

# NET Camera SDK Manual

---

## 简介

---

此SDK仅支持私有协议相机，其他协议如RTSP/GIGE等暂不支持

提供控制和取流示例，支持C++/C#/Python调用，支持Windows/Linux平台  
Windows/Linux平台支持x86/x64架构，arm架构等暂不支持

C312/C313/C314/L313/L314/M112/M114/M202/L116均为老版私有协议相机，F801W也为老版私有协议，只是部分控制接口有区别

**C312/C313/C314/L313/L314 1.8.0版本，L116 1.7.0版本开始仅支持新版协议**

其余相机均为新版私有协议相机

有线仅支持MJPEG格式，分辨率帧率列表及各个参数范围参考附录，码率最大为86000

WiFi相机必须使用H264格式，1920x1080分辨率，码率最大为30000

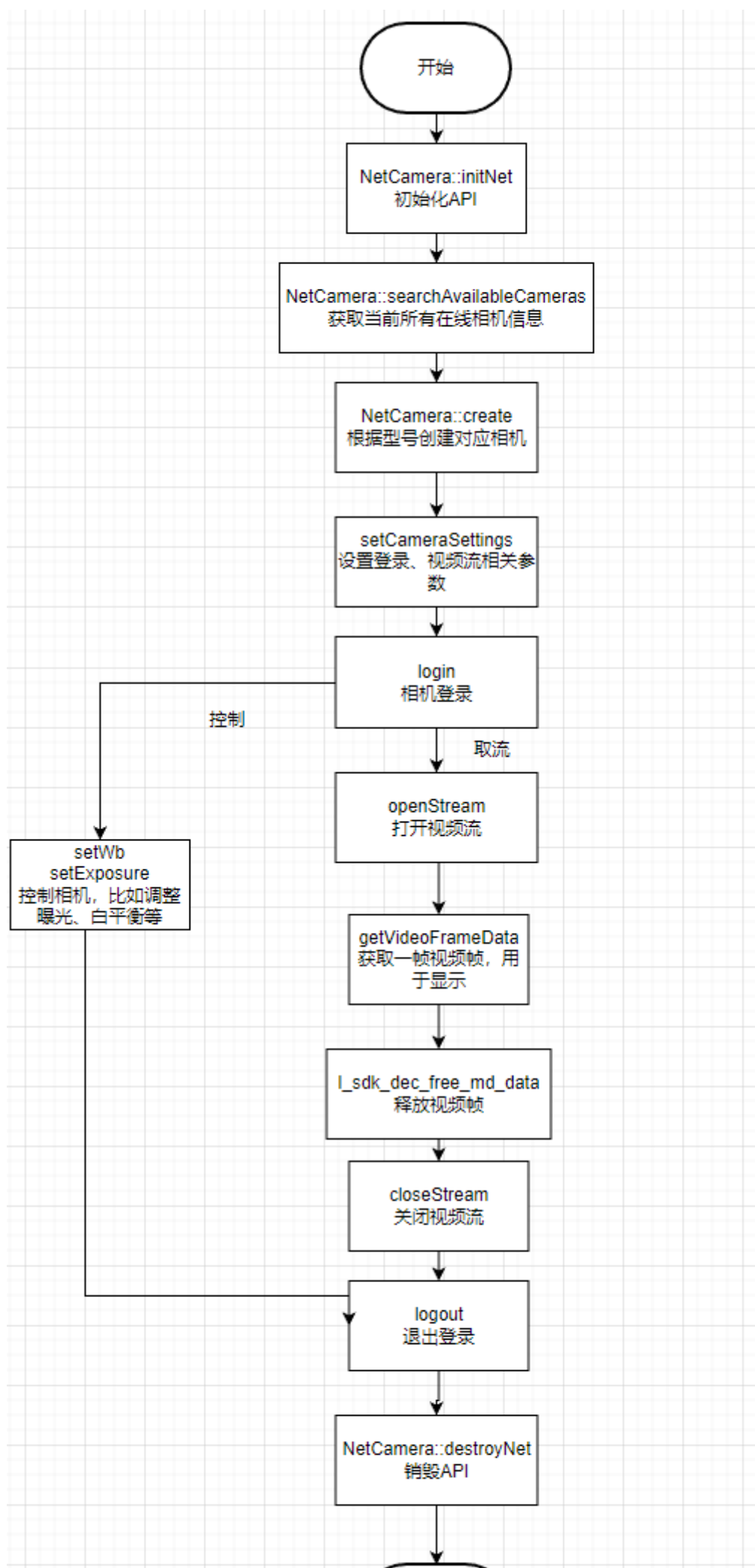
注：其中F801W仅支持H264格式，1920x1080分辨率，帧率最大为25帧，码率最大为10000

