

Hierarchy For All Packages

Package Hierarchies:

com._604robotics.robot2012, com._604robotics.robot2012.aiming, com._604robotics.robot2012.autonomous, com._604robotics.robot2012.balancing, com._604robotics.robot2012.camera, com._604robotics.robot2012.configuration, com._604robotics.robot2012.machine, com._604robotics.robot2012.physics, com._604robotics.robot2012.rotation, com._604robotics.robot2012.vision, com._604robotics.utils, edu.wpi.first.wpilibj, frc.vision

Class Hierarchy

- java.lang.**Object**
 - com._604robotics.robot2012.aiming.**Aiming**
 - com._604robotics.robot2012.balancing.**Balancing**
 - com._604robotics.robot2012.physics.**BallFireInfo**
 - com._604robotics.utils.**DeadbandedSource** (implements edu.wpi.first.wpilibj.PIDSource)
 - com._604robotics.utils.**DuaVictor** (implements edu.wpi.first.wpilibj.PIDOutput)
 - com._604robotics.robot2012.rotation.**DummyRotationProvider** (implements com._604robotics.robot2012.rotation.RotationProvider)
 - com._604robotics.robot2012.machine.**ElevatorMachine** (implements com._604robotics.robot2012.machine.StrangeMachine)
 - com._604robotics.utils.**LinearController**
 - javax.microedition.midlet.**MIDlet**
 - edu.wpi.first.wpilibj.**RobotBase**
 - edu.wpi.first.wpilibj.**SimpleRobot**
 - com._604robotics.robot2012.**Robot2012Orange**
 - com._604robotics.robot2012.rotation.**NaiveRotationProvider** (implements com._604robotics.robot2012.rotation.RotationProvider)
 - com._604robotics.robot2012.physics.**Physics**
 - com._604robotics.robot2012.machine.**PickupMachine** (implements com._604robotics.robot2012.machine.StrangeMachine)
 - edu.wpi.first.wpilibj.**PIDController** (implements edu.wpi.first.wpilibj.parsing.IUtility)
 - com._604robotics.utils.**ConvertingPIDController**
 - com._604robotics.utils.**UpDownPIDController**
 - com._604robotics.robot2012.autonomous.**PIDDriveEncoderDifference** (implements edu.wpi.first.wpilibj.PIDSource)
 - com._604robotics.robot2012.autonomous.**PIDDriveEncoderOutput** (implements edu.wpi.first.wpilibj.PIDOutput)
 - com._604robotics.robot2012.autonomous.**PIDDriveGyro** (implements edu.wpi.first.wpilibj.PIDOutput)
 - com._604robotics.robot2012.aiming.**Point2d**
 - com._604robotics.robot2012.aiming.**Point3d**
 - com._604robotics.robot2012.vision.**Point3d**
 - com._604robotics.robot2012.aiming.**PointAndAngle3d**
 - com._604robotics.robot2012.camera.**RemoteCameraTCP** (implements com._604robotics.robot2012.camera.CameraInterface)
 - edu.wpi.first.wpilibj.**SensorBase**
 - edu.wpi.first.wpilibj.**Encoder** (implements edu.wpi.first.wpilibj.CounterBase, edu.wpi.first.wpilibj.parsing.ISensor, edu.wpi.first.wpilibj.PIDSource)
 - com._604robotics.utils.**EncoderOffset**
 - com._604robotics.utils.**EncoderPIDSource**
 - edu.wpi.first.wpilibj.**Gyro** (implements edu.wpi.first.wpilibj.parsing.ISensor, edu.wpi.first.wpilibj.PIDSource)
 - com._604robotics.utils.**Gyro360** (implements edu.wpi.first.wpilibj.PIDSource)
 - edu.wpi.first.wpilibj.**GyroHax**
 - com._604robotics.utils.**CompensatingGyro**
 - edu.wpi.first.wpilibj.**PWM**
 - edu.wpi.first.wpilibj.**SafePWM** (implements edu.wpi.first.wpilibj.MotorSafety)
 - edu.wpi.first.wpilibj.**Victor** (implements edu.wpi.first.wpilibj.parsing.IDeviceController, edu.wpi.first.wpilibj.SpeedController)
 - com._604robotics.utils.**SpringableVictor**
 - edu.wpi.first.wpilibj.**Relay** (implements edu.wpi.first.wpilibj.parsing.IDeviceController)
 - com._604robotics.utils.**SpringableRelay**
 - edu.wpi.first.wpilibj.**SolenoidBase** (implements edu.wpi.first.wpilibj.parsing.IDeviceController)
 - edu.wpi.first.wpilibj.**DoubleSolenoid**
 - com._604robotics.utils.**SpringableDoubleSolenoid**
 - com._604robotics.robot2012.physics.**ShooterAnglePick**
 - com._604robotics.robot2012.machine.**ShooterMachine** (implements com._604robotics.robot2012.machine.StrangeMachine)
 - com._604robotics.robot2012.rotation.**SlightlySmarterRotationProvider** (implements com._604robotics.robot2012.rotation.RotationProvider)
 - com._604robotics.robot2012.rotation.**SlowbroRotationProvider** (implements com._604robotics.robot2012.rotation.RotationProvider)
 - com._604robotics.robot2012.vision.**Target**
 - frc.vision.**Target**
 - com._604robotics.robot2012.machine.**TurretMachine** (implements com._604robotics.robot2012.machine.StrangeMachine)
 - com._604robotics.utils.**UpDownPIDController.Gains**
 - com._604robotics.utils.**VelocityController**
 - com._604robotics.utils.**XboxController**

Interface Hierarchy

- com._604robotics.robot2012.configuration.**ActuatorConfiguration**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.ELEVATOR**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.ELEVATOR.DEADBAND**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.ELEVATOR.TOLERANCE**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.RING_LIGHT**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.SOLENOID_HOPPER**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.SOLENOID_PICKUP**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.SOLENOID_SHIFTER**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.SOLENOID_SHOOTER**
- com._604robotics.robot2012.configuration.**ActuatorConfiguration.TURRET_POSITION**
- com._604robotics.robot2012.configuration.**AutonomousConfiguration**
- com._604robotics.robot2012.configuration.**ButtonConfiguration**
- com._604robotics.robot2012.configuration.**ButtonConfiguration.Driver**
- com._604robotics.robot2012.configuration.**ButtonConfiguration.Manipulator**
- com._604robotics.robot2012.configuration.**ButtonConfiguration.Manipulator.Elevator**

- `com._604robotics.robot2012.camera.CameraInterface`
- `com._604robotics.robot2012.machine.ElevatorMachine.ElevatorState`
- `com._604robotics.robot2012.machine.PickupMachine.PickupState`
- `com._604robotics.robot2012.configuration.PortConfiguration`
- `com._604robotics.robot2012.configuration.PortConfiguration.Controllers`
- `com._604robotics.robot2012.configuration.PortConfiguration.Encoders`
- `com._604robotics.robot2012.configuration.PortConfiguration.Encoders.Drive`
- `com._604robotics.robot2012.configuration.PortConfiguration.Motors`
- `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics`
- `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.HOPPER_SOLENOID`
- `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.PICKUP_SOLENOID`
- `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHIFTER_SOLENOID`
- `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHOOTER_SOLENOID`
- `com._604robotics.robot2012.configuration.PortConfiguration.Relays`
- `com._604robotics.robot2012.configuration.PortConfiguration.Sensors`
- `com._604robotics.robot2012.rotation.RotationProvider`
- `com._604robotics.robot2012.configuration.SensorConfiguration`
- `com._604robotics.robot2012.configuration.SensorConfiguration.Encoders`
- `com._604robotics.robot2012.machine.ShooterMachine.ShooterState`
- `com._604robotics.robot2012.machine.StrangeMachine`
- `com._604robotics.robot2012.machine.TurretMachine.TurretState`
- `com._604robotics.utils.XboxController.Axis`
- `com._604robotics.utils.XboxController.Button`
- `com._604robotics.utils.XboxController.Button.DPad`
- `com._604robotics.utils.XboxController.Stick`

