

Kowa External Control Library Python wrapper

v1.6.1



Kowa Optronics Co., Ltd.

1. INTRODUCTION

KowaExternalControlLibrary is a control library for external devices connected to Kowa Optronics Co., Ltd. cameras. The following document describes the functions by wrapped for python.

2. Kowa External Control Library

This chapter describes the functions of the Kowa External Control python-wrapped library. The following example describes and explains the function:

Item	Contents
Function	A short description of the function.
Syntax	The function's syntax description in Python.
Parameter	Description of function parameters.
Description	Description of the purpose and use of the function.
Return	Here you find the type and range of the return values.
Reference	A list of related routines.

2-1. REMOTEFOCUSUNIT CONTROL FUNCTIONS

2-1-1. getFWVersion

- Function

Get the firmware version of the Remote Focus Unit.

- Syntax

```
getFWVersion(camera) -> int
```

- Parameters

- `camera`: Connected camera.

- Description

This function gets the firmware version of the remote focus unit. This function allows you to check if the Remote Focus Unit is properly connected.

- Return

Return the firmware version of the Remote Focus Unit. Return the current version if the communication was successful, or 0 if it was not successful.

2-1-2. addStepCount

- Function

Moves the focus position from the present position to NEAR position.

- Syntax

```
addStepCount(camera, step_count: int)
```

- Parameters

- `camera`: Connected camera.
- `step_count`: It specifies the distance. If the end of the focus position is exceeded, the amount is ignored.

- Description

It moves the focus position to NEAR by the specified amount from the current position.

- Return

none

2-1-3. subStepCount

- Function

Moves the focus position from the present position to FAR position.

- Syntax

```
subStepCount(camera, step_count: int)
```

- Parameters

- `camera`: Connected camera.
- `step_count`: It specifies the distance. If the end of the focus position is exceeded, the amount is ignored.

- Description

It moves the focus position to FAR by the specified amount from the current position.

- Return

none

2-1-4. setPPS

- Function

Change the motor rotation speed.

- Syntax

```
setPPS(camera, pps: int)
```

- Parameters

- `camera`: Connected camera.
- `pps`: It specifies how many pulses per second are sent to the motor.

- Description

This function sets the motor speed (pulse/sec). Use it when changing from the default value or the current setting value.

Check the hardware specifications for the optimal values. Specifying a value that is too large may cause a step-out.

- Return

none

- Reference

getPPS

2-1-5. getPPS

- Function

Gets the current rotation speed of the motor.

- Syntax

```
getPPS(camera) -> int
```

- Parameters

- `camera`: Connected camera.

- Description

This function gets the current motor speed (pulse/sec).

- Return

Return the number of pulses per second to send to the present motor.

- Reference

setPPS

2-1-6. setMotorPhase

- Function

Change the motor excitation method.

- Syntax

```
setMotorPhase(camera, motor_phase: int)
```

- Parameters

- `camera`: Connected camera.
- `motor_phase`: It specifies the motor energization by specifying an enumerator for the motor's energization method (refer to the "Enumerator" section and

KowaExternalControlLib.h in this document for defining the enumerator "phase_type").

- Description

This function sets the motor excitation method (2-phase excitation or 1-2-phase excitation). Use it when changing from the default value or the current setting value. To move to the target focus position faster, set to 2-phase excitation. To adjust the focus position more precisely, set 1-2 phase excitation.

- Return

none

- Reference

getMotorPhase

2-1-7. getMotorPhase

- Function

Gets the current motor activation method.

- Syntax

```
getMotorPhase(camera) -> int
```

- Parameters

- `camera`: Connected camera.

- Description

This function acquires the currently set motor excitation method (2-phase excitation or 1-2-phase excitation).

- Return

Return an enumerator of the motor's energization scheme (refer to the "Enumerators" section and KowaExternalControlLib.h in this document for defining the enumerator "phase_type").

- Reference

setMotorPhase

2-1-8. setOrigin

- Function

Returns the focus position to the origin.

- Syntax

```
setOrigin(camera)
```

- Parameters

- `camera`: Connected camera.

- Description

This function sends a command to return the focus position to the origin.

- Return

none

2-1-9. forceStop

- Function

Forcibly stops the motor.

- Syntax

```
forceStop(camera)
```

- Parameters

- `camera`: Connected camera.

- Description

This function sends a command to forcibly stop the motor operation.

- Return

none

2-1-10. getInfo

- Function

Acquires the current status of the Remote Focus Unit.

- Syntax

```
getInfo(camera) -> Info
```

- Parameters

- `camera`: Connected camera.

- Description

This function gets the current state of the remote focus unit as a structure.
Use the functions described in "Reference" below to obtain information individually.

- Return

Return the out-of-step condition, motor operation status, focus position, home return operation status, and the number of slits that have passed since starting from the origin as a structure (For the definition of structure "RemoteFocusInfo", refer to "Structure" section and KowaExternalControlLib.h of this manual).