SDK仅依赖FFmpeg，版本为n4.1.3

## 快速上手

---

调用流程如下





结束

## 流程说明

上图主要包含了相机控制和获取视频流的流程，两者互不影响，可以单独使用，也可共同使用

## SDK API初始化

首先调用 `initNet` 进行初始化操作，整个程序开始时调用一次，不建议频繁调用

## 枚举相机

调用 `searchAvailableCameras` 获取当前所有在线的相机信息，供用户自行选择

## 创建相机

调用 `create` 根据型号及软件版本号创建对应相机对象

## 设置登录、视频流参数

调用 `setCameraSettings` 设置登录、视频流参数，该相机支持的视频流格式参考附录

新版网络协议的相机支持通过 `getStreamAbility` 来获取相机支持的视频流格式，但需登录后才可获取

## 登录相机

调用 `login` 登录到对应IP地址的相机，以调用控制接口

## 控制相机

调用 `setXX/getXX` 来控制/获取相机的某些参数，例如曝光、白平衡、聚焦等

## 取流

首先调用 `openStream` 打开视频流，分辨率、格式和帧率为 `setCameraSettings` 传入的参数

再调用 `getVideoFrameData` 获取一帧视频帧，可用于显示，或图像处理等其他用途

视频帧使用完毕后调用 `l_sdk_dec_free_md_data` 进行释放，避免内存泄漏

如不再取流，调用 `closeStream` 以关闭视频流

## 相机退出登录

调用 `logout` 退出登录，调用完毕后，将无法控制及取流

## SDK API销毁

当SDK使用完毕后，调用 `destroyNet` 释放SDK占用的资源，整个程序结束时调用一次，不建议频繁调用

## 接口参考

---

### 初始化SDK

SDK初始化操作，在整个程序开始时执行一次，不建议频繁调用

```
1 | static void initNet();
```

## 检测当前环境在线相机

返回所有在线相机设备信息

```
1 | static std::vector<NetInfo> searchAvailableCameras();
```

## 创建相机

根据型号及版本号创建对应相机对象，用于区分新老协议相机

```
1 | static std::unique_ptr<NetCamera> create(const std::string &cameraModel, int version);
```

## 设置登录、视频流参数

```
1 | bool setCameraSettings(const CameraSettings &settings);
```

```
1 | settings.netInfo.username = "admin";
2 | settings.netInfo.password = "123456";
3 | struct CameraSettings
4 | {
5 |     NetInfo netInfo; //相机用户名密码
6 |     CameraResolution resolution; //相机分辨率
7 |     StreamFormat streamFormat = StreamFormat::kJPEG;
8 |     PixFormat pixFormat = PixFormat::kYUV420P;
9 |     // cbr（定码率） / vbr（变码率）
10 |    std::string rcMode; //码率控制模式（默认
    cbr）
11 |    // high（高） / medium（中） / low（低）
12 |    std::string quality; //图像质量
13 |    int frameRate = 0; //帧率
14 |    int bitrate = 0; //比特率（51200-
    86000）
15 |    int keyFrameInterval = 0; //关键帧间隔（一般未3*
    帧率，但是关键帧对mpeg无效，设置为90就可以）
16 | };
```

## 相机登录

```
1 | bool login();
```

## 获取当前已登录相机的各参数

```
1 | virtual CameraInfo getNetCameraAllInfo();
```

```
1 | CameraInfo NetCamera::getNetCameraAllInfo()
2 | {
3 |     getRangeInfo();
4 |     getControlInfo(); // 获取控制信息
```

```

5      getFocusRegionInfo(); // 获取聚焦区域信息
6      getExposureInfo(); // 获取曝光信息
7      getWbInfo(); // 获取白平衡信息
8      getFlipMirrorStatus(); // 获取翻转镜像状态
9      getGammaValue(); // 获取伽马值
10     getDefogStatus(); // 获取除雾状态
11     getFlickStatus(); // 获取抗频闪状态
12     getSystemBaseInfo(); // 相机基础系统信息
13     getIpv4Info(); // 获取IP
14     return m_cameraInfo;
15 }

