
PyBuildit API Reference (Ver.1.0.1)

Smart Robotics Co.Ltd

2020 年 03 月 22 日

Contents:

第 1 章	pybuildit package	1
1.1	Submodules	1
	Python モジュール索引	17

第 1 章

pybuildit package

1.1 Submodules

`pybuildit.const.state2str(s)`

状態 ID を文字列に変換する

`pybuildit.const.faults2str(s)`

フォルト情報を文字列に変換する

`pybuildit.const.paramid2str(s)`

`pybuildit.lib.to_deg(rawPosition)`

位置情報の単位を [360/65536 * 度] から [度] に変換する

`pybuildit.lib.from_deg(deg)`

位置情報の単位を [度] から [360/65536 * 度] に変換する

`pybuildit.lib.to_rpm(rawVelocity)`

速度情報の単位を [1/100 * rpm] から [rpm] に変換する

`pybuildit.lib.from_rpm(rpm)`

速度情報の単位を [rpm] から [1/100 * rpm] に変換する

exception `pybuildit.lib.MCPError`

ベースクラス: `exceptions.Exception`

モーター制御プロトコルエラー用例外ベースクラス

exception `pybuildit.lib.MCPTransportError`

ベースクラス: `pybuildit.lib.MCPError`

通信層でのエラー用例外ベースクラス

exception `pybuildit.lib.MCPApplicationError`

ベースクラス: `pybuildit.lib.MCPError`

アプリケーション層でのエラー用例外ベースクラス

exception `pybuildit.lib.InvalidFormatError`

ベースクラス: `pybuildit.lib.MCPTransportError`

不正なメッセージフォーマット例外

exception `pybuildit.lib.InvalidCRCError`

ベースクラス: `pybuildit.lib.MCPTransportError`

不正な CRC 例外

exception `pybuildit.lib.UnexpectedMessageTypeError`

ベースクラス: `pybuildit.lib.MCPTransportError`

予想外のメッセージタイプ受信例外

exception `pybuildit.lib.InvalidCommandPayloadSizeError`

ベースクラス: `pybuildit.lib.MCPTransportError`

不正なペイロードサイズ例外

exception `pybuildit.lib.TimeoutError` (*e*)

ベースクラス: `pybuildit.lib.MCPTransportError`

タイムアウト例外

exception `pybuildit.lib.InvalidMessageTypeError`

ベースクラス: `pybuildit.lib.MCPApplicationError`

不正なメッセージタイプ例外

exception `pybuildit.lib.InvalidCommandPayloadError`

ベースクラス: `pybuildit.lib.MCPApplicationError`

不正なペイロード例外

exception `pybuildit.lib.InvalidOperationError` (*st*)

ベースクラス: `pybuildit.lib.MCPApplicationError`

不正な操作例外

exception `pybuildit.lib.OutOfPositionLimitError`

ベースクラス: `pybuildit.lib.MCPApplicationError`

位置範囲制限外エラー

exception `pybuildit.lib.OtherError(e)`

ベースクラス: `pybuildit.lib.MCPApplicationError`

その他の例外

exception `pybuildit.lib.WaitUntilTimeoutError(e)`

ベースクラス: `exceptions.Exception`

`wait_until` メソッド用タイムアウト例外

class `pybuildit.lib.MCPStatus(status)`

MCP status を抽象化したクラス

unnotified_error()

未通知のエラーが発生していれば 1、そうでなければ 0 を返す

state()

Buildit Actuator の状態を返す

str_state()

Buildit Actuator の状態を文字列で返す

class `pybuildit.lib.ServoStatus(qssr)`

query servo status コマンドの結果を抽象化したクラス

position (`unit='raw'`)

センサで計測された位置を指定された単位に換算して返す

パラメタ `unit` (`str`, `default "raw"`) - "raw": [360/65536 * 度] "deg": [度]

velocity (`unit='raw'`)

センサで計測された速度を指定された単位に換算して返す

パラメタ `unit` (`str`, `default "raw"`) - "raw": [1/100 * rpm] "rpm": [rpm]

current()

センサで計測された電流値 [mA] を返す

ref()

現在の制御指令値を返す。ただし、制御状態でなければ 0 を返す

- 位置制御の場合は [360/65536 * 度]
- 速度制御の場合は [1/100 * rpm]

temperature()

センサで計測された温度 [摂氏度] を返す

faults ()

