov_gimbal_ros/set_camera_action service (0: record and stop recording). And the recorded video will be saved to the TF card

2. Acquire camera video image (acquire the ROS image via the /amov_gimbal_ros/gimbal_image topic)

```
roslaunch amov_gimbal_sdk_ros gimbal_image_G1.launch
```

NODE DESCRIPTION

Node	Topic	Service
gimbal_node	/amov_gimbal_ros/gimbal_state	/amov_gimbal_ros/set_camera_action
	/amov_gimbal_ros/gimbal_control	
camera_node	/amov_gimbal_ros/amov_camera_image	

CONTACT US

-
- AMOVLAB official website: https://www.amovlab.com/
- AMOVLAB forum: https://bbs.amovlab.com/

AMOVLAB G1 GIMBAL CAMERA SDK (PYTHON SDK)

INTRODUCTION

- The AMOVLAB G1 Gimbal Camera is an optical gimbal camera with high performance and low cost
- Smart gimbal camera = gimbal + camera + AI chip + human - computer interaction software + deep learning
- Python SDK(Python 3)

INSTALL DEPENDENCIES

```
pip3 install pytest
```

DOWNLOAD SDK:

Clone repository

```
git clone --recursive https://gitee.com/amovlab/gimbal-sdk-python.git
```

RUN THE SAMPLE:

1. Use the following command for SDK compilation

```
pip3 install ./gimbal-sdk-python
```

2. amov_gimbal_python.py file can be seen in the gimbal-sdk-python folder

3. Connect the serial port, and make sure there is already a serial port /dev/ttyUSB* with the command ls /dev/ttyUSB*

4. Methods for running the sample:

- Obtain status data of the Gimbal Camera, such as IMU angle and encoder angle

  ```
  python3 amov_gimbal_python.py
  ```

- Input 1 Take photos

- Input 2 Start and stop recording
- Input 3 Angle control (note: angle rate needs to be specified)
- Input 4 Angular rate control
- Input 5 Reset
- Input 6 Acquire status data of the gimbal camera, such as IMU angle and encoder angle

5. Acquire the RTSP video stream of gimbal camera (hardware decoding on the Jetson NX platform)
   - Use the following command to view the image of the gimbal camera

     ```
     python3 amov_gimbal_image.py
     ```

CONTACT US

- AMOVLAB official website: https://www.amovlab.com/
- AMOVLAB forum: https://bbs.amovlab.com/

G1 GIMBAL CAMERA DEVELOPERS KIT - INTRODUCTION

The hardware composition of the G1 Gimbal Camera developer kit generally consists of G1 Gimbal Camera and AllSpark onboard computer. In terms of software, there are two gimbal camera visual tracking algorithms: KCF boxed target tracking algorithm and yoloV5+DeepSort multi-target tracking algorithm.

Hardware composition of G1 Gimbal Camera developers kit:

- G1 Gimbal Camera
- AllSpark onboard computer

G1 Gimbal Camera developers kit algorithm:

- KCF boxed target tracking algorithm
- YoloV5+DeepSort multi-target tracking algorithm (TensorRT inference acceleration)

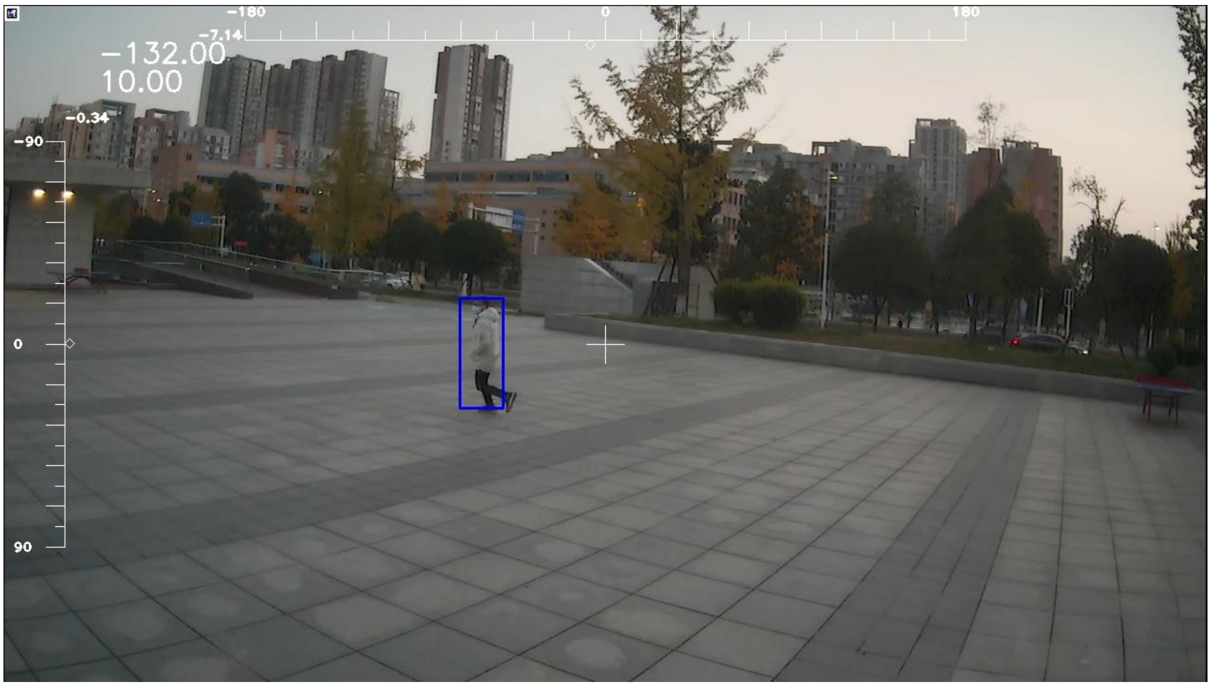
Software version of G1 Gimbal Camera developers kit:

