Camera SDK Features Guide

Python

Release Date: June 11, 2025

Version: V1.0

SHANGHAI PERCIPIO TECHNOLOGY LIMITED

Table of Contents

1 Features Under the *Device* Component

1.1 TY_BOOL_GVSP_RESEND

```
param =  cl.DevParamFromBool(True)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_BOOL_GVSP_RESEND, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_BOOL_GVSP_RESEND)

m_status=status.toBool()
```

1.2 TY_BOOL_CMOS_SYNC

```
param =  cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_BOOL_CMOS_SYNC, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_BOOL_CMOS_SYNC)

m_status=status.toBool()
```

1.3 TY_INT_ACCEPTABLE_PERCENT

```
para =cl.DevParamFromInt(80)

cl.DeviceSetParameter(handle,TY_COMPONENT_DEVICE,TY_INT_ACCEPTABLE_PERCENT,para)

value = cl.DeviceGetParameter(handle,TY_COMPONENT_DEVICE,TY_INT_ACCEPTABLE_PERCENT)

m_value = value.toInt()

value_min = value.mMin()

value_max = value.mMax()

value_inc = value.mInc()

print('min {} max {} inc {} cunrrent {}'.format(value_min,value_max,value_inc,m_value))
```

1.4 TY_INT_PACKET_SIZE

```
param =  cl.DevParamFromInt(1500)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PACKET_SIZE, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PACKET_SIZE)

m_read_param=read_param.toInt()
```

1.5 TY_INT_PACKET_DELAY

```
param =  cl.DevParamFromInt(0)
```

```
cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PACKET_DELAY, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PACKET_DELAY)
m_read_param=read_param.toInt()
```

1.6 TY_BOOL_KEEP_ALIVE_ONOFF

```
param = cl.DevParamFromBool(True)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_BOOL_KEEP_ALIVE_ONOFF, param)
status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_BOOL_KEEP_ALIVE_ONOFF)
m_status=status.toBool()
print('TY_BOOL_KEEP_ALIVE_ONOFF status {}'.format(m_status))
```

1.7 TY_INT_KEEP_ALIVE_TIMEOUT

```
param =  cl.DevParamFromInt(3000)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_KEEP_ALIVE_TIMEOUT, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_KEEP_ALIVE_TIMEOUT)
m_read_param=read_param.toInt()
```

1.8 TY_ENUM_TIME_SYNC_TYPE

```
param =  cl.DevParamFromEnum(TY_TIME_SYNC_TYPE_HOST)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_ENUM_TIME_SYNC_TYPE, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_ENUM_TIME_SYNC_TYPE)
m_read_param=read_param.toEnum()
value_list = read_param.eList()
for i in range (len(value_list)):
  print('description {} , value {}'.format(value_list[i].description,value_list[i].value))
```

1.9 TY_BOOL_TIME_SYNC_READY

```
status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_BOOL_TIME_SYNC_READY)
m_status=status.toBool()
```

1.10 TY_ENUM_STREAM_ASYNC

```
para =cl.DevParamFromEnum(TY_STREAM_ASYNC_ALL)
cl.DeviceSetParameter(handle,TY_COMPONENT_DEVICE,TY_ENUM_STREAM_ASYNC,para)
value = cl.DeviceGetParameter(handle,TY_COMPONENT_DEVICE,TY_ENUM_STREAM_ASYNC)
```

```
m_value = value.toEnum()

print('current value {}'.format(m_value))

value_list = value.eList()

for i in range (len(value_list)):

  print('description {} , value {}'.format(value_list[i].description,value_list[i].value))
```

1.11 TY_INT_CAPTURE_TIME_US

```
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_CAPTURE_TIME_US)

m_read_param=read_param.toInt()
```

1.12 TY_INT_NTP_SERVER_IP

```
ip = cl.IPv4StringToInt('0.0.0.0')

param =  cl.DevParamFromInt(ip)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_NTP_SERVER_IP, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_NTP_SERVER_IP)

read_param_m  = read_param.toInt()

print('ip',read_param_m)
```

1.13 TY_INT_PERSISTENT_IP

```
ip = cl.IPv4StringToInt('0.0.0.0')

param =  cl.DevParamFromInt(ip)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PERSISTENT_IP, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PERSISTENT_IP)

read_param_m  = read_param.toInt()

print('ip',read_param_m)
```

1.14 TY_INT_PERSISTENT_SUBMASK

```
netmask = cl.IPv4StringToInt('0.0.0.0')

param =  cl.DevParamFromInt(netmask)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PERSISTENT_SUBMASK, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PERSISTENT_SUBMASK)

read_param_m  = read_param.toInt()

print('netmask',read_param_m)
```