```

## 打开视频流

```

1 | bool openStream();

```

## 获取一帧视频帧

```

1 | bool getVideoFrameData(l_md_data_t *&frameData);

```

## 退出登录

```

1 | bool logout();

```

## 关闭视频流

```

1 | bool closeStream();

```

## 销毁SDK

当SDK使用完毕后，调用以释放SDK占用的资源，在整个程序结束时执行一次，不建议频繁调用

```

1 | static void destroyNet();

```

## 变更视频流参数

```

1 | bool changeStreamSettings(const CameraSettings &settings);

```

## 聚焦设置

```

1 | // mode为聚焦模式
2 | // value为手动聚焦值（手动聚焦下有效，M系列不支持聚焦值设置，A系类C系列聚焦范围1-400）
3 | // speed 聚焦速度（无效）
4 | virtual bool setFocus(FocusMode mode, int value, int speed = 3);
5 |
6 | // 聚焦区域设置
7 | // enable = true时启用
8 | // x x轴位置（小框，0-16；大框，0-4）
9 | // y y轴位置（小框，0-14；大框，0-4）
10 | // mode为聚焦区域大小（0-小 1-大 目前仅使用小）

```

```

11 virtual bool setFocusRegion(bool enable, int x, int y, int mode = 0);
12 virtual bool focusAdd(); // M系列独有功能，增加1点聚焦值
13 virtual bool focusSub(); // M系列独有功能，减少1点聚焦值
14
15 // 设置聚焦区域，可自由设置聚焦区域位置及大小
16 // enable 是否启用
17 // x      区域左上角x坐标
18 // y      区域左上角y坐标
19 // width  区域宽度
20 // height 区域高度
21
22 virtual bool setFocusRegionV2(bool enable, int x, int y, int width, int
height);
23 virtual bool getFocusRegionV2(bool &enable, int &x, int &y, int &width, int
&height);
24
25 // 获取当前聚焦状态
26 // focusing true-正在聚焦，false-聚焦完成
27 virtual bool getFocusState(bool &focusing);
28
29 // 设置变倍
30 // ezoomValue - 光学变倍值
31 // dzoomValue - 数字变倍值
32 // speed 变倍速率 0-低 1-中 2-高
33 // zoomMode ZoomAbsolute 绝对变倍模式，ezoom值控制变倍位置；
34 //          ZoomRelative 相对变倍模式，当ezoom值为0-停止变倍，1-缩小，-1-放大
35 virtual bool setZoom(int ezoomValue, int dzoomValue, int speed, ZoomMode
mode = ZoomMode::ZoomAbsolute);

```

## 曝光设置

```

1 // ExposureMode 为曝光模式
2 // compensation 为目标亮度（自动曝光模式下有效）
3 // gain 为增益（手动曝光模式下有效）
4 // shutterTime 为快门时间（手动曝光模式下有效）
5 virtual bool setExposure(ExposureMode mode, int compensation, int gain, int
shutterTime);

```

## 白平衡设置

```

1 // wbMode 为白平衡模式
2 // red为红色、green为绿色、blue为蓝色（手动白平衡模式下有效，范围为0-4095）
3 virtual bool setWb(WbMode mode, int red, int green, int blue);
4
5 // 设置白平衡色温
6 // 色温和 RGB值是互斥关系，不可同时设置。修改色温，相机内部实现为调节RGB分量
7 virtual bool setWbTemperature(int temperature);

```

## 镜像翻转

```

1 virtual bool setFlip(bool enable); // 垂直翻转（enable = true启用翻转）
2 virtual bool setMirror(bool enable); // 水平翻转（enable = true启用镜像）