Overview Package Class **Tree** Deprecated Index Help

Prev Next Frames No Frames All Classes

All Classes

ActuatorConfiguration
ActuatorConfiguration.ELEVATOR
ActuatorConfiguration.ELEVATOR.DEADBAND
ActuatorConfiguration.ELEVATOR.TOLERANCE
ActuatorConfiguration.RING_LIGHT
ActuatorConfiguration.SOLENOID_HOPPER
ActuatorConfiguration.SOLENOID_PICKUP
ActuatorConfiguration.SOLENOID_SHIFTER
ActuatorConfiguration.SOLENOID_SHOOTER
ActuatorConfiguration.TURRET_POSITION
Aiming
AutonomousConfiguration
Balancing
BallFireInfo
ButtonConfiguration
ButtonConfiguration.Driver
ButtonConfiguration.Manipulator
ButtonConfiguration.Manipulator.Elevator
CameraInterface
CompensatingGyro
ConvertingPIDController
DeadbandedSource
DualVictor
DummyRotationProvider
ElevatorMachine
ElevatorMachine.ElevatorState
EncoderOffset
EncoderPIDSource
Gyro360
GyroHax
LinearController
NaiveRotationProvider
Physics
PickupMachine
PickupMachine.PickupState
PIDDriveEncoderDifference
PIDDriveEncoderOutput
PIDDriveGyro
Point2d
Point3d
Point3d
PointAndAngle3d
PortConfiguration
PortConfiguration.Controllers
PortConfiguration.Encoders
PortConfiguration.Encoders.Drive
PortConfiguration.Motors
PortConfiguration.Pneumatics
PortConfiguration.Pneumatics.HOPPER_SOLENOID
PortConfiguration.Pneumatics.PICKUP_SOLENOID
PortConfiguration.Pneumatics.SHIFTER_SOLENOID
PortConfiguration.Pneumatics.SHOOTER_SOLENOID
PortConfiguration.Relays
PortConfiguration.Sensors
RemoteCameraTCP
Robot2012Orange
RotationProvider
SensorConfiguration
SensorConfiguration.Encoders
ShooterAnglePick
ShooterMachine
ShooterMachine.ShooterState
SlightlySmarterRotationProvider
SlowbroRotationProvider
SpringableDoubleSolenoid
SpringableRelay
SpringableVictor
StrangeMachine
Target
Target
TurretMachine
TurretMachine.TurretState
UpDownPIDController
UpDownPIDController.Gains
VelocityController
XboxController
XboxController.Axis
XboxController.Button
XboxController.Button.DPad
XboxController.Stick

edu.wpi.first.wpilibj

Class GyroHax

java.lang.Object
 edu.wpi.first.wpilibj.SensorBase
 edu.wpi.first.wpilibj.Gyro
 edu.wpi.first.wpilibj.GyroHax

All Implemented Interfaces:

IDevice, ISensor, PIDSource

Direct Known Subclasses:

CompensatingGyro

public class **GyroHax**
extends Gyro

Extender class for the Gyro class that exposes the underlying AnalogChannel.

Author:

Michael Smith

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Constructor Summary

Constructors

Constructor and Description

GyroHax(AnalogChannel channel)
Initializes a new GyroHax on the specified AnalogChannel.

GyroHax(int port)
Initializes a new GyroHax on the specified PWM port.

GyroHax(int slot, int port)
Initializes a new GyroHax on the specified PWM port on the specified module port.

Method Summary

Methods

Modifier and Type	Method and Description
AnalogChannel	getAnalogChannel () Gets the raw AnalogChannel.

Methods inherited from class edu.wpi.first.wpilibj.Gyro

free, getAngle, pidGet, reset, setSensitivity

Methods inherited from class edu.wpi.first.wpilibj.SensorBase

checkAnalogChannel, checkAnalogModule, checkDigitalChannel, checkDigitalModule, checkPWMChannel, checkPWMModule, checkRelayChannel, checkRelayModule, checkSolenoidChannel, checkSolenoidModule, getDefaultAnalogModule, getDefaultDigitalModule, getDefaultSolenoidModule, setDefaultAnalogModule, setDefaultDigitalModule, setDefaultSolenoidModule

Methods inherited from class java.lang.Object

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

Constructor Detail

GyroHax

```
public GyroHax(int port)
```

Initializes a new GyroHax on the specified PWM port. Note that port must be 1 or 2!

Parameters:

port - The PWM port the gyro is plugged into. Must be 1 or 2!

GyroHax

```
public GyroHax(int slot,
               int port)
```

Initializes a new GyroHax on the specified PWM port on the specified module port. Note that port must be 1 or 2!

Parameters:

slot - The module slot the gyro is plugged into.

port - The PWM port the gyro is plugged into. Must be 1 or 2!

GyroHax

```
public GyroHax(AnalogChannel channel)
```

Initializes a new GyroHax on the specified AnalogChannel. Note that port must be 1 or 2!

Parameters:

channel - The AnalogChannel the gyro is plugged into.

Method Detail

getAnalogChannel

```
public AnalogChannel getAnalogChannel()
```

Gets the raw AnalogChannel.

Returns:

The raw AnalogChannel.

Hierarchy For Package edu.wpi.first.wpilibj

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
 - edu.wpi.first.wpilibj.SensorBase
 - edu.wpi.first.wpilibj.Gyro (implements edu.wpi.first.wpilibj.parsing.ISensor, edu.wpi.first.wpilibj.PIDSource)
 - edu.wpi.first.wpilibj.GyroHax

Package edu.wpi.first.wpilibj

The WPI Robotics library (WPILibJ) is a set of Java classes that interfaces to the hardware in the FRC control system and your robot.

See: [Description](#)

Class Summary	
Class	Description
GyroHax	Extender class for the Gyro class that exposes the underlying AnalogChannel.

Package edu.wpi.first.wpilibj Description

The WPI Robotics library (WPILibJ) is a set of Java classes that interfaces to the hardware in the FRC control system and your robot. There are classes to handle sensors, motors, the driver station, and a number of other utility functions like timing and field management. The library is designed to:

- Deal with all the low level interfacing to these components so you can concentrate on solving this year's robot problem. This is a philosophical decision to let you focus on the higher-level design of your robot rather than deal with the details of the processor and the operating system.
- Understand everything at all levels by making the full source code of the library available. You can study (and modify) the algorithms used by the gyro class for oversampling and integration of the input signal or just ask the class for the current robot heading. You can work at any level.

Classes

GyroHax

Deprecated API

Contents

All Classes

ActuatorConfiguration
ActuatorConfiguration.ELEVATOR
ActuatorConfiguration.ELEVATOR.DEADBAND
ActuatorConfiguration.ELEVATOR.TOLERANCE
ActuatorConfiguration.RING_LIGHT
ActuatorConfiguration.SOLENOID_HOPPER
ActuatorConfiguration.SOLENOID_PICKUP
ActuatorConfiguration.SOLENOID_SHIFTER
ActuatorConfiguration.SOLENOID_SHOOTER
ActuatorConfiguration.TURRET_POSITION
Aiming
AutonomousConfiguration
Balancing
BallFireInfo
ButtonConfiguration
ButtonConfiguration.Driver
ButtonConfiguration.Manipulator
ButtonConfiguration.Manipulator.Elevator
CameraInterface
CompensatingGyro
ConvertingPIDController
DeadbandedSource
DualVictor
DummyRotationProvider
ElevatorMachine
ElevatorMachine.ElevatorState
EncoderOffset
EncoderPIDSource
Gyro360
GyroHax
LinearController
NaiveRotationProvider
Physics
PickupMachine
PickupMachine.PickupState
PIDDriveEncoderDifference
PIDDriveEncoderOutput
PIDDriveGyro
Point2d
Point3d
Point3d
PointAndAngle3d
PortConfiguration
PortConfiguration.Controllers
PortConfiguration.Encoders
PortConfiguration.Encoders.Drive
PortConfiguration.Motors
PortConfiguration.Pneumatics
PortConfiguration.Pneumatics.HOPPER_SOLENOID
PortConfiguration.Pneumatics.PICKUP_SOLENOID
PortConfiguration.Pneumatics.SHIFTER_SOLENOID
PortConfiguration.Pneumatics.SHOOTER_SOLENOID
PortConfiguration.Relays
PortConfiguration.Sensors
RemoteCameraTCP
Robot2012Orange
RotationProvider
SensorConfiguration
SensorConfiguration.Encoders
ShooterAnglePick
ShooterMachine
ShooterMachine.ShooterState
SlightlySmarterRotationProvider
SlowbroRotationProvider
SpringableDoubleSolenoid
SpringableRelay
SpringableVictor
StrangeMachine
Target
Target
TurretMachine
TurretMachine.TurretState
UpDownPIDController
UpDownPIDController.Gains
VelocityController
XboxController
XboxController.Axis
XboxController.Button
XboxController.Button.DPad
XboxController.Stick

Packages

Package	Description
com_604robotics.robot2012	
com_604robotics.robot2012.aiming	
com_604robotics.robot2012.autonomous	
com_604robotics.robot2012.balancing	
com_604robotics.robot2012.camera	
com_604robotics.robot2012.configuration	
com_604robotics.robot2012.machine	
com_604robotics.robot2012.physics	
com_604robotics.robot2012.rotation	
com_604robotics.robot2012.vision	
com_604robotics.utils	
edu.wpi.first.wpilibj	The WPI Robotics library (WPILibJ) is a set of Java classes that interf aces tothe hardware in the FRC control system and you r robot.
frc.vision	

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Overview

The [Overview](#) page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain six categories:

- Interfaces (*italic*)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description
- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary
- Field Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description
- Required Element Summary
- Optional Element Summary
- Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

All Classes

The [All Classes](#) link shows all classes and interfaces except non-static nested types.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

Constant Field Values

The [Constant Field Values](constant-values.html) page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

Overview	Package	Class	Tree	Deprecated	Index	Help
Prev	Next	Frames	No Frames	All Classes		

A B C D E F G H I L M N O P R S T U V W X Y Z

A**A** - Static variable in interface `com._604robotics.utils.XboxController.Button`**ACCELEROMETER** - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Sensors`**ACCELEROMETER_DRIVE_POWER** - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration`**ACCELEROMETER_SENSITIVITY** - Static variable in interface `com._604robotics.robot2012.configuration.SensorConfiguration`**ACCELEROMETER_UPPER_RADIANS** - Static variable in interface `com._604robotics.robot2012.configuration.SensorConfiguration`**ActuatorConfiguration** - Interface in `com._604robotics.robot2012.configuration`

Actuator polarity and power configuration.