- Ubuntu：18.04
- ROS：melodic
- OpenCV：3.3.1
- TensortRT：7.

ROS software feature package of G1 Gimbal Camera developers kit:

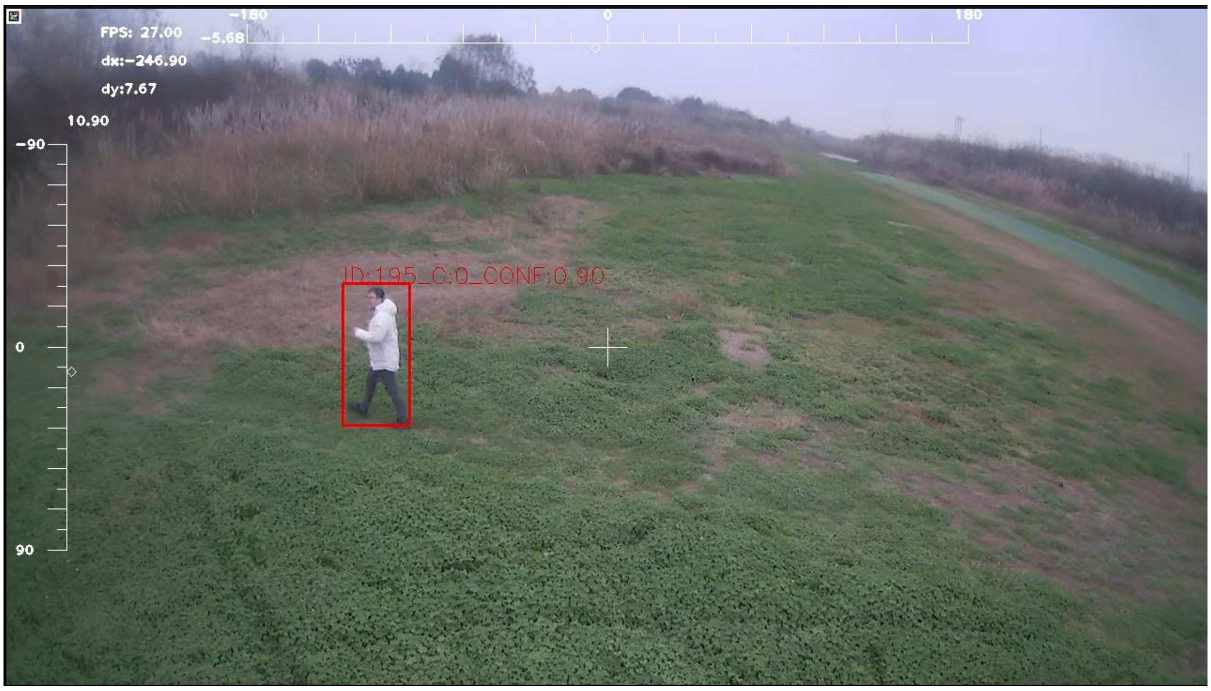
- G1 Gimbal Camera ROS feature package
- G1 Gimbal Camera controller feature package
- Corresponding computer vision algorithms

KCF boxed target tracking algorithm:



KCF is a relatively lightweight tracker with simple deployment, low resource consumption and high portability, and its algorithm frame rate can reach 30FPS, which meets the requirements of UAV for target tracking.

YoloV5+DeepSort multi-target tracking algorithm:



The YoloV5+DeepSort multi-target tracking algorithm feature package already contains the model of the COCO common dataset for YOLOV5 target detection and the model of the DeepSort tracker. The YoloV5+DeepSort multi-target tracking algorithm uses the TensorRT inference framework for acceleration, and its algorithm frame rate can reach 22–35FPS, which meets the requirements of UAV for target tracking. Combined with the high-precision and high-speed response control features of the high-performance G1 Gimbal Camera, the target will be targeted in the center of the field of view.

G1 Gimbal Camera Developers Kit-Frequently Asked Questions

- Why there is no image?
  - It is possible that the onboard computer is not configured with a fixed IP.

> *Note*
> *PDF documents lag behind the online documents, so the online documents will ultimately prevail!*

AMOVLAB

TEL: 028-87872048

Email address: service@amovauto.com

Address: No. 66 Dayu East Road, Qingrong Town, Pidu District, Chengdu City