フォルト値 (発生したフォルトに応じた値の論理和) を返す

```
SERVO_FAULT_FOC_DURATION    = 0x0001  SERVO_FAULT_OVER_VOLT    = 0x0002
SERVO_FAULT_UNDER_VOLT     = 0x0004  SERVO_FAULT_OVER_TEMP   = 0x0008
SERVO_FAULT_BREAK_IN      = 0x0040  SERVO_FAULT_STOP_CONTROL_ERROR = 0x0100
SERVO_FAULT_STOP_TIMEOUT  = 0x0200  SERVO_FAULT_EXTERNAL    = 0x0800
```

tuple ()

位置、速度、電流値、指令値、温度、フォルトのタプルを返す

class pybuildit.lib.**Buildit** (port='/dev/ttyUSB0', baud=115200, timeout_ms=3000)

ベースクラス: object

モーター制御プロトコルをカプセル化し、Buildit Actuator と通信する為のクラス

サンプル

```
>>> from pybuildit import *
>>>
>>> buildit = Buildit(port="/dev/ttyUSB0", timeout_ms=3000) #for Linux
>>> buildit = Buildit(port="COM8", timeout_ms=3000) #for Win
>>>
>>> deviceId = 1
>>>
>>> qss = buildit.query_servo_status(deviceId)
>>>
>>> print("position[deg]: ", qss.position(unit="deg"))
>>> print("velocity[rpm]: ", qss.velocity(unit="rpm"))
>>> print("temperature[ ]: ", qss.temperature())
>>> print("state: ", buildit.last_mcp_status().str_state())
>>>
>>> buildit.force_ready(deviceId)
>>> buildit.set_ref_velocity(deviceId, from_rpm(42.5))
>>> buildit.set_ref_position(deviceId, from_deg(180))
```

is_open ()

シリアルポートがオープンされていれば True、そうでなければ False を返す

open (port='/dev/ttyUSB0', baud=115200)

シリアルポートをオープンする

パラメタ

- **baud** (int, default 115200) – ボーレート
- **port** (str, default "/dev/ttyUSB0") – シリアルポートの名前 (例: Linux なら /dev/ttyUSB0, Windows なら COM8 等)

close()

シリアルポートをクローズする

last_mcp_status()

最新の応答に含まれる MCPStatus を返します (ただし、一度も通信していなければ None)

wait_until (devId, pred, timeout_ms=-1)

Buildit Actuator がユーザーが指定した状態になるか、タイムアウトするまでブロックする。ブロック解除条件は query_servo_status(devId) の結果が pred 関数の戻り値を True にすることである。

サンプル

位置が 0 より大きくなるまで待つ例

```
>>> buildit.wait_until(devId, lambda state, query_status_result: query_status_
↳result.position() > 0)
```

パラメタ

- **devId** (*int*) – デバイス ID
- **pred** (*function*) – 第一引数として Buildit Actuator の状態、第二引数として query_servo_status の結果を受け取って True もしくは False を返す関数
- **timeout_ms** (*int*) – タイムアウト値 (ミリ秒)。ただし、タイムアウト値に負の値を指定した場合はタイムアウト処理は行われない

wait_until_state (devId, targetState, timeout_ms=-1)

devId で指定された Buildit Actuator の状態遷移完了をポーリングで待つ

パラメタ

- **devId** (*int*) – デバイス ID
- **targetState** (*int*) – 状態 ID
- **timeout_ms** (*int*) – タイムアウト値 (ミリ秒)

wait_until_stop (dev_id, timeout_ms=-1, vel_thres=10, vel_unit='raw')

devId で指定された位置制御状態にある Buildit Actuator が停止されるまでブロックする。既に停止している状態ではブロックされない。例えばレディ状態から回転させる為に位置制御指令値を設定した直後にこの関数を実行してもブロックされるとは限らない。

パラメタ

- **devId** (*int*) – デバイス ID

- `pos (int)` – 停止予定位置
- `timeout_ms (int)` – タイムアウト値 (ミリ秒)
- `vel_thres (float)` – 停止速度の許容誤差
- `vel_unit (str, default "raw")` – `vel_thres` の単位 "raw": [360/65536 * 度] "deg": [度]

`wait_until_stop_at (dev_id, pos, timeout_ms=-1, pos_thres=10, pos_unit='raw', vel_thres=10, vel_unit='raw')`