ActuatorConfiguration.ELEVATOR - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.ELEVATOR.DEADBAND** - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.ELEVATOR.TOLERANCE** - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.RING_LIGHT** - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.SOLENOID_HOPPER** - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.SOLENOID_PICKUP** - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.SOLENOID_SHIFTER** - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.SOLENOID_SHOOTER** - Interface in `com._604robotics.robot2012.configuration`**ActuatorConfiguration.TURRET_POSITION** - Interface in `com._604robotics.robot2012.configuration`**AIM_AND_SHOOT** - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator`**aimAndShoot()** - Method in class `com._604robotics.robot2012.Robot2012Orange`

Aim at backboard, shoot.

AIMED - Static variable in interface `com._604robotics.robot2012.machine.TurretMachine.TurretState`**Aiming** - Class in `com._604robotics.robot2012.aiming`

Utility class for various aiming functions and such.

Aiming() - Constructor for class `com._604robotics.robot2012.aiming.Aiming`**angle** - Variable in class `com._604robotics.robot2012.physics.BallFireInfo`**angle** - Variable in class `com._604robotics.robot2012.vision.Target`

This is the angle of the target, relative to the camera.

angle_uncertainty - Variable in class `com._604robotics.robot2012.vision.Target`

This is the uncertainty of the angle of the target.

angleDeg - Variable in class `com._604robotics.robot2012.physics.ShooterAnglePick`**angleRad** - Variable in class `com._604robotics.robot2012.physics.ShooterAnglePick`**angleSlope** - Variable in class `com._604robotics.robot2012.physics.ShooterAnglePick`**AUTO_BALANCE** - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Driver`**autonomous()** - Method in class `com._604robotics.robot2012.Robot2012Orange`

Automated drive for autonomous mode.

AutonomousConfiguration - Interface in `com._604robotics.robot2012.configuration`

Autonomous mode configuration.

B**B** - Static variable in interface `com._604robotics.utils.XboxController.Button`**Back** - Static variable in interface `com._604robotics.utils.XboxController.Button`**BACKWARD_DISTANCE** - Static variable in interface `com._604robotics.robot2012.configuration.AutonomousConfiguration`**BACKWARD_DISTANCE_SIDES** - Static variable in interface `com._604robotics.robot2012.configuration.AutonomousConfiguration`**BACKWARD_DRIVE_POWER** - Static variable in interface `com._604robotics.robot2012.configuration.AutonomousConfiguration`**Balancing** - Class in `com._604robotics.robot2012.balancing`

Utility class for automated balancing assistance.

Balancing() - Constructor for class `com._604robotics.robot2012.balancing.Balancing`**BallFireInfo** - Class in `com._604robotics.robot2012.physics`

Class representing info for firing a ball.

BallFireInfo(ShooterAnglePick, double, double) - Constructor for class `com._604robotics.robot2012.physics.BallFireInfo`

Initializes a new BallFireInfo.

begin() - Method in interface `com._604robotics.robot2012.camera.CameraInterface`

Launches the CameraInterface.

begin() - Method in class `com._604robotics.robot2012.camera.RemoteCameraTCP`
Initializes communication.

betterVersionOfGetFiringVelocity(double, double, double) - Method in class `com._604robotics.robot2012.physics.Physics`
This function determines the firing velocities (and time) for a given distance (horizontally, and vertically) and a vertical velocity at which the ball should enter the hoop.

betterVersionOfGetFiringVelocity(double, double) - Method in class `com._604robotics.robot2012.physics.Physics`
This function guesses a good vertical velocity to enter the hoop, then determines the firing velocities (and time) for a given distance (horizontally, and vertically).

ButtonConfiguration - Interface in `com._604robotics.robot2012.configuration`
Button configuration.

ButtonConfiguration.Driver - Interface in `com._604robotics.robot2012.configuration`

ButtonConfiguration.Manipulator - Interface in `com._604robotics.robot2012.configuration`

ButtonConfiguration.Manipulator.Elevator - Interface in `com._604robotics.robot2012.configuration`

C

calculate() - Method in class `com._604robotics.utils.LinearController`
Function that performs the output calculation.

CameraInterface - Interface in `com._604robotics.robot2012.camera`
Represents a method for obtaining processed vision data from the camera.

`com._604robotics.robot2012` - package `com._604robotics.robot2012`

`com._604robotics.robot2012.aiming` - package `com._604robotics.robot2012.aiming`

`com._604robotics.robot2012.autonomous` - package `com._604robotics.robot2012.autonomous`

`com._604robotics.robot2012.balancing` - package `com._604robotics.robot2012.balancing`

`com._604robotics.robot2012.camera` - package `com._604robotics.robot2012.camera`

`com._604robotics.robot2012.configuration` - package `com._604robotics.robot2012.configuration`

`com._604robotics.robot2012.machine` - package `com._604robotics.robot2012.machine`

`com._604robotics.robot2012.physics` - package `com._604robotics.robot2012.physics`

`com._604robotics.robot2012.rotation` - package `com._604robotics.robot2012.rotation`

`com._604robotics.robot2012.vision` - package `com._604robotics.robot2012.vision`

`com._604robotics.utils` - package `com._604robotics.utils`

CompensatingGyro - Class in `com._604robotics.utils`
Gyro with manual compensation-setting support.

CompensatingGyro(int) - Constructor for class `com._604robotics.utils.CompensatingGyro`
Initializes a new CompensatingGyro on the specified PWM port.

CompensatingGyro(int, int) - Constructor for class `com._604robotics.utils.CompensatingGyro`
Initializes a new CompensatingGyro on the specified PWM port on the specified module port.

CompensatingGyro(AnalogChannel) - Constructor for class `com._604robotics.utils.CompensatingGyro`
Initializes a new CompensatingGyro on the specified AnalogChannel.

COMPRESSOR - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics`

ConvertingPIDController - Class in `com._604robotics.utils`
An extender of a PIDController that converts between units when getting and setting a setpoint.

ConvertingPIDController(double, double, double, PIDSource, PIDOutput) - Constructor for class `com._604robotics.utils.ConvertingPIDController`
Allocate a PID object with the given constants for P, I, D, using a 50ms period.

ConvertingPIDController(double, double, double, PIDSource, PIDOutput, double) - Constructor for class `com._604robotics.utils.ConvertingPIDController`
Allocate a PID object with the given constants for P, I, D

crank(int) - Method in class `com._604robotics.robot2012.machine.ElevatorMachine`

crank(int) - Method in class `com._604robotics.robot2012.machine.PickupMachine`

crank(int) - Method in class `com._604robotics.robot2012.machine.ShooterMachine`

crank(int) - Method in interface `com._604robotics.robot2012.machine.StrangeMachine`
Causes the Machine to strive for the target state.

crank(int) - Method in class `com._604robotics.robot2012.machine.TurretMachine`

D

D - Variable in class `com._604robotics.utils.UpDownPIDController.Gains`

deadband(double, double, double, double) - Static method in class `com._604robotics.robot2012.Robot2012Orange`
If a value is within a range, set it to a specific value.

DeadbandedSource - Class in `com._604robotics.utils`
Implements a PIDSource, wrapping around another PIDSource, with a deadband range.

DeadbandedSource(PIDSource) - Constructor for class `com._604robotics.utils.DeadbandedSource`
Initializes a new DeadbandedSource.

defaultAiming - Static variable in class `com._604robotics.robot2012.aiming.Aiming`

disable() - Method in class `com._604robotics.utils.VelocityController`
Disables the VelocityController.

disabled() - Method in class `com._604robotics.robot2012.Robot2012Orange`
The robot is disabled.

DOWN - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator.Elevator`

Down - Static variable in interface `com._604robotics.utils.XboxController.Button.DPad`

DPAD - Static variable in interface `com._604robotics.utils.XboxController.Stick`

DRIVE - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Controllers`

DualVictor - Class in `com._604robotics.utils`

Control two Victors like they're one.

DualVictor(int, int) - Constructor for class `com._604robotics.utils.DualVictor`

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

DualVictor(int, int, int, int) - Constructor for class `com._604robotics.utils.DualVictor`

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

DualVictor(Victor, Victor) - Constructor for class `com._604robotics.utils.DualVictor`

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

DummyRotationProvider - Class in `com._604robotics.robot2012.rotation`

Dummy implementor of a RotationProvider, for testing purposes.

DummyRotationProvider(PIDController) - Constructor for class `com._604robotics.robot2012.rotation.DummyRotationProvider`

Initializes a new DummyRotationProvider, giving it control over the specified PIDController.

E

`edu.wpi.first.wpilibj` - package `edu.wpi.first.wpilibj`

The WPI Robotics library (WPILibJ) is a set of Java classes that interfaces to the hardware in the FRC control system and your robot.

