

com._604robotics.utils

Class DualVictor

java.lang.Object
com._604robotics.utils.DualVictor

All Implemented Interfaces:

PIDOutput

```
public class DualVictor
extends Object
implements PIDOutput
```

Control two Victors like they're one. Useful for PID controllers. Also, it's springable (see SpringableVictor).

Constructor Summary

Constructors

Constructor and Description
DualVictor (int leftPort, int rightPort) Initialize a DualVictor with a left and a right PWM port.
DualVictor (int leftSlot, int leftPort, int rightSlot, int rightPort) Initializes a DualVictor with left and right slot and PWM port.
DualVictor (Victor leftVictor, Victor rightVictor) Initializes a DualVictor with left and right slot and PWM port.

Method Summary

Methods

Modifier and Type	Method and Description
double	get () Checks the current power the Victors are set to.
boolean	getSprung () Has the victor been sprung?
void	pidWrite (double output) Function to hook into the PIDController.
void	reload () If the Victor has been sprung, unspring it; if not, set the output to 0.
void	set (double speed) Sets the power of the Victors.
void	setController (PIDController controller) Sets the PIDController for this DualVictor, if there is one.
void	setDeadband (double lowerDeadband, double upperDeadband) Sets the deadband for the DualVictor.
void	setLeftInversion (boolean inversion) Sets the inversion for the "left" Victor.
void	setRightInversion (boolean inversion) Sets the inversion for the "right" Victor.
void	setSafetyEnabled (boolean enabled) Sets whether or not safety is enabled.
void	spring () Springs the victor.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DualVictor

```
public DualVictor(int leftPort,
                  int rightPort)
```

Initialize a DualVictor with a left and a right PWM port.

Parameters:

- leftPort - The PWM port of the "left" Victor.
- rightPort - The PWM port of the "right" Victor.

DualVictor

```
public DualVictor(int leftSlot,
                  int leftPort,
                  int rightSlot,
                  int rightPort)
```

Initializes a DualVictor with left and right slot and PWM port.

Parameters:

- leftSlot - The slot the "left" Victor is plugged into.
- leftPort - The PWM port of the "left" Victor.
- rightSlot - The slot the "right" Victor is plugged into.
- rightPort - The PWM port of the "right" Victor.

DualVictor

```
public DualVictor(Victor leftVictor,
                  Victor rightVictor)
```

Initializes a DualVictor with left and right slot and PWM port.

Parameters:

- leftVictor - The "left" Victor.
- rightVictor - The "right" Victor.

Method Detail

getSprung

```
public boolean getSprung()
```

Has the victor been sprung?

Returns:

- Whether or not the victor has been sprung.

spring

```
public void spring()
```

Springs the victor.

setLeftInversion

```
public void setLeftInversion(boolean inversion)
```

Sets the inversion for the "left" Victor.

Parameters:

- inversion - Is it inverted?

setRightInversion

```
public void setRightInversion(boolean inversion)
```

Sets the inversion for the "right" Victor.

Parameters:

`inversion` - Is it inverted?

get

```
public double get()
```

Checks the current power the Victors are set to.

Returns:

The current power the Victors are set to.

set

```
public void set(double speed)
```

Sets the power of the Victors.

Parameters:

`speed` - The speed to set.

pidWrite

```
public void pidWrite(double output)
```

Function to hook into the PIDController. Sets the power of the Victors.

Specified by:

`pidWrite` in interface `PIDOutput`

Parameters:

`output` - The speed to set.

setDeadband

```
public void setDeadband(double lowerDeadband,  
                        double upperDeadband)
```

Sets the deadband for the DualVictor. The default is no deadband.

Parameters:

`lowerDeadband` - The lower bound of the deadband.

`upperDeadband` - The upper bound of the deadband.

setSafetyEnabled

```
public void setSafetyEnabled(boolean enabled)
```

Sets whether or not safety is enabled.

Parameters:

`enabled` - Whether or not safety is enabled.

reload

```
public void reload()
```

If the Victor has been sprung, unsprung it; if not, set the output to 0.

setController

```
public void setController(PIDController controller)
```

Sets the PIDController for this DualVictor, if there is one. If the PIDController is enabled, reload will assume it's updating it, and won't reset the output to 0.

Parameters:

controller - The PIDController for this DualVictor.

Overview Package **Class** Tree Deprecated Index Help

Prev Class **Next Class** Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method