`devId` で指定された位置制御状態にある Buildit Actuator が `pos` で指定された位置で停止されるまでブロックする

パラメタ

- `devId (int)` – デバイス ID
- `pos (int)` – 停止予定位置
- `timeout_ms (int)` – タイムアウト値 (ミリ秒)
- `pos_thres (float)` – 停止位置の許容誤差
- `pos_unit (str, default "raw")` – `pos` 及び `pos_thres` の単位 "raw": [360/65536 * 度] "deg": [度]
- `vel_thres (float)` – 停止速度の許容誤差
- `vel_unit (str, default "raw")` – `vel_thres` の単位 "raw": [360/65536 * 度] "deg": [度]

`query_servo_status (devId)`

`devId` で指定された Buildit Actuator の状態を問い合わせる

パラメタ `devId (int)` – デバイス ID

戻り値

戻り値の型 `ServoStatus` のオブジェクトを返す

`hold (devId)`

`devId` で指定された、ブレーキ保持状態、ブレーキ解除状態、レディ状態のいずれかの状態にある Buildit Actuator をブレーキ保持状態に遷移させる

パラメタ `devId (int)` – デバイス ID

`ready (devId)`

`devId` で指定された、ブレーキ保持状態、ブレーキ解除状態、レディ状態のいずれかの状態にある Buildit Actuator をレディ状態に遷移させる

パラメタ **devId** (*int*) – デバイス ID

force_hold (*devId*)

devId で指定された、Buildit Actuator をブレーキ保持状態に遷移させる

パラメタ **devId** (*int*) – デバイス ID

force_ready (*devId*)

devId で指定された、Buildit Actuator をレディ状態に遷移させる

パラメタ **devId** (*int*) – デバイス ID

force_free (*devId*)

devId で指定された、Buildit Actuator をブレーキ解除状態に遷移させる

パラメタ **devId** (*int*) – デバイス ID

free (*devId*)

devId で指定された、ブレーキ保持状態、ブレーキ解除状態、レディ状態のいずれかの状態にある Buildit Actuator をブレーキ解除状態に遷移させる Buildit Actuator を free 状態に遷移させる

パラメタ **devId** (*int*) – デバイス ID

stop (*devId*, *timeout_ms=500*)

devId で指定された、電流制御状態、速度制御状態、位置制御状態、レディ状態のいずれかの状態にある Buildit Actuator に対して保護停止を行う。保護停止が成功した場合、Buildit Actuator はレディ状態に遷移する。

パラメタ

- **devId** (*int*) – デバイス ID
- **timeout_ms** (*int*) – タイムアウト [ミリ秒]

clear_fault (*devId*)

devId で指定された、フォルト状態にある Buildit Actuator に対して復帰を試みる。ブレーキ解除フォルト状態であればブレーキ解除状態に ブレーキ保持フォルト状態であればブレーキ保持状態に遷移する。

パラメタ **devId** (*int*) – デバイス ID

reset_rotation (*devId*, *rot=0*)

devId で指定された、レディ状態、位置制御状態のいずれでもない Buildit Actuator の電源投入後のの累計回転数を初期化する。

パラメタ

- **devId** (*int*) – デバイス ID
- **rot** (*int*) – リセット後の回転数

set_ref_current (*devId, cur*)

devId で指定された、電流制御状態、速度制御状態、位置制御状態、レディ状態のいずれかの状態にある Buildit Actuator を電流指令値を設定した上で電流制御状態に遷移させる

パラメタ

- **devId** (*int*) – デバイス ID
- **cur** (*int*) – 電流指令値 [mA]

戻り値 **cur** – センサで計測された電流値 [mA]

戻り値の型 `int`

get_ref_current (*devId*)

devId で指定された、電流制御状態にある Buildit Actuator の電流指令値 [mA] を取得する

get_current_KP (*devId*)

devId で指定された Buildit Actuator の電流制御用比例ゲインを取得する

get_current_KI (*devId*)

devId で指定された Buildit Actuator の電流制御用積分ゲインを取得する

get_current_max_Iterm (*devId*)

devId で指定された Buildit Actuator の電流制御用積分項の上限値を取得する (非推奨)

get_current_min_Iterm (*devId*)

devId で指定された Buildit Actuator の電流制御用積分項の下限値を取得する (非推奨)

set_current_max_limit (*devId, v*)

devId で指定された Buildit Actuator の電流指令値の上限値を設定する (-5000 ~ 5000)

get_current_max_limit (*devId*)

devId で指定された Buildit Actuator の電流指令値の上限値を取得する

set_current_min_limit (*devId, v*)

devId で指定された Buildit Actuator の電流指令値の下限値を設定する (-5000 ~ 5000)

get_current_min_limit (*devId*)

devId で指定された Buildit Actuator の電流指令値の下限値を取得する

set_ref_velocity (*devId, vel, unit='raw'*)

devId で指定された、電流制御状態、速度制御状態、位置制御状態、レディ状態のいずれかの状態にある Buildit Actuator を速度指令値 `vel` を設定して速度制御状態に遷移させる