EitherTrigger - Static variable in interface `com._604robotics.utils.XboxController.Button`

ELEVATOR_A - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Encoders`

ELEVATOR_B - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Encoders`

ELEVATOR_LEFT - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Motors`

ELEVATOR_LIMIT_SWITCH - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Sensors`

ELEVATOR_POWER_MAX - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration`

ELEVATOR_POWER_MIN - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration`

ELEVATOR_RIGHT - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Motors`

ElevatorMachine - Class in `com._604robotics.robot2012.machine`

Machine to control the elevator.

ElevatorMachine(PIDController, Encoder) - Constructor for class `com._604robotics.robot2012.machine.ElevatorMachine`

Initializes a new ElevatorMachine.

ElevatorMachine.ElevatorState - Interface in `com._604robotics.robot2012.machine`

Various possible states the elevator can be in.

enable() - Method in class `com._604robotics.utils.VelocityController`

Enables the VelocityController.

EncoderOffset - Class in `com._604robotics.utils`

Encoder extender that return the value of Encoder.get() when pidGet is called.

EncoderOffset(int, int, int, int, boolean) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, int, int) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, int, int, boolean, CounterBase.EncodingType) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, int, int, int, int, boolean) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, int, int, int, int) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, boolean) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, boolean, CounterBase.EncodingType) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, int, boolean) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(int, int, int) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(DigitalSource, DigitalSource, boolean) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(DigitalSource, DigitalSource) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(DigitalSource, DigitalSource, boolean, CounterBase.EncodingType) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(DigitalSource, DigitalSource, DigitalSource, boolean) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderOffset(DigitalSource, DigitalSource, DigitalSource) - Constructor for class `com._604robotics.utils.EncoderOffset`

Encoder constructor.

EncoderPIDSource - Class in `com._604robotics.utils`

Encoder extender that return the value of Encoder.get() when pidGet is called.

EncoderPIDSource(int, int, int, int, boolean) - Constructor for class `com._604robotics.utils.EncoderPIDSource`

Encoder constructor.

EncoderPIDSource(int, int, int, int, int) - Constructor for class `com._604robotics.utils.EncoderPIDSource`

Encoder constructor.

EncoderPIDSource(int, int, int, int, boolean, CounterBase.EncodingType) - Constructor for class `com._604robotics.utils.EncoderPIDSource`

Encoder constructor.
EncoderPIDSource(int, int, int, int, int, int, boolean) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(int, int, int, int, int, int) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(int, int, boolean) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(int, int) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(int, int, boolean, CounterBase.EncodingType) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(int, int, int, boolean) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(int, int, int) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(DigitalSource, DigitalSource, boolean) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(DigitalSource, DigitalSource) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(DigitalSource, DigitalSource, boolean, CounterBase.EncodingType) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(DigitalSource, DigitalSource, DigitalSource, boolean) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
EncoderPIDSource(DigitalSource, DigitalSource, DigitalSource) - Constructor for class com._604robotics.utils.EncoderPIDSource
 Encoder constructor.
end() - Method in interface com._604robotics.robot2012.camera.CameraInterface
 Disables the CameraInterface.
end() - Method in class com._604robotics.robot2012.camera.RemoteCameraTCP
 Ends communication.

F

FORWARD - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.TURRET_POSITION
FORWARD - Static variable in interface com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator.Elevator
FORWARD - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.HOPPER_SOLENOID
FORWARD - Static variable in interface com._604robotics.robot2012.machine.TurretMachine.TurretState
FORWARD_DISTANCE - Static variable in interface com._604robotics.robot2012.configuration.AutonomousConfiguration
FORWARD_DRIVE_POWER - Static variable in interface com._604robotics.robot2012.configuration.AutonomousConfiguration
 frc.vision - package frc.vision

G

get() - Method in class com._604robotics.utils.DualVictor
 Checks the current power the Vectors are set to.
getActualVelocity() - Method in class com._604robotics.utils.VelocityController
 Gets the actual, current velocity.
getAnalogChannel() - Method in class edu.wpi.first.wpilibj.GyroHax
 Gets the raw AnalogChannel.
getAngle() - Method in class com._604robotics.utils.Gyro360
 Gets the angle of the gyro, constrained to 360 degrees.
getAngleAndRelXYZOfTarget(double, double, double, double, double, double, double, double) - Method in class com._604robotics.robot2012.aiming.Aiming
 Get the angle from the targets, and the relative distances of the corners of the target as perceived by the camera.
getAngleOfTarget(double, double, double, double, double, double, double, double, double, double) - Method in class com._604robotics.robot2012.aiming.Aiming
 This function gets the direction the target is facing, relative to the camera.
getAxis(int) - Method in class com._604robotics.utils.XboxController
 Get the value of the specified axis.
GetBallFiringInfo(double, double, double, double, double) - Method in class com._604robotics.robot2012.physics.Physics
 This function will determine how to fire the ball if the shooter only has 2 vertical angles.
getButton(int) - Method in class com._604robotics.utils.XboxController
 Get whether or not the specified button is currently pressed.
getDownGains() - Method in class com._604robotics.utils.UpDownPIDController
 Gets the Gains for going down.
getJoystick() - Method in class com._604robotics.utils.XboxController
 Gets the underlying Joystick object.
getRaw() - Method in class com._604robotics.utils.EncoderOffset
getRealSetpoint() - Method in class com._604robotics.utils.ConvertingPIDController
 Gets the "real" setpoint of the PIDController.
getRecordedTime() - Method in interface com._604robotics.robot2012.camera.CameraInterface
 Gets the estimated time since the last packet was received.
getRecordedTime() - Method in class com._604robotics.robot2012.camera.RemoteCameraTCP
 Records the time elapsed between reception of data packets from camera.
getRelXYZOfTarget(double, double, double, double) - Method in class com._604robotics.robot2012.aiming.Aiming
 Remember that this requires the camera to be "perfectly" flat, and the targets to be "perfectly" vertical.
getRelXYZOfTarget(Target) - Method in class com._604robotics.robot2012.aiming.Aiming
getSetpoint() - Method in class com._604robotics.utils.ConvertingPIDController
getSpeedforBalance(double) - Static method in class com._604robotics.robot2012.balancing.Balancing

Given a specific gyro reading, returns what speed you should be going at.

getSprung() - Method in class com._604robotics.utils.DualVictor
Has the victor been sprung?

getSprung() - Method in class com._604robotics.utils.SpringableDoubleSolenoid
Has the DoubleSolenoid been sprung?

getSprung() - Method in class com._604robotics.utils.SpringableRelay
Has the Relay been sprung?

getSprung() - Method in class com._604robotics.utils.SpringableVictor
Has the victor been sprung?

getStick(int) - Method in class com._604robotics.utils.XboxController
Get whether or not there's a value reading on the stick.

getSubparFiringVelocity(double, double, double) - Method in class com._604robotics.robot2012.physics.Physics
This untested function might determine the firing velocity for a given distance (horizontally, and vertically) and the angle of the shooter.

getTarget() - Method in class com._604robotics.utils.LinearController
Gets the current target.

getTargets() - Method in interface com._604robotics.robot2012.camera.CameraInterface
Returns the most recently-obtained array of Target that represents the visible targets.

getTargets() - Method in class com._604robotics.robot2012.camera.RemoteCameraTCP
Returns the last targets acquired from the remote software.

getToggle(int) - Method in class com._604robotics.utils.XboxController
Get the toggle state of the specified button.

getUpGains() - Method in class com._604robotics.utils.UpDownPIDController
Gets the Gains for going up.

getUPS() - Method in class com._604robotics.robot2012.camera.RemoteCameraTCP
Returns the number of updates received per second.

getVelocity() - Method in class com._604robotics.utils.VelocityController
Gets the current target velocity.

getX() - Method in class com._604robotics.robot2012.vision.Point3d

getY() - Method in class com._604robotics.robot2012.vision.Point3d

getZ() - Method in class com._604robotics.robot2012.vision.Point3d

Gyro360 - Class in com._604robotics.utils
Extender class to constrain the output of a Gyro to 360 degrees, looping.

Gyro360(int) - Constructor for class com._604robotics.utils.Gyro360
Initializes a new Gyro360 on the specified PWM port.

Gyro360(int, int) - Constructor for class com._604robotics.utils.Gyro360
Initializes a new Gyro360 on the specified PWM port on the specified module port.

Gyro360(AnalogChannel) - Constructor for class com._604robotics.utils.Gyro360
Initializes a new Gyro360 on the specified AnalogChannel.

GYRO_BALANCE - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Sensors

GYRO_DRIFT - Static variable in interface com._604robotics.robot2012.configuration.SensorConfiguration

GYRO_HEADING - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Sensors

GYRO_RESET - Static variable in interface com._604robotics.robot2012.configuration.ButtonConfiguration.Driver

GyroHax - Class in edu.wpi.first.wpilibj
Extender class for the Gyro class that exposes the underlying AnalogChannel.

GyroHax(int) - Constructor for class edu.wpi.first.wpilibj.GyroHax
Initializes a new GyroHax on the specified PWM port.

GyroHax(int, int) - Constructor for class edu.wpi.first.wpilibj.GyroHax
Initializes a new GyroHax on the specified PWM port on the specified module port.

GyroHax(AnalogChannel) - Constructor for class edu.wpi.first.wpilibj.GyroHax
Initializes a new GyroHax on the specified AnalogChannel.