1.15 TY_INT_PERSISTENT_GATEWAY

```
gateway =cl.IPv4StringToInt('0.0.0.0')

param =  cl.DevParamFromInt(gateway)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PERSISTENT_GATEWAY, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_PERSISTENT_GATEWAY)

read_param_m  = read_param.toInt()

print('gateway',read_param_m)
```

1.16 TY_ENUM_CONFIG_MODE

```
  para =cl.DevParamFromEnum(TY_CONFIG_MODE_PRESET0)

  cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_ENUM_CONFIG_MODE,para)

  value = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_ENUM_CONFIG_MODE)

  m_value = value.toEnum()

  value_list = value.eList()

  for i in range (len(value_list)):

    print(' {}'.format(value_list[i].value))
```

1.17 TY_ENUM_TEMPERATURE_ID

```
# Get temperature sensor list

param5 = cl.DevParamFromEnum(TY_TEMPERATURE_LEFT)

error5 = cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_ENUM_TEMPERATURE_ID, param5)

read_param5 = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_ENUM_TEMPERATURE_ID)

m_read_param5 = read_param5.toEnum()

print(f"current value {m_read_param5}")

l_read_param5 = read_param5.eList()

for i in range(len(l_read_param5)):

print(f"{l_read_param5[i].value}")

# Get temperature value

temp_mode = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_ENUM_TEMPERATURE_ID)

if temp_mode.isEmpty():

print('Temperature is not support!')

else :
```

```
list = temp_mode.eList()

for idx in range(len(list)):

mode = list[idx]

temp = cl.DeviceControlReadTemperature(handle, cl.Value(mode))

print('{}: {} TEMP: {}'.format(idx, cl.Description(mode), temp))
```

1.18 TY_ENUM_TRIGGER_POL

```
para =cl.DevParamFromEnum(TY_TRIGGER_POL_FALLINGEDGE)

cl.DeviceSetParameter(handle,TY_COMPONENT_DEVICE,TY_ENUM_TRIGGER_POL,para)

value = cl.DeviceGetParameter(handle,TY_COMPONENT_DEVICE,TY_ENUM_TRIGGER_POL)

m_value = value.toEnum()

print('current value {}'.format(m_value))

value_list = value.eList()

for i in range (len(value_list)):

  print(' {}'.format(value_list[i].value))
```

1.19 TY_INT_FRAME_PER_TRIGGER

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_FRAME_PER_TRIGGER, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEVICE, TY_INT_FRAME_PER_TRIGGER)

m_read_param=read_param.toInt()
```

2 Features Under the *Laser* Component

2.1 TY_BOOL_LASER_AUTO_CTRL

```
param=cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_LASER, TY_BOOL_LASER_AUTO_CTRL, param)

status =cl.DeviceGetParameter(handle, TY_COMPONENT_LASER, TY_BOOL_LASER_AUTO_CTRL)

m_status=status.toBool()
```

2.2 TY_BOOL_RGB_FLASHLIGHT

```
param =  cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_LASER, TY_BOOL_RGB_FLASHLIGHT, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_LASER, TY_BOOL_RGB_FLASHLIGHT)

m_status=status.toBool()
```

2.3 TY_BOOL_IR_FLASHLIGHT

```
param =  cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_LASER, TY_BOOL_IR_FLASHLIGHT, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_LASER, TY_BOOL_IR_FLASHLIGHT)

m_status=status.toBool()
```

2.4 TY_INT_LASER_POWER

```
param =  cl.DevParamFromInt(100)

cl.DeviceSetParameter(handle, TY_COMPONENT_LASER, TY_INT_LASER_POWER, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_LASER, TY_INT_LASER_POWER)

m_read_param=read_param.toInt()
```

2.5 TY_INT_RGB_FLASHLIGHT_INTENSITY

```
param =  cl.DevParamFromInt(0)

cl.DeviceSetParameter(handle, TY_COMPONENT_LASER, TY_INT_RGB_FLASHLIGHT_INTENSITY ,param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_LASER, TY_INT_RGB_FLASHLIGHT_INTENSITY)

m_read_param=read_param.toInt()
```

2.6 TY_INT_IR_FLASHLIGHT_INTENSITY

```
param =  cl.DevParamFromInt(0)

cl.DeviceSetParameter(handle, TY_COMPONENT_LASER, TY_INT_IR_FLASHLIGHT_INTENSITY ,param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_LASER, TY_INT_IR_FLASHLIGHT_INTENSITY)

m_read_param=read_param.toInt()
```

3 Features Under the *Depth* Component

3.1 TY_FLOAT_SCALE_UNIT