パラメタ

- **devId** (*int*) – デバイス ID
- **vel** (*int* もしくは *float*) – 速度指令値

- **unit** (*str*) – 引数と戻り値の単位

"raw": [1/100 * rpm] "rpm": [rpm]

戻り値 **vel** – センサで計測された速度

戻り値の型 **int**

get_ref_velocity (*devId*, *unit='raw'*)

devId で指定された、速度制御状態にある Buildit Actuator の速度指令値を取得する

パラメタ

- **devId** (*int*) – デバイス ID

- **unit** (*str*) – 戻り値の単位

"raw": [1/100 * rpm] "rpm": [rpm]

set_velocity_KP (*devId*, *v*)

devId で指定された Buildit Actuator の速度制御用比例ゲインを設定する

get_velocity_KP (*devId*)

devId で指定された Buildit Actuator の速度制御用比例ゲインを取得する

set_velocity_KI (*devId*, *v*)

devId で指定された Buildit Actuator の速度制御用積分ゲインを設定する

get_velocity_KI (*devId*)

devId で指定された Buildit Actuator の速度制御用積分ゲインを取得する

set_velocity_KD (*devId*, *v*)

devId で指定された Buildit Actuator の速度制御用微分ゲインを設定する

get_velocity_KD (*devId*)

devId で指定された Buildit Actuator の速度制御用積分微分ゲインを取得する

set_velocity_max_Iterm (*devId*, *v*)

devId で指定された Buildit Actuator の速度制御用積分項の上限値を設定する (非推奨)

get_velocity_max_Iterm (*devId*)

devId で指定された Buildit Actuator の速度制御用積分項の上限値を取得する (非推奨)

set_velocity_min_Iterm (*devId*, *v*)

devId で指定された Buildit Actuator の速度制御用積分項の下限値を設定する (非推奨)

get_velocity_min_Iterm (*devId*)

devId で指定された Buildit Actuator の速度制御用積分項の下限値を取得する (非推奨)

set_velocity_max_limit (*devId*, *v*, *unit='raw'*)

devId で指定された Buildit Actuator の速度指令値の上限値を設定する (raw: -5000 ~ 5000)

パラメタ

- **devId** (*int*) – デバイス ID
- **v** (*int* もしくは *float*) – 速度制限値
- **unit** (*str*) – 引数の単位

"raw": [1/100 * rpm] "rpm": [rpm]

get_velocity_max_limit (*devId*, *unit='raw'*)

devId で指定された Buildit Actuator の速度指令値の上限値を取得する

パラメタ

- **devId** (*int*) – デバイス ID
- **unit** (*str*) – 戻り値の単位

"raw": [1/100 * rpm] "rpm": [rpm]

set_velocity_min_limit (*devId*, *v*, *unit='raw'*)

devId で指定された Buildit Actuator の速度指令値の下限値を設定する (raw: -5000 ~ 5000)

パラメタ

- **devId** (*int*) – デバイス ID
- **v** (*int* もしくは *float*) – 位置制限値
- **unit** (*str*) – 引数の単位

"raw": [1/100 * rpm] "rpm": [rpm]

get_velocity_min_limit (*devId*, *unit='raw'*)

devId で指定された Buildit Actuator の速度指令値の下限値を取得する

パラメタ

- **devId** (*int*) – デバイス ID
- **unit** (*str*) – 戻り値の単位

"raw": [1/100 * rpm] "rpm": [rpm]

set_ref_position (*devId*, *pos*, *unit='raw'*)

devId で指定された、電流制御状態、速度制御状態、位置制御状態、レディ状態のいずれかの状態にある Buildit Actuator を位置指令値 *pos* を設定して位置制御状態に遷移させる

パラメタ

- **devId** (*int*) – デバイス ID
- **pos** (*int* もしくは *float*) – 位置指令値
- **unit** (*str*) – 引数と戻り値の単位

"raw": [360/65536 * 度] "deg": [度]