H

h - Variable in class frc.vision.Target

HIGH - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.DEADBAND

HIGH - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR

HIGH - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.TOLERANCE

HIGH - Static variable in interface com._604robotics.robot2012.machine.ElevatorMachine.ElevatorState

HIGH_GEAR - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_SHIFTER

HIGH_GEAR - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHIFTER_SOLENOID

HOPPER - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Motors

HOPPER_POWER - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration

HOPPER_POWER_REVERSE - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration

horizontalAngle - Variable in class com._604robotics.robot2012.physics.BallFireInfo

I

I - Variable in class com._604robotics.utils.UpDownPIDController.Gains

IN - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_PICKUP

IN - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.PICKUP_SOLENOID

IN - Static variable in interface com._604robotics.robot2012.machine.PickupMachine.PickupState

isEnabled() - Method in class com._604robotics.utils.VelocityController

Is the VelocityController currently enabled?

isInRange(double, double, double) - Static method in class com._604robotics.robot2012.Robot2012Orange

Figures out if a value is within a specific range.

L

LB - Static variable in interface com._604robotics.utils.XboxController.Button

LEFT - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.TURRET_POSITION

LEFT - Static variable in interface com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator.Elevator

LEFT - Static variable in interface com._604robotics.robot2012.machine.TurretMachine.TurretState

Left - Static variable in interface com._604robotics.utils.XboxController.Button.DPad

LEFT_A - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Encoders.Drive

LEFT_B - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Encoders.Drive

LEFT_DRIVE - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Motors

LEFT_DRIVE_INCHES_PER_CLICK - Static variable in interface com._604robotics.robot2012.configuration.SensorConfiguration.Encoders

LEFT_STICK - Static variable in interface com._604robotics.utils.XboxController.Stick

LEFT_STICK_X - Static variable in interface com._604robotics.utils.XboxController.Axis

LEFT_STICK_Y - Static variable in interface com._604robotics.utils.XboxController.Axis

LeftStick - Static variable in interface com._604robotics.utils.XboxController.Button

LinearController - Class in com._604robotics.utils

This class implements a controller with a horizontal segment, a linear segment, and finally a coasting segment.

LinearController(PIDSource, PIDOutput, double, double, double, double) - Constructor for class com._604robotics.utils.LinearController

Initializes a new LinearController.

LOW - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.DEADBAND

LOW - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR

LOW - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.TOLERANCE

LOW - Static variable in interface com._604robotics.robot2012.machine.ElevatorMachine.ElevatorState

LOW_GEAR - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_SHIFTER

LOW_GEAR - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHIFTER_SOLENOID

LOWER_ANGLE - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_SHOOTER

LOWER_ANGLE - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHOOTER_SOLENOID

LT - Static variable in interface com._604robotics.utils.XboxController.Button

M

MANIPULATOR - Static variable in interface com._604robotics.robot2012.configuration.PortConfiguration.Controllers

MEDIUM - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR

MEDIUM - Static variable in interface com._604robotics.robot2012.machine.ElevatorMachine.ElevatorState

MEDIUM_LOWER - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.DEADBAND

MEDIUM_LOWER - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.TOLERANCE

MEDIUM_UPPER - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.DEADBAND

MEDIUM_UPPER - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.TOLERANCE

N

NaiveRotationProvider - Class in com._604robotics.robot2012.rotation

A naive implementation of a RotationProvider,

NaiveRotationProvider(PIDController, CameraInterface, Encoder) - Constructor for class com._604robotics.robot2012.rotation.NaiveRotationProvider

Initializes a new NaiveRotationProvider, giving it control over the specified PIDController.

O

OFF - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.RING_LIGHT

OKAY_TO_TURN - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR

ON - Static variable in interface com._604robotics.robot2012.configuration.ActuatorConfiguration.RING_LIGHT

onTarget() - Method in class `com._604robotics.utils.LinearController`

Are we there yet?

operatorControl() - Method in class `com._604robotics.robot2012.Robot2012Orange`

Operator-controlled drive for Teleop mode.

OUT - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_PICKUP`

OUT - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.PICKUP_SOLENOID`

OUT - Static variable in interface `com._604robotics.robot2012.machine.PickupMachine.PickupState`

P

P - Variable in class `com._604robotics.utils.UpDownPIDController.Gains`

Physics - Class in `com._604robotics.robot2012.physics`

Used for determining launch velocities of the ball.

Physics() - Constructor for class `com._604robotics.robot2012.physics.Physics`

PICKUP - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator`

PICKUP - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Motors`

PICKUP_OKAY - Static variable in interface `com._604robotics.robot2012.machine.ElevatorMachine.ElevatorState`

PICKUP_POWER - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration`

PickupMachine - Class in `com._604robotics.robot2012.machine`

Machine to control the pneumatic pickup.

PickupMachine(DoubleSolenoid) - Constructor for class `com._604robotics.robot2012.machine.PickupMachine`

Initializes a new PickupMachine.

PickupMachine.PickupState - Interface in `com._604robotics.robot2012.machine`

Possible states the pickup could be in.

PIDDriveEncoderDifference - Class in `com._604robotics.robot2012.autonomous`

This class implements a PIDSource, based on the difference of values between two encoders.

PIDDriveEncoderDifference(Encoder, Encoder) - Constructor for class `com._604robotics.robot2012.autonomous.PIDDriveEncoderDifference`

Initializes a new PIDDriveEncoderDifference, based on the given encoders.

PIDDriveEncoderOutput - Class in `com._604robotics.robot2012.autonomous`

This class implements the default PIDOutput class provided in the WPILib API.

PIDDriveEncoderOutput(RobotDrive, boolean) - Constructor for class `com._604robotics.robot2012.autonomous.PIDDriveEncoderOutput`

Initializes a new PIDDriveEncoderOutput.

PIDDriveEncoderOutput(RobotDrive) - Constructor for class `com._604robotics.robot2012.autonomous.PIDDriveEncoderOutput`

Initializes a new PIDDriveEncoderOutput.

PIDDriveGyro - Class in `com._604robotics.robot2012.autonomous`

Driving shim for the gyro-based PID-turning controller thing.

PIDDriveGyro(RobotDrive) - Constructor for class `com._604robotics.robot2012.autonomous.PIDDriveGyro`

Initializes a new PIDDriveGyro, based on the given RobotDrive.

pidGet() - Method in class `com._604robotics.robot2012.autonomous.PIDDriveEncoderDifference`

Gets the difference between the two encoder values, as an output to a PID controller.

pidGet() - Method in class `com._604robotics.utils.DeadbandedSource`

Hooks into PIDSource - gets the value to send to the PIDController.

pidGet() - Method in class `com._604robotics.utils.EncoderPIDSource`

Hooks into the PIDSource interface.

pidGet() - Method in class `com._604robotics.utils.Gyro360`

Implements the pidGet() function in the type PIDSource, allowing this class to be used as such.

pidWrite(double) - Method in class `com._604robotics.robot2012.autonomous.PIDDriveEncoderOutput`

Robot will drive with the configured power, and swerve determined by the encoder readings.

pidWrite(double) - Method in class `com._604robotics.robot2012.autonomous.PIDDriveGyro`

Writes the output from the PIDController to the RobotDrive, in the form of a turn value.

pidWrite(double) - Method in class `com._604robotics.utils.DualVictor`

Function to hook into the PIDController.

pidWrite(double) - Method in class `com._604robotics.utils.SpringableVictor`

Function to hook into the PIDController.

Point2d - Class in `com._604robotics.robot2012.aiming`

Represents a single point on the 2D plane.

Point2d(double, double) - Constructor for class `com._604robotics.robot2012.aiming.Point2d`

Initializes a new Point2d.

Point3d - Class in `com._604robotics.robot2012.aiming`

Represents a single point in 3D space.

Point3d() - Constructor for class `com._604robotics.robot2012.aiming.Point3d`

Initializes a new Point3d.

Point3d(double, double, double) - Constructor for class `com._604robotics.robot2012.aiming.Point3d`

Initializes a new Point3d.

Point3d - Class in `com._604robotics.robot2012.vision`

This represents a point in 3d space

Point3d(double, double, double) - Constructor for class `com._604robotics.robot2012.vision.Point3d`

PointAndAngle3d - Class in `com._604robotics.robot2012.aiming`

A class to hold a 3d point.

PointAndAngle3d(double, double, double, double) - Constructor for class `com._604robotics.robot2012.aiming.PointAndAngle3d`

Initializes variables for the point.

PointAndAngle3d(Point3d, double) - Constructor for class `com._604robotics.robot2012.aiming.PointAndAngle3d`

Initializes variables for the point.

PortConfiguration - Interface in `com._604robotics.robot2012.configuration`

Port configuration.

PortConfiguration.Controllers - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Controllers - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Encoders - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Encoders.Drive - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Motors - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Pneumatics - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Pneumatics.HOPPER_SOLENOID - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Pneumatics.PICKUP_SOLENOID - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Pneumatics.SHIFTER_SOLENOID - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Pneumatics.SHOOTER_SOLENOID - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Relays - Interface in `com._604robotics.robot2012.configuration`

PortConfiguration.Sensors - Interface in `com._604robotics.robot2012.configuration`

PRESSURE_SWITCH - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics`

PUSH - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_HOPPER`

R

RB - Static variable in interface `com._604robotics.utils.XboxController.Button`

refreshGains() - Method in class `com._604robotics.utils.UpDownPIDController`
Updates the gains for the current direction.