- Reference

```
getStepoutState  
isMotorBusy  
getStepCount  
getOriginState  
getSlitLocation
```

2-1-11. getFlags

- Function

Gets the error flag of the remote focus module.

- Syntax


```
getFlags(camera) -> int
```

- Parameters

- `camera`: Connected camera.

- Description

This function gets the remote focus unit error flags.

- Return

Returns an error flag (bit[0] 0:Normal 1:step-out, bit[4] 0:Normal 1:Focus Position Limit).

- Reference

`getInfo`

2-1-12. isMotorBusy

- Function

Gets the operating status of the motor.

- Syntax

```
isMotorBusy(camera) -> bool
```

- Parameters

- `camera`: Connected camera.

- Description

This function obtains the operating status of the motor.

- Return

Returns the motor operating status (bit[0] 0:IDLE 1:BUSY).

- Reference

`getInfo`

2-1-13. getStepCount

- Function

Gets the current focus position.

- Syntax

```
getStepCount(camera) -> int
```

- Parameters

- `camera`: Connected camera.

- Description

This function gets the current focus position.

- Return

This function returns the focus position from the origin.

- Reference

`getInfo`

2-1-14. getOriginState

- Function

Acquires the operation status of origin.

- Syntax

```
getOriginState(camera) -> int
```

- Parameters

- `camera`: Connected camera.

- Description

This function acquires the operation status of origin return.

Check the status with this function during the Origin Return operation, and perform the next operation after the Origin Return operation is completed.

- Return

Returns the enumerator corresponding to the status of the return to origin operation (see the "Enumerator" section and KowaExternalControlLib.h of this document for the definition of the enumerator "origin_state_type").

- Reference

getInfo

2-1-15. getSlitLocation

- Function

Obtains the number that has passed through the slit inside the unit.

- Syntax

```
getSlitLocation(camera) -> int
```

- Parameters

- `camera`: Connected camera.

- Description

The remote focus unit has a slit inside, which is used as a reference to return to the home position.

This function obtains the number of slits that have passed from the origin.

This function is intended for maintenance and does not need to be called when using the Remote Focus Unit.

- Return

Returns the number of slits that have passed from the origin.

- Reference

getInfo

2-2. PWM CONTROL FUNCTION

2-2-1. setParameter

- Function

Set the parameters for PWM control.

- Syntax

```
setParameter(camera, mode: int, delay: int, width: int, duty:
int)
```

- Parameters

- `camera`: Connected camera.
- `mode`: It specifies the triggering method (0:MANUAL 1:STROBE).
- `delay`: It specifies how long the flash will fire after a STROBE trigger (us).
- `width`: It specifies the duration of STROBE output (us).
- `duty`: It specifies PWM control duty cycle in STROBE from 6% (0) to 100% (15) in 16 steps.

- Description

It sets the parameters for PWM control.

- Return

none

- Reference

`getParameter`

2-2-2. getParameter

- Function

Returns PWM control parameters.

- Syntax

```
getParameter(camera) -> tuple[int, int, int, int]
# [mode, delay, width, duty]
```

- Parameters

- `camera`: Connected camera.

- Description

It retrieves the parameters for PWM control.

- Return

- `mode`: Return the configured triggering method (0:MANUAL 1:STROBE).
- `delay`: Return the time from the trigger to the flash in STROBE mode (us).
- `width`: Return the duration when STROBE is active (us).
- `duty`: Return the duty cycle of PWM control set in STROBE from 0 (6%) to 15 (100%) in 16 steps.

- Reference

`setParameter`

2-2-3. setManualCtrl

- Function

Sends a command to turn PWM control on/off (MANUAL only).

- Syntax

```
setManualCtrl(camera, enable: bool)
```

- Parameters

- `camera`: Connected camera.

- `enable`: It specifies ON/OFF of PWM (0:OFF 1:ON).

- Description

It sends a command to turn PWM control on/off.

- Return

none

2-3. STRUCTURES

2-3-1. Info

flags:

Error flag (bit[0]:Step-out flag bit[4]:Focus position limit flag)

is_busy:

Operating status of the motor (0:IDLE 1:BUSY)

step_count:

Current focus position

origin_state:

Origin return operation status (refer to the "Enumerator" section and KowaExternalControlLib.h in this document for the enumerator "origin_state_type" definition)

slit_location:

Passing number of slits (see `RemoteFocusgetSlitLocation` function for slits)

2-4. ENUMERATOR

2-4-1. phase_type

Motor excitation method

Name	Value	Description
pt_full_step	0	2-phase Excitation
pt_half_step	1	1-2 Phase Excitation

2-4-2. origin_state_type

Origin return operation status



Name	Value	Description
ost_not_implemented	0	Not performed
ost_complete	1	Complete
ost_operating	2	Running
ost_error	3	Failed