```
param =  cl.DevParamFromFloat(0.0125)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_FLOAT_SCALE_UNIT, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_FLOAT_SCALE_UNIT)

m_read_param=read_param.toFloat()
```

3.2 TY_ENUM_DEPTH_QUALITY

```
para =cl.DevParamFromEnum(TY_DEPTH_QUALITY_BASIC)

cl.DeviceSetParameter(handle,TY_COMPONENT_DEPTH_CAM,TY_ENUM_DEPTH_QUALITY,para)

value = cl.DeviceGetParameter(handle,TY_COMPONENT_DEPTH_CAM,TY_ENUM_DEPTH_QUALITY)

m_value = value.toEnum()

value_list = value.eList()

for i in range (len(value_list)):

  print(' {}'.format(value_list[i].value))
```

3.3 TY_BOOL_TOF_ANTI_INTERFERENCE

```
param =  cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_TOF_ANTI_INTERFERENCE, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_TOF_ANTI_INTERFERENCE)

m_status=status.toBool()
```

3.4 TY_INT_TOF_CHANNEL

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM,  TY_INT_TOF_CHANNEL, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_CHANNEL)

m_read_param=read_param.toInt()
```

3.5 TY_INT_TOF_MODULATION_THRESHOLD

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_MODULATION_THRESHOLD, param)
```

```
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM,
TY_INT_TOF_MODULATION_THRESHOLD)
m_read_param=read_param.toInt()
```

3.6 TY_INT_TOF_ANTI_SUNLIGHT_INDEX

```
param =  cl.DevParamFromInt(1)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_ANTI_SUNLIGHT_INDEX, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_ANTI_SUNLIGHT_INDEX)
m_read_param=read_param.toInt()
```

3.7 TY_INT_TOF_HDR_RATIO

```
param =  cl.DevParamFromInt(100)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_HDR_RATIO, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_HDR_RATIO)
m_read_param=read_param.toInt()
```

3.8 TY_INT_TOF_JITTER_THRESHOLD

```
param =  cl.DevParamFromInt(1)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_JITTER_THRESHOLD, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_TOF_JITTER_THRESHOLD)
m_read_param=read_param.toInt()
```

3.9 TY_INT_SGBM_IMAGE_NUM

```
param =  cl.DevParamFromInt(3)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_IMAGE_NUM, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_IMAGE_NUM)
m_read_param=read_param.toInt()
```

3.10 TY_INT_SGBM_DISPARITY_NUM

```
param =  cl.DevParamFromInt(1)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_DISPARITY_NUM, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_DISPARITY_NUM)
```

```
m_read_param=read_param.toInt()
```

3.11 TY_INT_SGBM_DISPARITY_OFFSET

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_DISPARITY_OFFSET, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_DISPARITY_OFFSET)

m_read_param=read_param.toInt()
```

3.12 TY_INT_SGBM_SEMI_PARAM_P1

```
param =  cl.DevParamFromInt(10000)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_SEMI_PARAM_P1, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_SEMI_PARAM_P1)

m_read_param=read_param.toInt()
```

3.13 TY_INT_SGBM_SEMI_PARAM_P2

```
param =  cl.DevParamFromInt(0)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_SEMI_PARAM_P2, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_SEMI_PARAM_P2)

m_read_param=read_param.toInt()
```

3.14 TY_INT_SGBM_UNIQUE_FACTOR

```
param =  cl.DevParamFromInt(511)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_UNIQUE_FACTOR, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_UNIQUE_FACTOR)

m_read_param=read_param.toInt()
```

3.15 TY_INT_SGBM_UNIQUE_ABSDIFF

```
param =  cl.DevParamFromInt(10000)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_UNIQUE_ABSDIFF, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_UNIQUE_ABSDIFF)

m_read_param=read_param.toInt()
```

3.16 TY_INT_SGBM_MATCH_WIN_WIDTH

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM,  TY_INT_SGBM_MATCH_WIN_WIDTH, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_MATCH_WIN_WIDTH)

m_read_param=read_param.toInt()
```

3.17 TY_INT_SGBM_MATCH_WIN_HEIGHT

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_MATCH_WIN_HEIGHT, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_MATCH_WIN_HEIGHT)

m_read_param=read_param.toInt()
```

3.18 TY_BOOL_SGBM_LRC

```
param =  cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_SGBM_LRC, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_SGBM_LRC)

m_status=status.toBool()
```