REGULAR - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_HOPPER`

reload() - Method in class `com._604robotics.utils.DualVictor`

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

reload() - Method in class `com._604robotics.utils.SpringableDoubleSolenoid`

If the DoubleSolenoid has been sprung, unsprung it; if not, set the output to the default output.

reload() - Method in class `com._604robotics.utils.SpringableRelay`

If the Relay has been sprung, unsprung it; if not, set the output to the default output.

reload() - Method in class `com._604robotics.utils.SpringableVictor`

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

RemoteCameraTCP - Class in `com._604robotics.robot2012.camera`

Implements a CameraInterface that draws data from a TCP connection.

RemoteCameraTCP() - Constructor for class `com._604robotics.robot2012.camera.RemoteCameraTCP`

reset() - Method in class `com._604robotics.utils.EncoderOffset`

Resets the Encoder.

resetToggles() - Method in class `com._604robotics.utils.XboxController`

Resets the toggle registry for the contrller.

REVERSE - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.HOPPER_SOLENOID`

RIGHT - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration.TURRET_POSITION`

RIGHT - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator.Elevator`

RIGHT - Static variable in interface `com._604robotics.robot2012.machine.TurretMachine.TurretState`

Right - Static variable in interface `com._604robotics.utils.XboxController.Button.DPad`

RIGHT_A - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Encoders.Drive`

RIGHT_B - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Encoders.Drive`

RIGHT_DRIVE - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Motors`

RIGHT_DRIVE_INCHES_PER_CLICK - Static variable in interface `com._604robotics.robot2012.configuration.SensorConfiguration.Encoders`

RIGHT_STICK - Static variable in interface `com._604robotics.utils.XboxController.Stick`

RIGHT_STICK_X - Static variable in interface `com._604robotics.utils.XboxController.Axis`

RIGHT_STICK_Y - Static variable in interface `com._604robotics.utils.XboxController.Axis`

RightStick - Static variable in interface `com._604robotics.utils.XboxController.Button`

RING_LIGHT_DIRECTION - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Relays`

RING_LIGHT_PORT - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Relays`

Robot2012Orange - Class in `com._604robotics.robot2012`

Main class for the 2012 robot code.

Robot2012Orange() - Constructor for class `com._604robotics.robot2012.Robot2012Orange`

Constructor.

robotinit() - Method in class `com._604robotics.robot2012.Robot2012Orange`

Initializes the robot on startup.

RotationProvider - Interface in `com._604robotics.robot2012.rotation`

Based on external feedback, aims the turret at the target.

RT - Static variable in interface `com._604robotics.utils.XboxController.Button`

S

SensorConfiguration - Interface in `com._604robotics.robot2012.configuration`

Sensor configuration.

SensorConfiguration.Encoders - Interface in `com._604robotics.robot2012.configuration`

set(double) - Method in class `com._604robotics.utils.DualVictor`

Sets the power of the Victors.

set(DoubleSolenoid.Value) - Method in class `com._604robotics.utils.SpringableDoubleSolenoid`

Sets the direction of the DoubleSolenoid.

set(Relay.Value) - Method in class `com._604robotics.utils.SpringableRelay`

Sets the direction of the Relay.

set(double) - Method in class `com._604robotics.utils.SpringableVictor`

Sets the power of the Victor.

setAccumulatorCenter(int) - Method in class `com._604robotics.utils.CompensatingGyro`

Manually sets the center for the accumulator.

setAngleGains(double, double, double) - Method in class `com._604robotics.utils.VelocityController`

Based on gyro angles TODO - javadoc

setCoastingRange(double, double) - Method in class `com._604robotics.utils.LinearController`

Updates the coasting values.

setController(PIDController) - Method in class `com._604robotics.utils.DeadbandedSource`

Sets the PIDController the source is fed into.

setController(PIDController) - Method in class `com._604robotics.utils.DualVictor`

Sets the PIDController for this DualVictor, if there is one.

setController(PIDController) - Method in class `com._604robotics.utils.SpringableVictor`

Sets the PIDController for this Victor, if there is one.

setConversionFactor(double) - Method in class `com._604robotics.utils.ConvertingPIDController`

Sets the factor to use when doing conversion on `setSetpoint` and `getSetpoint`.

setDeadband(double, double) - Method in class `com._604robotics.utils.DeadbandedSource`

Sets the range for the deadband.

setDeadband(double, double) - Method in class `com._604robotics.utils.DualVictor`

Sets the deadband for the DualVictor.

setDeadband(int, double, double) - Method in class `com._604robotics.utils.XboxController`

Sets the deadband for a particular axis.

setDefaultPosition(double) - Method in class `com._604robotics.robot2012.rotation.DummyRotationProvider`

setDefaultPosition(double) - Method in class `com._604robotics.robot2012.rotation.NaiveRotationProvider`

setDefaultPosition(double) - Method in interface `com._604robotics.robot2012.rotation.RotationProvider`

Sets the "default" position, if no targets can be located.

setDefaultPosition(double) - Method in class `com._604robotics.robot2012.rotation.SlightlySmarterRotationProvider`

setDefaultPosition(double) - Method in class `com._604robotics.robot2012.rotation.SlowbroRotationProvider`

setDownGains(UpDownPIDController.Gains) - Method in class `com._604robotics.utils.UpDownPIDController`

Sets the gains for going down.

setGains(double, double, double) - Method in class `com._604robotics.utils.VelocityController`

Reconfigures the gains on the PIDController.

setHorizontalRange(double, double) - Method in class `com._604robotics.utils.LinearController`

Updates the horizontal values.

setLeftInversion(boolean) - Method in class `com._604robotics.utils.DualVictor`

Sets the inversion for the "left" Victor.

setOffset(int) - Method in class `com._604robotics.utils.EncoderOffset`

Sets the offset value for the Encoder.

setRealSetpoint(double) - Method in class `com._604robotics.utils.ConvertingPIDController`

Sets the "real" setpoint of the PIDController.

setRightInversion(boolean) - Method in class `com._604robotics.utils.DualVictor`

Sets the inversion for the "right" Victor.

setSafetyEnabled(boolean) - Method in class `com._604robotics.utils.DualVictor`

Sets whether or not safety is enabled.

setSetpoint(double) - Method in class `com._604robotics.utils.ConvertingPIDController`

setSetpoint(double) - Method in class `com._604robotics.utils.UpDownPIDController`

Sets the setpoint to go to.

setShooterSpeed(double) - Method in class `com._604robotics.robot2012.machine.ShooterMachine`

Sets the shooter speed to use when, well, shooting.

setTarget(double) - Method in class `com._604robotics.utils.LinearController`

Sets the current target.

setTurretSidewaysPosition(double) - Method in class `com._604robotics.robot2012.machine.TurretMachine`

Sets the position to use as "SIDEWAYS".

setUpGains(UpDownPIDController.Gains) - Method in class `com._604robotics.utils.UpDownPIDController`

Sets the gains for going up.

setVelocity(double) - Method in class `com._604robotics.utils.VelocityController`

Sets the target velocity.

setX(double) - Method in class `com._604robotics.robot2012.vision.Point3d`

Sets the X value of this Point

setY(double) - Method in class `com._604robotics.robot2012.vision.Point3d`

Sets the Y value of this Point

setZ(double) - Method in class `com._604robotics.robot2012.vision.Point3d`

Sets the Z value of this Point

SHIFT - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Driver`

SHOOTER_LEFT - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Motors`

SHOOTER_RIGHT - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Motors`

ShooterAnglePick - Class in `com._604robotics.robot2012.physics`

Enum-ish thing of angles to shoot at.

ShooterAnglePick(double) - Constructor for class `com._604robotics.robot2012.physics.ShooterAnglePick`

Initializes a new ShooterAnglePick.

shooterAnglePickBottom - Static variable in class `com._604robotics.robot2012.physics.ShooterAnglePick`

shooterAnglePickTop - Static variable in class `com._604robotics.robot2012.physics.ShooterAnglePick`

ShooterMachine - Class in `com._604robotics.robot2012.machine`

Machine to control the shooter/hopper system during firing.

ShooterMachine(DualVictor, Victor) - Constructor for class `com._604robotics.robot2012.machine.ShooterMachine`

Initializes a new ShooterMachine.

ShooterMachine.ShooterState - Interface in `com._604robotics.robot2012.machine`

The possible states the shooter could be in.

SHOOTING - Static variable in interface `com._604robotics.robot2012.machine.ShooterMachine.ShooterState`

SIDEWAYS - Static variable in interface `com._604robotics.robot2012.machine.TurretMachine.TurretState`

SlightlySmarterRotationProvider - Class in `com._604robotics.robot2012.rotation`

A slightly smarter implementation of a rotation provider, which tries to account for network delay, etc.

