Ailipu Technology CO.LTD

UVC USB Camera Linux Development Specification

V4L2 Program Set Camera specification parameters

Parameters include: Brightness, contrast, saturation, hue, sharpness, gain, gamma, exposure, white balance, Focusing.

Unified use ioctl Function set or read parameters, set parameter's cmd: VIDIOC_S_CTRL, read parameter cmd: VIDIOC_G_CTRL

Incoming arg is a structure v4l2_control

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

Among them id is command word of the parameters, value Is the value of the parameter.

For USB camera, V4L2 provide Setting parameters interfaces include:

Brightness, contrast, saturation, hue, sharpness, gain, exposure, gamma, white balance, focus and other

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)

```
// Auto Exposure
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);
//Manual Exposure
```

```
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);

// Set exposure value

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)

```
// Auto white balance
struct v4l2_control control_s;
control_s.id = V4L2_CID_AUTO_WHITE_BALANCE;
control_s.value = 1;
ioctl(dev, VIDIOC_S_CTRL, &control_s);

// manual white balance

  struct v4l2_control control_s;
control_s.id = V4L2_CID_AUTO_WHITE_BALANCE;
control_s.value = 0;
ioctl(dev, VIDIOC_S_CTRL, &control_s);

//  set the value of white balance
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) ---（ need to use Camera, which support auto fucus）

```
// auto focus
```

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

//manual focus
// first close auto focus
 control.id = V4L2_CID_FOCUS_AUTO;
control.value = 0;
ioctl(dev, VIDIOC_S_CTRL, &control);

// Get the current focus value
ioctl(dev, VIDIOC_G_CTRL, &control);

//  set focus value
int focus_value;
control.id = V4L2_CID_FOCUS_ABSOLUTE;
control.value = focus_value;
ioctl(dev, VIDIOC_S_CTRL, &control);
```