3.19 TY_INT_SGBM_LRC_DIFF

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM,  TY_INT_SGBM_LRC_DIFF, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_LRC_DIFF)

m_read_param=read_param.toInt()
```

3.20 TY_BOOL_SGBM_MEDFILTER

```
param =  cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_SGBM_MEDFILTER, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_SGBM_MEDFILTER)

m_status=status.toBool()
```

3.21 TY_INT_SGBM_MEDFILTER_THRESH

```
param =  cl.DevParamFromInt(1)

cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM,  TY_INT_SGBM_MEDFILTER_THRESH, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_MEDFILTER_THRESH)
```

```
m_read_param=read_param.toInt()
```

3.22 TY_INT_SGBM_SEMI_PARAM_P1_SCALE

```
param =  cl.DevParamFromInt(1)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM,  TY_INT_SGBM_SEMI_PARAM_P1_SCALE, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM,
TY_INT_SGBM_SEMI_PARAM_P1_SCALE)
m_read_param=read_param.toInt()
```

3.23 TY_INT_SGBM_TEXTURE_THRESH

```
param =  cl.DevParamFromInt(0)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_TEXTURE_THRESH,param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_SGBM_TEXTURE_THRESH)
m_read_param=read_param.toInt()
value_min = read_param.mMin()
value_max = read_param.mMax()
value_inc = read_param.mInc()
print('min {} max {} inc {} current {}'.format(value_min,value_max,value_inc,m_read_param))
```

3.24 TY_INT_MAX_SPECKLE_SIZE

```
param =  cl.DevParamFromInt(200)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_MAX_SPECKLE_SIZE, param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_MAX_SPECKLE_SIZE)
m_read_param=read_param.toInt()
```

3.25 TY_INT_MAX_SPECKLE_DIFF

```
param =  cl.DevParamFromInt(200)
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_MAX_SPECKLE_DIFF ,param)
read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_INT_MAX_SPECKLE_DIFF)
m_read_param=read_param.toInt()
```

3.26 TY_BOOL_SGBM_HFILTER_HALF_WIN

```
param =  cl.DevParamFromBool(False)
```

```
cl.DeviceSetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_SGBM_HFILTER_HALF_WIN, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_DEPTH_CAM, TY_BOOL_SGBM_HFILTER_HALF_WIN)

m_status=status.toBool()
```

4 Features Under the *RGB* Component

4.1 TY_INT_EXPOSURE_TIME

```
param =  cl.DevParamFromInt(4096)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_EXPOSURE_TIME, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_EXPOSURE_TIME)

m_read_param=read_param.toInt()
```

4.2 TY_INT_ANALOG_GAIN

```
param =  cl.DevParamFromInt(4096)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_ANALOG_GAIN, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_ANALOG_GAIN)

m_read_param=read_param.toInt()
```

4.3 TY_BOOL_AUTO_EXPOSURE

```
param = cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_BOOL_AUTO_EXPOSURE, param)

status =cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_BOOL_AUTO_EXPOSURE)

m_status=status.toBool()
```

4.4 TY_BOOL_AUTO_GAIN

```
param = cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_BOOL_AUTO_GAIN, param)

status =cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_BOOL_AUTO_GAIN)

m_status=status.toBool()
```

4.5 TY_BOOL_AUTO_AWB

```
param =cl.DevParamFromBool(True)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_BOOL_AUTO_AWB, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_BOOL_AUTO_AWB)

m_status=status.toBool()
```

4.6 TY_INT_R_GAIN

```
param =  cl.DevParamFromInt(4096)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_R_GAIN, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_R_GAIN)

m_read_param=read_param.toInt()
```

4.7 TY_INT_G_GAIN

```
param =  cl.DevParamFromInt(4096)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_G_GAIN, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_G_GAIN)

m_read_param=read_param.toInt()
```

4.8 TY_INT_B_GAIN

```
param =  cl.DevParamFromInt(4096)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_B_GAIN, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_B_GAIN)

m_read_param=read_param.toInt()
```

4.9 TY_STRUCT_AEC_ROI

```
roi = PercipioAecROI(0,0,640,480)

param =cl.DevParamFromPercipioAecROI(roi)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_STRUCT_AEC_ROI, param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_STRUCT_AEC_ROI)

m_read_param=read_param.toArray()

print('aec roi',m_read_param)
```

4.10 TY_INT_AE_TARGET_Y

```
param =  cl.DevParamFromInt(4000)

cl.DeviceSetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_AE_TARGET_Y ,param)

read_param = cl.DeviceGetParameter(handle, TY_COMPONENT_RGB_CAM, TY_INT_AE_TARGET_Y)

m_read_param=read_param.toInt()
```

5 Features Under the *IR* Component

5.1 TY_INT_GAIN