戻り値 **pos** – センサで計測された位置

戻り値の型 `int`

get_ref_position (*devId*, *unit='raw'*)

devId で指定された、位置制御状態にある Buildit Actuator の位置指令値 [360/65536 * 度] を取得する

パラメタ

- **devId** (*int*) – デバイス ID
- **unit** (*str*) – 戻り値の単位

"raw": [360/65536 * 度] "deg": [度]

set_position_KP (*devId*, *v*)

devId で指定された Buildit Actuator の位置制御用比例ゲインを設定する

get_position_KP (*devId*)

devId で指定された Buildit Actuator の位置制御用比例ゲインを取得する

set_position_KI (*devId*, *v*)

devId で指定された Buildit Actuator の位置制御用積分ゲインを設定する

get_position_KI (*devId*)

devId で指定された Buildit Actuator の位置制御用積分ゲインを取得する

set_position_KD (*devId*, *v*)

devId で指定された Buildit Actuator の位置制御用微分ゲインを設定する

get_position_KD (*devId*)

devId で指定された Buildit Actuator の位置制御用積分微分ゲインを取得する

set_position_max_Iterm (*devId*, *v*)

devId で指定された Buildit Actuator の位置制御用積分項の上限値を設定する (非推奨)

get_position_max_Iterm (*devId*)

devId で指定された Buildit Actuator の位置制御用積分項の上限値を取得する (非推奨)

set_position_min_Iterm (*devId*, *v*)

devId で指定された Buildit Actuator の位置制御用積分項の下限値を設定する (非推奨)

get_position_min_Iterm (*devId*)

devId で指定された Buildit Actuator の位置制御用積分項の下限值を取得する (非推奨)

set_position_max_limit (*devId*, *v*, *unit='raw'*)

devId で指定された Buildit Actuator の位置指令値の上限値を設定する

パラメタ

- **devId** (*int*) – デバイス ID
- **v** (*int* もしくは *float*) – 位置制限値
- **unit** (*str*) – 引数の単位

"raw": [360/65536 * 度] "deg": [度]

get_position_max_limit (*devId*, *unit='raw'*)

devId で指定された Buildit Actuator の位置指令値の上限値を取得する

パラメタ

- **devId** (*int*) – デバイス ID
- **unit** (*str*) – 戻り値の単位

"raw": [360/65536 * 度] "deg": [度]

set_position_min_limit (*devId*, *v*, *unit='raw'*)

devId で指定された Buildit Actuator の位置指令値の下限值を設定する [360/65536 * 度]

パラメタ

- **devId** (*int*) – デバイス ID
- **v** (*int* もしくは *float*) – 位置制限値
- **unit** (*str*) – 引数の単位

"raw": [360/65536 * 度] "deg": [度]

get_position_min_limit (*devId*, *unit='raw'*)

devId で指定された Buildit Actuator の位置指令値の下限值を取得する [360/65536 * 度]

パラメタ

- **devId** (*int*) – デバイス ID
- **unit** (*str*) – 戻り値の単位

"raw": [360/65536 * 度] "deg": [度]

set_position_offset (*devId*, *v*, *unit='raw'*)

devId で指定された Buildit Actuator の位置センサのユーザーオフセットを設定する [360/65536 * 度]
(16bit 符号付き整数)

get_position_offset (*devId*, *unit='raw'*)

devId で指定された Buildit Actuator の位置センサのユーザーオフセットを取得する [360/65536 * 度]

set_device_id (*devId*, *v*)

devId で指定された Buildit Actuator のデバイス ID を設定する (1 ~ 127 の整数)

get_device_id (*devId*)

devId で指定された Buildit Actuator のデバイス ID を取得する

find_device_id ()

接続中の応答可能なデバイスのうち最も小さなデバイス ID を返す

save_servo_params (*devId*, *fileName=None*)

devId で指定された Buildit Actuator のサーボパラメーターをファイルに保存する

load_servo_params (*devId*, *filepath*)

サーボパラメーターをファイルから読みだし、*devId* で指定された Buildit Actuator に反映する

save_configuration (*devId*, *fileName=None*)

devId で指定された Buildit Actuator の設定をファイルに保存する

move_with_period (*devId*, *goal*, *period_ms*, *acc=0.2*, *dec=0.2*, *pos_thres=10*, *unit='raw'*)

