

ServoX

Get

get_max_value

Methode

```
get_max_value() -> int
```

Parameter-Erklärung

Diese Methode hat **keine Parameter**.

Rückgabe

- `int`: Der maximal mögliche Wert, den der Servo annehmen kann. Dieser Wert definiert die **oberste Grenze** der Stellbewegung des Servos.

Allgemeine Erklärung

Die Methode gibt den **höchsten Wert des Servos** zurück, der für die Steuerung der Servo-Position relevant ist. Sie ist besonders nützlich, um **Bewegungsbereiche zu definieren**, sicherzustellen, dass keine mechanischen Grenzen überschritten werden, und die **Kalibrierung von Servos** zu unterstützen.

get_min_value

Methode

```
get_min_value() -> int
```

Parameter-Erklärung

Diese Methode hat **keine Parameter**.

Rückgabe

- `int`: Der minimal mögliche Wert, den der Servo annehmen kann. Dieser Wert definiert die **unterste Grenze** der Stellbewegung des Servos.

Allgemeine Erklärung

Die Methode gibt den **niedrigsten Wert des Servos** zurück, der für die Steuerung der Servo-Position relevant ist. Sie ist besonders nützlich, um **Bewegungsbereiche zu definieren**, mechanische Begrenzungen einzuhalten und die **Kalibrierung von Servos** sicherzustellen.

get_pos

Methode

```
get_pos() -> int
```

Parameter-Erklärung

Diese Methode hat **keine Parameter**.

Rückgabe

- `int`: Die aktuelle **Position des Servos**. Der Wert zeigt an, auf welchem Punkt innerhalb des definierten Bewegungsbereichs sich der Servo momentan befindet.

Allgemeine Erklärung

Die Methode liefert die **aktuelle Position des Servos** zurück. Sie ist nützlich, um den Servo **zu überwachen**, Bewegungsabläufe zu **kontrollieren** oder sicherzustellen, dass der Servo **präzise auf die gewünschte Position eingestellt** ist.

Set

set_pos

Methode

```
set_pos(value: int, enabler_needed: bool = True) -> None
```

Parameter-Erklärung

value

- `int`: Die gewünschte Position, auf die der Servo gesetzt werden soll. Der Wert muss innerhalb des definierten **Bewegungsbereichs** des Servos liegen.

enabler_needed

- `bool` (optional, Standard: `True`): Gibt an, ob der Servo-Port vor der Bewegung **aktiviert** werden soll.
 - `True`: Der Port wird automatisch aktiviert und nach der Bewegung deaktiviert.
 - `False`: Der Port wird nicht aktiviert, daher bewegt sich der Servo nur, wenn er zuvor manuell aktiviert wurde.

Rückgabe

- `None`: Diese Methode gibt keinen Wert zurück.

Allgemeine Erklärung

Die Methode **setzt den Servo auf die gewünschte Position**. Sie berücksichtigt optional, ob der Servo-Port automatisch aktiviert werden soll, um die Bewegung durchzuführen. Dies ist wichtig für **präzise Steuerung, Automatisierung von Bewegungsabläufen** und **Schutz des Servos**, indem sichergestellt wird, dass er nur dann bewegt wird, wenn der Port aktiv ist.

add_to_pos

Methode

```
add_to_pos(value: int, enabler_needed: bool = True) -> None
```

Parameter-Erklärung

value

- `int`: Der Wert, der zur **aktuellen Servo-Position** addiert werden soll. Positive Werte bewegen den Servo in eine Richtung, negative Werte in die andere.

enabler_needed

- `bool` (optional, Standard: `True`): Gibt an, ob der Servo-Port vor der Bewegung **aktiviert** werden soll.
 - `True`: Der Port wird automatisch aktiviert und nach der Bewegung wieder deaktiviert.
 - `False`: Der Port wird nicht aktiviert; der Servo bewegt sich nur, wenn er zuvor manuell aktiviert wurde.

Rückgabe

- `None`: Diese Methode gibt keinen Wert zurück.

Allgemeine Erklärung

Die Methode **verändert die aktuelle Position des Servos um einen bestimmten Wert**. Sie ist nützlich für **relative Bewegungen**, bei denen der Servo **schrittweise angepasst** werden soll, statt direkt auf eine absolute Position gesetzt zu werden.

range_to_pos

Methode

```
range_to_pos(value: int, multi: int = 2, disabler_needed: bool = True) -> None
```

Parameter-Erklärung

value

- `int`: Die Zielposition, auf die der Servo **am Ende der Bewegung** eingestellt werden soll.

multi

- `int` (optional, Standard: `2`): Ein **Multiplikator**, der die Geschwindigkeit der Bewegung beeinflusst.
 - Höhere Werte -> schneller, aber weniger sanft.
 - Niedrigere Werte -> langsamer, aber glatter.

disabler_needed

- `bool` (optional, Standard: `True`): Gibt an, ob der Servo-Port nach Abschluss der Bewegung **deaktiviert** werden soll.
 - `True`: Der Port wird nach der Bewegung deaktiviert.
 - `False`: Der Port bleibt aktiv.

Rückgabe

- `None`: Diese Methode gibt keinen Wert zurück.

Allgemeine Erklärung

Die Methode **bewegt den Servo sanft von der aktuellen Position zur Zielposition**, wodurch ruckartige Bewegungen vermieden werden. Die Geschwindigkeit kann über den `multi`-Parameter angepasst werden, und mit `disabler_needed` wird gesteuert, ob der Servo-Port nach der Bewegung deaktiviert wird. Dies ist besonders nützlich für **präzise, flüssige Bewegungen**, z.B. beim **greifen, schwenken oder justieren** von Mechaniken.

range_from_to_pos

Methode

```
range_from_to_pos(interval: list, multi: int = 2, disabler_needed: bool = True) -> None
```

Parameter-Erklärung

interval

- `list ([int1, int2])`: Eine Liste mit **zwei Positionen**:
 - `int1`: Startposition des Servos.
 - `int2`: Zielposition, zu der der Servo sanft bewegt werden soll.

multi

- `int` (optional, Standard: `2`): Ein **Multiplikator**, der die Geschwindigkeit der Bewegung beeinflusst.
 - Höhere Werte → schneller, aber weniger sanft.
 - Niedrigere Werte → langsamer, aber glatter.

disabler_needed

- `bool` (optional, Standard: `True`): Gibt an, ob der Servo-Port nach Abschluss der Bewegung **deaktiviert** werden soll.
 - `True`: Deaktiviert den Port.
 - `False`: Lässt den Port aktiv.

Rückgabe

- `None`: Diese Methode gibt keinen Wert zurück.

Allgemeine Erklärung

Die Methode **bewegt den Servo sanft von einer Startposition zur Zielposition**, wie in der `interval`-Liste angegeben. Die Geschwindigkeit kann über den Parameter `multi` angepasst werden, und `disabler_needed` steuert, ob der Servo-Port nach der Bewegung deaktiviert wird. Dies ist besonders nützlich für **kontrollierte, flüssige Bewegungen**, z. B. beim **präzisen Schwenken, Greifen oder Justieren** von Mechaniken, ohne ruckartige Bewegungen zu erzeugen.