UVC USB Camera
Linux Development Specification

深圳市锐尔威视科技有限公司
RERVISION TECHNOLOGY CO., LTD

2014.10.22 V1.0

V4L2 编程 设置 Camera 属性参数

参数包括：亮度、对比度、饱和度、色调、清晰度、增益、伽玛、曝光、白平衡、对焦

统一使用 ioctl 函数设置或读取参数，设置参数的 cmd：VIDIOC_S_CTRL，读取参数 cmd：VIDIOC_G_CTRL

传入的 arg 是结构体 v4l2_control

```
struct v4l2_control {
    __u32  id;
    __s32  value;
};
```

其中 id 是各项参数的命令字，value 是各项参数的值。
对于 USB 摄像头，V4L2 提供的设置参数接口有：
亮度、对比度、饱和度、色调、清晰度、增益、曝光、伽玛、白平衡、对焦等

亮度(Brightness)

```
struct v4l2_control control_s;
control_s.id    = V4L2_CID_BRIGHTNESS;
control_s.value     = brightness_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

对比度（contrast）

```
struct v4l2_control control_s;
control_s.id    = V4L2_CID_CONTRAST;
control_s.value     = contrast_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

饱和度(Saturation)

```
struct v4l2_control control_s;
control_s.id    = V4L2_CID_SATURATION;
control_s.value     = saturation_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

色调(Hue)

```
struct v4l2_control control_s;
control_s.id    = V4L2_CID_HUE;
control_s.value     = hue_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

清晰度(Sharpness)

```
struct v4l2_control control_s;
control_s.id    = V4L2_CID_SHARPNESS;
control_s.value     = sharpness_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

增益(Gain)

```
struct v4l2_control control_s;
control_s.id    = V4L2_CID_GAIN;
control_s.value     = gain_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

伽玛(Gamma)

```
struct v4l2_control control_s;
control_s.id    = V4L2_CID_GAMMA;
control_s.value     = gamma_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

曝光(Exposure)

```
//自动曝光
struct v4l2_control control_s;
control_s.id        = V4L2_CID_EXPOSURE_AUTO;
control_s.value     = V4L2_EXPOSURE_APERTURE_PRIORITY;
ioctl(dev, VIDIOC_S_CTRL, &control_s);

//手动曝光
struct v4l2_control control_s;
control_s.id        = V4L2_CID_EXPOSURE_AUTO;
control_s.value     = V4L2_EXPOSURE_MANUAL;
ioctl(dev, VIDIOC_S_CTRL, &control_s);

//设置曝光值
struct v4l2_control control_s;
control_s.id    = V4L2_CID_EXPOSURE_ABSOLUTE;
control_s.value     = expouse_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

白平衡(White Balance)

```
//自动白平衡
struct v4l2_control control_s;
control_s.id      = V4L2_CID_AUTO_WHITE_BALANCE;
control_s.value   = 1;
ioctl(dev, VIDIOC_S_CTRL, &control_s);

//手动白平衡
struct v4l2_control control_s;
control_s.id      = V4L2_CID_AUTO_WHITE_BALANCE;
control_s.value   = 0;
ioctl(dev, VIDIOC_S_CTRL, &control_s);

//设置白平衡值
struct v4l2_control control_s;
control_s.id      = V4L2_CID_WHITE_BALANCE_TEMPERATURE;
control_s.value   = wb_value;
ioctl(dev, VIDIOC_S_CTRL, &control_s);
```

对焦(Focusing) ---（需要使用支持自动对焦的 Camera）

```
//自动对焦
struct v4l2_control control;
control.id      = V4L2_CID_FOCUS_AUTO;
control.value   =1;
ioctl(dev, VIDIOC_S_CTRL, &control);

//手动对焦
//先关闭自动对焦
control.id      = V4L2_CID_FOCUS_AUTO;
control.value   = 0;
ioctl(dev, VIDIOC_S_CTRL, &control);
//得到当前焦点值
ioctl(dev, VIDIOC_G_CTRL, &control);
//设置焦点值
int focus_value;
control.id      = V4L2_CID_FOCUS_ABSOLUTE;
control.value   = focus_value;
ioctl(dev, VIDIOC_S_CTRL, &control);
```