指定された期間で目標位置値で停止するように移動命令を発行する

PC 側のシステムのシリアル通信のレイテンシを 5ms 以下に設定すること

パラメタ

- **devId** (*int*) – デバイス ID
- **goal** (*float*) – 目標位置
- **period_ms** (*float*) – 移動時間 [ミリ秒]
- **acc** (*float*) – t1 に対する開始加速期間の割合 ([0.1, 1])
- **dec** (*float*) – t1 に対する終端減速期間の割合 ([0.1, 1])
- **pos_thres** (*float*) – 停止位置の許容誤差
- **unit** (*str*) – goal, pos_thres の単位

get_firmware_version (*devId*)

devId で指定された Buildit Actuator のファームウェアのバージョン情報を取得する

get_power_on_time (*devId*)

devId で指定された Buildit Actuator の累計通電時間を取得する

get_calibration_data (*devId*)

devId で指定された Buildit Actuator の補正値を取得する

get_position_sys_offset (*devId*)

devId で指定された Buildit Actuator の位置センサのシステムオフセットを取得する [360/65536 * 度]

get_log_info (*devId*)

devId で指定された Buildit Actuator のログ情報を取得する

パラメタ **devId** (*int*) – デバイス ID

戻り値 **readableSize** – 取得可能なログレコード数

戻り値の型 **int**

print_log (*devId*, *startIndex=0*, *readSize=-1*)

devId で指定された Buildit Actuator のログを出力する

パラメタ

- **devId** (*int*) – デバイス ID
- **startIndex** (*int*) – 取得するログレコードの開始番号
- **readSize** (*int*) – 取得するログレコードの個数

get_log (*devId*, *startIndex=0*, *readSize=-1*)

devId で指定された Buildit Actuator のログを取得する

パラメタ

- **devId** (*int*) – デバイス ID
- **startIndex** (*int*) – 取得するログレコードの開始番号
- **readSize** (*int*) – 取得するログレコードの個数

戻り値

- **startIndex** (*int*) – 取得したログレコードの開始番号
- **readSize** (*int*) – 取得したログレコードの個数
- **logRecords** (*bytes*) – 取得した ReadSize 個分のログレコード

fault (*devId*, *is_fatal=False*)

任意のフォルトを発生させる

パラメタ

- `devId (int)` – デバイス ID
- `is_fatal (bool)` – True ならシステムフォルト (モーターの通電を止めてブレーキがかかる。そして再起動しない限り LED の点灯以外は何も行わない) False ならフォルト状態に遷移させる (モーターの通電を止めてブレーキがかかる)

```
class pybuildit.unsafe.UnsafeBuildit (port='/dev/ttyUSB0',      baud=115200,      time-
                                     out_ms=3000)
```

ベースクラス: `pybuildit.lib.Buildit`

デバッグビルドファーム用拡張機能

`get_prot_stop_pin_timeout (devId)`

保護停止ピンによる停止間上限 [ms] を取得する

`get_stop_control_error_threshold (devId)`

停止異常閾値を取得する [360/65536 * 度]

`set_prot_stop_pin_timeout (devId, v)`

保護停止ピンによる停止間上限 [ms] を設定する

`set_stop_control_error_threshold (devId, v)`

停止異常閾値を設定する

`reset (devId)`

ソフトウェアリセットする

`debug (devId, mode)`

デバッグイベント発生

パラメタ

- `devId (int)` – デバイス ID
- `mode (int)` – モード (0:none, 1:position, 2:velocity)

`clear_log (devId)`

devId で指定された Buildit Actuator のログを消去する

`clear_power_on_time (devId)`

devId で指定された Buildit Actuator の累計通電時間を 0 リセットする

パラメタ `devId (int)` – デバイス ID

`set_calibration_data (devId, v)`

devId で指定された Buildit Actuator の補正值を設定する

`set_position_sys_offset (devId, v)`

devId で指定された Buildit Actuator の位置センサのシステムオフセットを設定する [360/65536 * 度]

set_current_KP (*devId*, *v*)

devId で指定された Buildit Actuator の電流制御用比例ゲインを設定する

set_current_KI (*devId*, *v*)

devId で指定された Buildit Actuator の電流制御用積分ゲインを設定する

set_current_max_Iterm (*devId*, *v*)

devId で指定された Buildit Actuator の電流制御用積分項の上限値を設定する

set_current_min_Iterm (*devId*, *v*)

devId で指定された Buildit Actuator の電流制御用積分項の下限値を設定する

Python モジュール索引

p

pybuildit, 1
pybuildit.const, 1
pybuildit.lib, 1
pybuildit.unsafe, 15