SlightlySmarterRotationProvider(PIDController, CameraInterface, Encoder) - Constructor for class

`com._604robotics.robot2012.rotation.SlightlySmarterRotationProvider`

Initializes a new SlightlySmarterRotationProvider.

SlowbroRotationProvider - Class in `com._604robotics.robot2012.rotation`

Implements a slow-er-ish, but more robust-ish, RotationProvider.

SlowbroRotationProvider(ConvertingPIDController, CameraInterface, Encoder) - Constructor for class

`com._604robotics.robot2012.rotation.SlowbroRotationProvider`

Initializes a new SlowbroRotationProvider.

speed - Variable in class `com._604robotics.robot2012.physics.BallFireInfo`

spring() - Method in class `com._604robotics.utils.DualVictor`

Springs the victor.

spring() - Method in class `com._604robotics.utils.SpringableDoubleSolenoid`

Springs the DoubleSolenoid.

spring() - Method in class `com._604robotics.utils.SpringableRelay`

Springs the Relay.

spring() - Method in class `com._604robotics.utils.SpringableVictor`

Springs the victor.

SpringableDoubleSolenoid - Class in `com._604robotics.utils`

Extender of a DoubleSolenoid providing an easier control flow.

SpringableDoubleSolenoid(int, int, DoubleSolenoid.Value) - Constructor for class `com._604robotics.utils.SpringableDoubleSolenoid`

Initializes a new SpringableDoubleSolenoid.

SpringableDoubleSolenoid(int, int, int, DoubleSolenoid.Value) - Constructor for class `com._604robotics.utils.SpringableDoubleSolenoid`

Initializes a new SpringableDoubleSolenoid.

SpringableRelay - Class in `com._604robotics.utils`

Extender of a Relay providing an easier control flow.

SpringableRelay(int, int, Relay.Direction, Relay.Value) - Constructor for class `com._604robotics.utils.SpringableRelay`

Initializes a new SpringableRelay.

SpringableRelay(int, Relay.Direction, Relay.Value) - Constructor for class `com._604robotics.utils.SpringableRelay`

Initializes a new SpringableRelay.

SpringableRelay(int, int, Relay.Value) - Constructor for class `com._604robotics.utils.SpringableRelay`

Initializes a new SpringableRelay.

SpringableRelay(int, Relay.Value) - Constructor for class `com._604robotics.utils.SpringableRelay`

Initializes a new SpringableRelay.

SpringableVictor - Class in `com._604robotics.utils`

Extender of a Victor providing an easier control flow.

SpringableVictor(int) - Constructor for class `com._604robotics.utils.SpringableVictor`

Initializes a new SpringableVictor on the given PWM port.

SpringableVictor(int, int) - Constructor for class `com._604robotics.utils.SpringableVictor`

Initializes a new SpringableVictor on the given module slot and PWM port.

Start - Static variable in interface `com._604robotics.utils.XboxController.Button`

StrangeMachine - Interface in `com._604robotics.robot2012.machine`

State manager for various components of the robot.

T

Target - Class in `com._604robotics.robot2012.vision`

Represents a target.

Target(double, double, double, double) - Constructor for class `com._604robotics.robot2012.vision.Target`

Target(double, double, double, double, double, double, double, double) - Constructor for class `com._604robotics.robot2012.vision.Target`

Target(Point3d, double) - Constructor for class `com._604robotics.robot2012.vision.Target`

Target() - Constructor for class `com._604robotics.robot2012.vision.Target`

Target - Class in `frc.vision`

An Object to hold target parameters.

Target() - Constructor for class `frc.vision.Target`

Blank constructor.

Target(int, int, int, int) - Constructor for class `frc.vision.Target`

test(int) - Method in class `com._604robotics.robot2012.machine.ElevatorMachine`

test(int) - Method in class `com._604robotics.robot2012.machine.PickupMachine`

test(int) - Method in class `com._604robotics.robot2012.machine.ShooterMachine`

test(int) - Method in interface `com._604robotics.robot2012.machine.StrangeMachine`
Tests if the Machine has yet attained the target state.

test(int) - Method in class `com._604robotics.robot2012.machine.TurretMachine`

TOGGLE_ANGLE - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator`

TOGGLE_HEIGHT - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator`

TOGGLE_LIGHT - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator`

TOGGLE_PICKUP - Static variable in interface `com._604robotics.robot2012.configuration.ButtonConfiguration.Driver`

TOLERANCE - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration.TURRET_POSITION`

toString() - Method in class `com._604robotics.robot2012.vision.Target`

TURRET_CALIBRATION_OFFSET - Static variable in interface `com._604robotics.robot2012.configuration.SensorConfiguration`

TURRET_DEGREES_PER_CLICK - Static variable in interface `com._604robotics.robot2012.configuration.SensorConfiguration.Encoders`

TURRET_OKAY - Static variable in interface `com._604robotics.robot2012.machine.ElevatorMachine.ElevatorState`

TURRET_ROTATION - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Motors`

TURRET_ROTATION_A - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Encoders`

TURRET_ROTATION_B - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Encoders`

TURRET_ROTATION_POWER_MAX - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration`

TURRET_ROTATION_POWER_MIN - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration`

TurretMachine - Class in `com._604robotics.robot2012.machine`
Machine to control the turret.

TurretMachine(PIDController, RotationProvider, Encoder) - Constructor for class `com._604robotics.robot2012.machine.TurretMachine`
Initializes a new TurretMachine.

TurretMachine.TurretState - Interface in `com._604robotics.robot2012.machine`
The possible states the turret could be in.

U

Up - Static variable in interface `com._604robotics.utils.XboxController.Button.DPad`

update() - Method in class `com._604robotics.robot2012.rotation.DummyRotationProvider`

update() - Method in class `com._604robotics.robot2012.rotation.NaiveRotationProvider`

update() - Method in interface `com._604robotics.robot2012.rotation.RotationProvider`
Updates the aiming of the turret.

update() - Method in class `com._604robotics.robot2012.rotation.SlightlySmarterRotationProvider`

update() - Method in class `com._604robotics.robot2012.rotation.SlowbroRotationProvider`

update() - Method in class `com._604robotics.utils.LinearController`
Updates the PIDOutput based on the latest data.

UpDownPIDController - Class in `com._604robotics.utils`
A PIDController with different gains for up and down.

UpDownPIDController(UpDownPIDController.Gains, UpDownPIDController.Gains, PIDSource, PIDOutput) - Constructor for class `com._604robotics.utils.UpDownPIDController`
Initializes a new UpDownPIDController.

UpDownPIDController.Gains - Class in `com._604robotics.utils`
A structure containing the P, I, and D gains.

UpDownPIDController.Gains(double, double, double) - Constructor for class `com._604robotics.utils.UpDownPIDController.Gains`

UPPER_ANGLE - Static variable in interface `com._604robotics.robot2012.configuration.ActuatorConfiguration.SOLENOID_SHOOTER`

UPPER_ANGLE - Static variable in interface `com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHOOTER_SOLENOID`

V

VelocityController - Class in `com._604robotics.utils`
Class for controlling a motor's velocity, rather than its power directly.

VelocityController(double, double, double, Encoder, Encoder, RobotDrive, Gyro) - Constructor for class `com._604robotics.utils.VelocityController`
Initializes a new VelocityController.

velToPow(double) - Static method in class `com._604robotics.robot2012.physics.Physics`
Returns an approximation of the power the shooter should be spun at

W

w - Variable in class `frc.vision.Target`

X

x - Variable in class `com._604robotics.robot2012.aiming.Point3d`

x - Variable in class `com._604robotics.robot2012.vision.Point3d`

x - Variable in class `com._604robotics.robot2012.vision.Point3d`
the x value

x - Variable in class `com._604robotics.robot2012.vision.Target`
x, y, and z represent the 3-d position of the target x will be positive when the target appears to be right of the center of the camera.

X - Static variable in interface `com._604robotics.utils.XboxController.Button`

x1 - Variable in class `frc.vision.Target`

x_uncertainty - Variable in class `com._604robotics.robot2012.vision.Target`
These are the uncertainties of the x, y, and z positions of the target.

XboxController - Class in `com._604robotics.utils`
Wrapper joystick class for the Xbox 360 controllers.

XboxController(int) - Constructor for class `com._604robotics.utils.XboxController`
Initialize a new XboxController on the specified port.

XboxController(Joystick) - Constructor for class `com._604robotics.utils.XboxController`
Initialize a new XboxController from the underlying Joystick.

XboxController.Axis - Interface in `com._604robotics.utils`
Enumeration for the available axes on the Xbox controller.

XboxController.Button - Interface in `com._604robotics.utils`
Enumeration for the available buttons on the Xbox controller.

XboxController.Button.DPad - Interface in `com._604robotics.utils`

XboxController.Stick - Interface in `com._604robotics.utils`
Enumeration for the available sticks on the Xbox controller.

Y

y - Variable in class `com._604robotics.robot2012.aiming.Point3d`

y - Variable in class `com._604robotics.robot2012.vision.Point3d`
the y value

y - Variable in class `com._604robotics.robot2012.vision.Target`
x, y, and z represent the 3-d position of the target x will be positive when the target appears to be right of the center of the camera.

