IPEVO Camera API v1.0.0.1

```
Camera
    |- UVCCamera
    |        |- IPEVOCamera
    |                   |- IPEVOCameraWithHID
    |                                |- P2V
    |                                |- P2V_R
    |                                |- VZ_1
    |                                |- Ziggi
    |                                |- ZiggiHD
    |                                |- ZiggiHD_2
    |- NetCamera
```

Camera

Syntax:

get camera from `CameraManager.sharedManager.cameras`

Property:

Name	Description
`model`	Obtain the camera model string. EX: IPEVO Point 2 View
`shortModel`	Obtain the camera short model string. EX: P2V
`CameraInstanceName`	Obtain the camera instance name string. EX: IPEVO Ziggi-HD #1

Function:

Name	Description
`hasCapability(Capability capability)`	Return camera capability
`PrepareToWork()`	Initiate some basic camera functions
`prepareToDetach()`	Prepare the camera for detachment
`supportedFormats()`	`return List<VideoResolutionFormat>`
`setFormat(VideoResolutionFormat format)`	Set the camera image stream format
`getFormat(out VideoResolutionFormat format)`	Obtain the current camera current stream format
`getDefaultFormat(out VideoResolutionFormat format)`	Obtain the default camera default stream format
`setBrightness(short value)`	Set brightness value
`getBrightness(out short value, PropertyValueType type)`	Obtain the brightness value
`setContrast(short value)`	Set contrast value. Set the value between `getContrast(out maxValue ,PropertyValueType.Maximum)` and `getContrast(out minValue,PropertyValueType.Minimum)`

`getContrast(out short value, PropertyValueType type)`	Obtain the contrast value.
	`The type can be...`
	`PropertyValueType.Default` or
	`PropertyValueType.Maximum` or
	`PropertyValueType.Minimum` or
	`PropertyValueType.Current`

`setGamma(short value)`	Set gamma value
`getGamma(out short value, PropertyValueType type)`	Obtain the gamma value
`setHue(short value)`	Set hue value
`getHue(out short value, PropertyValueType type)`	Obtain the hue value
`setSaturation(short value)`	Set saturation value
`getSaturation(out short value, PropertyValueType type)`	Obtain the saturation value
`setSharpness(short value)`	Set sharpness value
`getSharpness(out short value, PropertyValueType type)`	Obtain the shapeness value
`setWhiteBalance(short value)`	Set white balance value
`getWhiteBalance(out short value, PropertyValueType type)`	Obtain the white balance value
`setAutoWhiteBalance(bool value)`	Set auto white balance value
`getAutoWhiteBalance(out bool value, PropertyValueType type)`	Obtain the auto white balance value
`setFocus(bool isFocusLock, short focusDistance)`	Set focus mode and focus distance
`getFocus(out short value, PropertyValueType type)`	Obtain the focus mode value = 0, single focus

	value = 1, conitune focus
	value = 2, focus lock
	value = 3, focus not lock

`getFocusValue(out short value, PropertyValueType type)`	Obtain the focus distance
`setAutoExposureLock(bool value)`	Set auto exposure lock
`getAutoExposureLock(out bool value)`	Obtain the auto exposure lock
`setExposure(short value)`	Set exposure value 1~15
`getExposure(out short value, PropertyValueType type)`	Obtain the current exposure value
`getDefaultExposure(out short value)`	Obtain the default exposure value
`setFrequency(PowerlineFrequency value)`	Set power frequency
`getFrequency(out PowerlineFrequency value, PropertyValueType type)`	Obtain the power frequency
`getFirmwareVersion(out string ver)`	Obtain the camera firmware version
`startFocus()`	Let the camera start performing focus

CameraManager Property

Syntax:

```
CameraManager cameraManager = CameraManager.sharedManager;
cameraManager.startMonitor();
this.cameraListComboBox.ItemsSource = cameraManager.cameras;
cameraManager.stopMonitor();        //"must" call this before your Application Exit
```

Property:

Name	Description
`sharedManager`	Use this static property to obtain all CameraManger instance
`cameras`	Obtain the supported Camera
`NetCameraType`	`enum NetCameraType{ None, WS01, IZiggi }`

`Notification`	`DeviceAttached`
	`DeviceDetached`
	`DeviceSelected`
	`HID_SNAP_SHOT_BTN`
	`HID_FOCUS_BTN_PRESS`
	`HID_FOCUS_BEGIN`
	`HID_FOCUS_FINISH`
	`HID_S_AF`
	`HID_C_AF`
	`CAMERA_EXPOSURE_CHANGE`

Function:

Name	Description
`startMonitor()`	Start to monitor the changing camera status. When changes happen, you will receive

	notifications like "DeviceAttached" or "DeviceDetached".
`stopMonitor()`	Stop monitoring camera changes.

StreamProxy

Syntax:

Use the static instance StreamProxy.sharedProxy

Function:

Name	Description
`addStreamObserverEX(Camera camera, StreamObserverEX observer)`	Add stream observer
`removeStreamObserverEX(Camera camera, StreamObserverEX observer)`	Remove the stream observer
`stopStreamObserver(Camera camera)`	Stop the stream observer
`startStreamObserver(Camera camera)`	Start the stream observer