```

## 消紫边

```
1 | virtual bool setCAC(bool enable); // 消紫边 (enable = true启用消紫边)
```

## 抗频闪

```
1 | // 启用/关闭后，如原帧率为1080p 60fps / 4k 30fps，  
2 | // 重新调用changeStreamSettings修改帧率至1080p 60fps / 4k 30fps  
3 | // 否则可能出现帧率异常的情况  
4 | virtual bool setFlick(bool enable); // 抗频闪 (enable = true启用抗频闪)
```

## 黑白模式

```
1 | virtual bool setMonochrome(bool enable); // enable = true启用黑白模式（通过调节饱和度为0来实现黑白效果）
```

## 除雾设置

```
1 | virtual bool setDefog(bool enable); //enable = true启用除雾模式
```

## 补光灯设置

```
1 | // index 为补光灯分块序号  
2 | // level 为亮度等级  
3 | // 仅支持A系列  
4 | virtual bool setLed(int partIndex, int level);
```

## 对比度

```
1 | virtual bool setContrastValue(int value);
```

## 饱和度

```
1 | virtual bool setSaturationValue(int value);
```

## 色度

```
1 | virtual bool setHueValue(int value);
```

## 锐度

```
1 | virtual bool setSharpnessValue(int value);
```

## 伽马

```
1 | virtual bool setGammaValue(int value);
```

## 消反光

```
1 | virtual bool setDRCValue(int value);
```

## IP设置

```
1 | // 设置相机有线IP相关信息
2 | // dhcp      是否启用DHCP，有线DHCP暂不可用，默认为false
3 | // ip        相机新IP
4 | // gateway   网关
5 | // netmask   子网掩码
6 | // dns       dns地址
7 | // mac       MAC地址
8 | // 如在拉流状态，修改IP后视频流将会断开
9 | virtual bool setIpv4(bool dhcp, const std::string &ip, const std::string
  &gateway, const std::string &netmask, const std::string &dns1, const
  std::string &dns2, const std::string &mac);
```

## 设置无线IP地址

```
1 | // dhcp 是否启用DHCP
2 | // ip 相机新IP
3 | // gateway 网关
4 | // netmask 子网掩码
5 | // dns dns地址
6 | // mac MAC地址，仅支持单播MAC，即第一个字节最后一位不可为1
7 | virtual bool setWirelessIPv4(bool dhcp, const std::string &ip, const
  std::string &gateway, const std::string &netmask, const std::string &dns1,
  const std::string &dns2, const std::string &mac);
```

## 设置无线模式

```
1 | // type      ap/sta 无线模式
2 | // net       2.4g/5g 无线频段，目前仅支持5g
3 | // apSsid    AP模式下，相机AP名称(IP只为7.1)
4 | // apPasswd  相机AP密码
5 | // staSsid   STA模式下，路由器无线名称
6 | // staPasswd STA模式下，路由器无线密码
7 | // enc       wpa-wpa2 psk/wpa2 psk/psk/none
8 | //          AP模式下，为相机AP的加密方式；
9 | //          STA模式下，为路由器WIFI的加密方式
10 | virtual bool setWireless(const std::string &type, const std::string &net,
  const std::string &apSsid, const std::string &apPasswd, const std::string
  &staSsid, const std::string &staPasswd, const std::string &enc);
```

## 升级功能

```
1 | // upgradeFilePath 相机升级文件绝对路径
2 | // 升级文件名称规范 例：C313型号相机升级名称为：c313_update.1pk
3 | virtual bool upgrade(const std::string &upgradeFilePath);
4 | virtual bool getUpgradeStatus(std::string &status, double &percentage);
```

## 场景切换



```
1 // 0-生物, 1-工业, 2-金相, 3-单筒 (仅L116支持, 支持工业/金相/单筒)
2 virtual bool setScene(int mode);
```

## 锐度增强

```
1 virtual bool setSharpnessEnhancement(bool enable);
```

## 对比度增强

```
1 virtual bool setContrastEnhancementValue(int value);
```

## 图片流 (仅老私有协议相机支持)

```
1 // 获取当前图片流参数, 包括范围等信息
2 virtual bool getPicStream();
3
4 // 设置图片流参数
5 virtual bool setPicStream(const CameraPicStreamSettings &settings);
6 // 打开图片流
7 virtual bool requestPicStream();
8 // 关闭图片流
9 virtual bool closePicStream();
10
11 // 打开图片流后才可拍照, 保存图片到指定路径下
12 // filePath 文件绝对路径, 带后缀名, 仅支持jpeg
13 virtual bool savePic(const std::string &filePath);
```