```
para =cl.DevParamFromInt(80)

cl.DeviceSetParameter(handle,TY_COMPONENT_IR_CAM_LEFT,TY_INT_GAIN,para)

value = cl.DeviceGetParameter(handle,TY_COMPONENT_IR_CAM_LEFT,TY_INT_GAIN)

m_value = value.toInt()

value_min = value.mMin()

value_max = value.mMax()

value_inc = value.mInc()

print('min {} max {} inc {} cunrrent {}'.format(value_min,value_max,value_inc,m_value))
```

5.2 TY_BYTEARRAY_HDR_PARAMETER

```
para_array = cl.DevParamFromByteArray([2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 31, 0, 0,

0])

cl.DeviceSetParameter(handle,TY_COMPONENT_IR_CAM_LEFT,TY_BYTEARRAY_HDR_PARAMETER,para_array)

hdr_array =cl.DeviceGetParameter(handle,TY_COMPONENT_IR_CAM_LEFT,TY_BYTEARRAY_HDR_PARAMETER)

hdr_array_1 =hdr_array.toByteArray()

for i in range (len(hdr_array_1)):

  print(',',hdr_array_1[i],end='')
```

5.3 TY_BOOL_HDR

```
param =  cl.DevParamFromBool(False)

cl.DeviceSetParameter(handle, TY_COMPONENT_IR_CAM_LEFT, TY_BOOL_HDR, param)

status = cl.DeviceGetParameter(handle, TY_COMPONENT_IR_CAM_LEFT, TY_BOOL_HDR)

m_status=status.toBool()
```

5.4 TY_FLOAT_EXPOSURE_TIME_US

```
para =cl.DevParamFromFloat(10000)

cl.DeviceSetParameter(handle,TY_COMPONENT_IR_CAM_LEFT,TY_FLOAT_EXPOSURE_TIME_US,para)

read_param = cl.DeviceGetParameter(handle,TY_COMPONENT_IR_CAM_LEFT,TY_FLOAT_EXPOSURE_TIME_US)

m_read_param = read_param.toFloat()
```

```
value_min = read_param.fMin()

value_max = read_param.fMax()

value_inc = read_param.fInc()

print('min {} max {} inc {} current {}'.format(value_min,value_max,value_inc,m_read_param))
```

6 Other Common Features

6.1 TY_ENUM_IMAGE_MODE

```
para =cl.DevParamFromEnum(TY_PIXEL_FORMAT_DEPTH16 | TY_RESOLUTION_MODE_1280x960)

cl.DeviceSetParameter(handle,TY_COMPONENT_DEPTH_CAM,TY_ENUM_IMAGE_MODE,para)

value = cl.DeviceGetParameter(handle,TY_COMPONENT_DEPTH_CAM,TY_ENUM_IMAGE_MODE)

m_value = value.toEnum()

print('current value {}'.format(m_value))

value_list = value.eList()

for i in range (len(value_list)):

   print('description {}  , value {}'.format(value_list[i].description,value_list[i].value))
```

6.2 TYSetLogLevel

```
TYSetLogLevel(1)

# Log information output by the SDK on the console. Log level definition:

# TY_LOG_LEVEL_VERBOSE = 1,TY_LOG_LEVEL_DEBUG = 2,

# TY_LOG_LEVEL_INFO = 3,TY_LOG_LEVEL_WARNING = 4,

# TY_LOG_LEVEL_ERROR = 5,TY_LOG_LEVEL_NEVER = 9,
```

6.3 TYSetLogPrefix

```
TYSetLogPrefix(“Percipio PYTHON SDK”)

# Check if the log prefix on the console is ”Percipio Python SDK”
```

Percipio.XYZ is an industry leading provider of 3D cameras. We provide a broad range of 3D camera products to meet requirements from various applications, such as industrial, automotive, inspection, logistics, medical, education, security and commercial etc. We will continue to develop and optimize our product roadmap to support more 3D vision applications.

Percipio is an independent vendor of 3D machine vision solutions. We provide products and services to system integration customers rather than end users. This marketing strategy allows us to serve multiple sectors and segments, and also means that our success will be based on our customer's success. Together with our customer's industry specific expertise, we can support end users with implementing machine intelligence, which will improve productivity and/or reduce cost.

Affordable 3D Machine Vision

Contact Us

Purchase : info@pcp3d.com
Technical : support@pcp3d.com
Website : www.pcp3d.com
Documentation : doc.percipio.xyz/cam/latest/en/

Statement

* Data mentioned in this document is subject to change without notice.
* The data mentioned in this document may vary due to environmental factors and other reasons. Percipio does not assume any consequences arising therefrom.

YouTube