## 附录

---

相机名称	输出分辨率	自动 聚焦	手动 聚焦	智能 聚焦	一键 聚焦	智能区域 聚焦	曝光 区域	曝光	亮度	快门	增益	白平衡	色温	RGB	摄像头 特性	色度	饱和度	对比度	对比度增强 (LDC1)	锐度	数字 灰阶	zoom 范围	zoom 范围	黑白	降噪	伽马	遮光 50/60Hz	滑轨边	gamma 曲线	锐度 增强	对比度 增强	场景		
M202B (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265)	✓	✓	智能+ 智能-	✓	✓	小智能框 水平0°-16° 垂直0°-14°	✓	0°255	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0-56	0-144	✓	✓	0°10	×	✓	×	×	×	×	×	
M202 (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265)	✓	✓	智能+ 智能-	✓	✓	小智能框 水平0°-16° 垂直0°-14°	✓	0°255	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0-56	0-144	✓	✓	0°10	×	✓	×	×	×	×	×	
M112 (网络)	60FPS@1920x1080 (H265)	✓	✓	智能+ 智能-	✓	✓	小智能框 水平0°-16° 垂直0°-14°	✓	0°100	0°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0-50	0-130	✓	✓	0-3	×	✓	×	×	×	×	×	
M112B (网络)	60FPS@1920x1080 (H265)	✓	✓	智能+ 智能-	✓	✓	小智能框 水平0°-16° 垂直0°-14°	✓	0°100	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0-50	0-130	✓	✓	0-3	×	✓	×	×	×	×	×	
M114 (网络)	30FPS@1920x1080 (H265) 25FPS@3840x2160 (H265)	✓	✓	智能+ 智能-	✓	✓	小智能框 水平0°-16° 垂直0°-14°	✓	0°100	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0-56	0-144	✓	✓	0°3	×	✓	×	×	×	×	×	
M212 (网络)	30FPS@3840x2160 (H265) 60FPS@1920x1080 (H265) 30FPS@3840x2160 (H264) 60FPS@1280x720 (H265)	✓	✓	智能+ 智能-	✓	支持自定义	支持自定义	✓	0°255	0°16.50ms	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	✓	0°255	✓	0-56	0-144	✓	✓	1°170	✓	✓	✓	✓	×	×	×	
M122 (网络)	60FPS@1920x1080 (H265) 60FPS@1920x1080 (H265) 60FPS@1280x720 (H264) 60FPS@1920x1080 (H264)	✓	✓	智能+ 智能-	✓	支持自定义	支持自定义	✓	0°255	0°16.50ms	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	✓	0°255	✓	0-50	0-130	✓	✓	1°170	✓	✓	✓	✓	×	×	×	
C310 (网络)	60FPS@1920x1080 (H265)	✓	✓	1°400	✓	支持自定义	支持自定义	✓	0°255	0°16.50ms	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	1°170	✓	✓	✓	✓	×	✓	×	
C311 (网络)	60FPS@1920x1080 (H265)	✓	✓	1°400	✓	支持自定义	支持自定义	✓	0°255	0°16.50ms	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	1°170	✓	✓	✓	✓	×	✓	×	
L311 (网络)	60FPS@1920x1080 (H265)	×	×	×	×	×	×	✓	0°255	0°16.50ms	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	1°170	✓	✓	✓	✓	×	✓	×	
C312 (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265) 30FPS@2592x1444 (H265)	✓	✓	1°400	✓	✓	小智能框 水平0°-16° 垂直0°-14° 大智能框 水平0°-4° 垂直0°-4°	✓	0°250	1°333	0°24	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	✓	✓	✓	✓	×	×	×	×
C313T (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265)	✓	✓	1°400	✓	✓	小智能框 水平0°-16° 垂直0°-14° 大智能框 水平0°-4° 垂直0°-4°	✓	0°250	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	✓	✓	✓	✓	✓	×	×	×
C313 (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265)	✓	✓	1°400	✓	✓	小智能框 水平0°-16° 垂直0°-14° 大智能框 水平0°-4° 垂直0°-4°	✓	0°250	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	✓	✓	✓	✓	×	×	×	×
C314 (网络)	30FPS@1920x1080 (H265) 30FPS@3840x2160 (H265) 30FPS@4000x3000 (H265)	✓	✓	1°400	✓	✓	小智能框 水平0°-16° 垂直0°-14° 大智能框 水平0°-4° 垂直0°-4°	✓	0°250	1°333	0°27	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	✓	✓	✓	✓	×	×	×	×
L313 (03E)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265) 60FPS@1920x1080 (H212) 15FPS@3840x2160 (H212)	×	×	×	×	×	×	✓	0°250	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	✓	✓	✓	✓	✓	×	×	×
L313 (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265)	×	×	×	×	×	×	✓	0°250	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	✓	✓	✓	✓	✓	×	×	×
L314 (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265) 30FPS@4000x3000 (H265)	×	×	×	×	×	×	✓	0°250	1°333	0°38	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	✓	✓	✓	✓	✓	×	×	×
L116T (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265)	×	×	×	×	×	×	✓	0°255	0°333	0-38	✓	×	1-4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	×	✓	×	×	×	×	×	
L116 (网络)	60FPS@1920x1080 (H265) 30FPS@3840x2160 (H265)	×	×	×	×	×	×	✓	0°255	0°333	0-38	✓	×	1-4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	×	✓	×	×	×	×	×	
A311 (网络)	60FPS@1280x720 (H265) 60FPS@1920x1080 (H265)	✓	✓	0-400	✓	支持自定义	支持自定义	✓	0°250	0°16.50ms (66/4)	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
A312 (网络)	60FPS@1280x720 (H265) 60FPS@1920x1080 (H265)	✓	✓	0-400	✓	支持自定义	支持自定义	✓	0°250	0°16.50ms (66/4)	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
A321 (网络)	60FPS@1280x720 (H265) 60FPS@1920x1080 (H265)	×	×	×	×	×	×	✓	0°250	0°16.50ms (66/4)	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
A322 (网络)	60FPS@1280x720 (H265) 60FPS@1920x1080 (H265)	×	×	×	×	×	×	✓	0°250	0°16.50ms (66/4)	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
A3110 (USB)	60FPS@1280x720 (H212) 60FPS@1920x1080 (H212) 60FPS@1280x720 (H212) 60FPS@1920x1080 (H212)	✓	✓	0-400	✓	支持自定义	支持自定义	✓	0°250	0°16.50ms (66/4)	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	✓	✓	✓	✓	×	×	×	
A3120 (USB)	60FPS@1280x720 (H212) 60FPS@1920x1080 (H212) 60FPS@1280x720 (H212) 60FPS@1920x1080 (H212)	✓	✓	0-400	✓	支持自定义	支持自定义	✓	0°250	0°16.50ms (66/4)	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
A312B (网络)	60FPS@1280x720 (H265) 60FPS@1920x1080 (H265)	✓	✓	0-400	✓	支持自定义	支持自定义	✓	0°250	0°33ms	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
A313 (网络)	30fps@3840x2160(a.jpg) 60fps@1920x1080(a.jpg) 60fps@1920x1080(a.jpg)	✓	✓	0-400	✓	支持自定义	支持自定义	✓	0°250	0°16.50ms (66/4)	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
A314 (网络)	30fps@3840x2160(a.jpg) 30fps@3840x2160(a.jpg) 30fps@1920x1080(a.jpg)	✓	✓	0-400	✓	支持自定义	支持自定义	✓	0°250	0°33ms	0°30	✓	×	1°4095	✓	0°100	0°255	0°100	×	0°15	✓	0.7°4.8	✓	✓	×	1°170	0°255	✓	✓	✓	×	×	×	
F801F	25FPS@1280x972 (H264) 25FPS@1920x1080 (H264)	×	×	×	×	×	×	✓	0°255	0°333	0-38	✓	×	1-4095	✓	0°100	0°255	0°100	×	0°15	×	×	×	✓	×	0°3	×	✓	×	×	×	×	×	