Y - Static variable in interface `com._604robotics.utils.XboxController.Button`

y1 - Variable in class `frc.vision.Target`

y_uncertainty - Variable in class `com._604robotics.robot2012.vision.Target`
These are the uncertainties of the x, y, and z positions of the target.

Z

z - Variable in class `com._604robotics.robot2012.aiming.Point3d`

z - Variable in class `com._604robotics.robot2012.vision.Point3d`
the z value

z - Variable in class `com._604robotics.robot2012.vision.Target`
x, y, and z represent the 3-d position of the target x will be positive when the target appears to be right of the center of the camera.

z_uncertainty - Variable in class `com._604robotics.robot2012.vision.Target`
These are the uncertainties of the x, y, and z positions of the target.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Overview Package Class Tree Deprecated **Index** Help

Prev Next Frames No Frames All Classes

frc.vision

Class Target

java.lang.Object
frc.vision.Target

```
public class Target
extends Object
```

An Object to hold target parameters.

Author:
Kevin Parker , Sebastian Merz

Field Summary

Fields

Modifier and Type	Field and Description
int	h
int	w
int	x1
int	y1

Constructor Summary

Constructors

Constructor and Description
Target () Blank constructor.
Target (int x1, int y1, int w, int h)

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail

x1

public int x1

y1

public int y1

w

public int w

h

```
public int h
```

Constructor Detail

Target

```
public Target()
```

Blank constructor. Does nothing.

Target

```
public Target(int x1,  
              int y1,  
              int w,  
              int h)
```

Parameters:

x1 - The left x value for the target.

y1 - The bottom y value for the target.

w - The width of the target.

h - The height of the target.

[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](#)

Hierarchy For Package frc.vision

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
 - frc.vision.Target

Package frc.vision

Class Summary

Class	Description
Target	An Object to hold target parameters.

Classes

Target

com._604robotics.robot2012.autonomous

Class PIDDriveEncoderDifference

java.lang.Object
com._604robotics.robot2012.autonomous.PIDDriveEncoderDifference

All Implemented Interfaces:

PIDSource

```
public class PIDDriveEncoderDifference
extends Object
implements PIDSource
```

This class implements a PIDSource, based on the difference of values between two encoders.

Author:

Aaron Wang

Constructor Summary

Constructors

Constructor and Description

PIDDriveEncoderDifference(Encoder leftEncoder, Encoder rightEncoder)
Initializes a new PIDDriveEncoderDifference, based on the given encoders.

Method Summary

Methods

Modifier and Type	Method and Description
double	pidGet () Gets the difference between the two encoder values, as an output to a PID controller.

Methods inherited from class java.lang.Object

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

Constructor Detail

PIDDriveEncoderDifference

```
public PIDDriveEncoderDifference(Encoder leftEncoder,
                                Encoder rightEncoder)
```

Initializes a new PIDDriveEncoderDifference, based on the given encoders.

Parameters:

- leftEncoder - The left encoder to monitor the value of.
- rightEncoder - The right encoder to monitor the value of.

Method Detail

pidGet

```
public double pidGet()
```

Gets the difference between the two encoder values, as an output to a PID controller.

Specified by:

Example 1:

`pidGet` in interface `PIDSource`

Returns:

The difference between the two encoder values.

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

Hierarchy For Package com._604robotics.robot2012.autonomous

Package Hierarchies:

All Packages

Class Hierarchy

- java.lang.**Object**
 - com._604robotics.robot2012.autonomous.**PIDDriveEncoderDifference** (implements edu.wpi.first.wpilibj.PIDSource)
 - com._604robotics.robot2012.autonomous.**PIDDriveEncoderOutput** (implements edu.wpi.first.wpilibj.PIDOutput)
 - com._604robotics.robot2012.autonomous.**PIDDriveGyro** (implements edu.wpi.first.wpilibj.PIDOutput)

Package com._604robotics.robot2012.autonomous

Class Summary

Class	Description
PIDDriveEncoderDifference	This class implements a PIDSource, based on the difference of values between two encoders.
PIDDriveEncoderOutput	This class implements the default PIDOutput class provided in the WPILib API.
PIDDriveGyro	Driving shim for the gyro-based PID-turning controller thing.

Classes

PIDDriveEncoderDifference
PIDDriveEncoderOutput
PIDDriveGyro

com. _ 6 0 4 rob otics. rob utt2 ondu s . a

Class PIDDriveGyro

java.lang.Object
com. _ 6 0 4 rob otics. rob utt2 ondu s PIDDriveUyro

All Implemented Interfaces:

PIDOutput

```
public class PIDDriveGyro
extends Object
implements PIDOutput
```

Driving sh im Ror th yro based PIDTurning controller th ing.

Author:

M ich a Smith

Constructor Summary

Constructors
Constructor and Description
PIDDriveGyro (RobotDrive driveTrain) Initiali, es a new PIDDriveUyro(b ased on th e vgin z ob otDei.

Method Summary

Methods	
Modifier and Type	Method and Description
void	<code>pidWrite</code> (double output) Writes the output from the PIDController to the zobelDriftForm output value.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

PIDDriveGyro
<pre>public PIDDriveGyro(RobotDrive driveTrain)</pre> <p>Initiali, es a new PIDDriveUyro(b ased on th e vgin z ob otDei.</p> <p>Parameters:</p> <p>driveTrain B Th e z ob vDut j ect to control.</p>

Method Detail

pidWrite
<pre>public void pidWrite(double output)</pre> <p>W rites th atput Rrom th e PIDController to th e z obve(Drith e Rorm oRuntvalue.</p> <p>Specified by:</p> <p>pidWrite in interRacePIDOutput</p>

Parameters:

output B Th output oR th e PIDController.

Overview Package **Class** Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested bField bConstr bM eth od Detail: Field bConstr bM eth od

com._604robotics.robot2012.autonomous

Class PIDDriveEncoderOutput

java.lang.Object
com._604robotics.robot2012.autonomous.PIDDriveEncoderOutput

All Implemented Interfaces:

PIDOutput

```
public class PIDDriveEncoderOutput
extends Object
implements PIDOutput
```

This class implements the default PIDOutput class provided in the WPILib API. The class determines motor power to the robot drive so that the robot will drive backwards, depending on the encoder values.

Author:

Aaron Wang , Michael Smith

Constructor Summary

Constructors

Constructor and Description
<code>PIDDriveEncoderOutput(RobotDrive driveTrain)</code> Initializes a new PIDDriveEncoderOutput.
<code>PIDDriveEncoderOutput(RobotDrive driveTrain, boolean inversion)</code> Initializes a new PIDDriveEncoderOutput.

Method Summary

Methods

Modifier and Type	Method and Description
void	<code>pidWrite(double output)</code> Robot will drive with the configured power, and swerve determined by the encoder readings.

Methods inherited from class java.lang.Object

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

Constructor Detail

PIDDriveEncoderOutput

```
public PIDDriveEncoderOutput(RobotDrive driveTrain,
                             boolean inversion)
```

Initializes a new PIDDriveEncoderOutput.

Parameters:

- `driveTrain` - The RobotDrive object to control.
- `inversion` - Should the output be inverted?

PIDDriveEncoderOutput

```
public PIDDriveEncoderOutput(RobotDrive driveTrain)
```

Initializes a new PIDDriveEncoderOutput.

Parameters:

- `driveTrain` - The RobotDrive object to control.

Method Detail

pidWrite

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

Robot will drive with the configured power, and swerve determined by the encoder readings.

Specified by:

`pidWrite` in interface `PIDOutput`

Parameters:

`output` - The output of the PID controller.

[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](#)

Hierarchy For Package com._604robotics.robot2012.nphysics

Package Hierarchies
All Packages

Package Hierarchy

- java.lang.**Object**
 - com._604robotics.robot2012.p6 0 sites.001R **Object**
 - com._604robotics.robot2012.p6 0 sites.001R **Physics**
 - com._604robotics.robot2012.p6 0 sites.001R **Shooter** **GoPick**

Package com._604robotics.robot2012.physics

Class Summary

Class	Description
BallFireInfo	Class representing info for firing a ball.
Physics	Used for determining launch velocities of the ball.
ShooterAnglePick	Enum-ish thing of angles to shoot at.

Classes

BallFireInfo
Physics
ShooterAnglePick

com._604robotics.robot2012.physics

Class Physics

java.lang.Object
com._604robotics.robot2012.physics.Physics

```
public class Physics
extends Object
```

Used for determining launch velocities of the ball. It gives velocity as a function of displacement and final vertical velocity

Author:

Kevin Parker

Constructor Summary

Constructors

Constructor and Description
Physics()

Method Summary

Methods

Modifier and Type	Method and Description
Point2d	betterVersionOfgetFiringVelocity (double distH, double distV) This function guesses a good vertical velocity to enter the hoop, then determines the firing velocities (and the) f or a given distance (horiz onta), and vertical).
Point2d	betterVersionOfgetFiringVelocity (double distH, double distV, double verticalVel) This function determines the firing velocities (and the) f or a given distance (horiz onta), and vertical) and a vertical velocity at which the ball should enter the hoop.
BallFireInfo	GetBallFiringInfo (double xDist, double yDist, double zDist, double robotVelX, double robotVelZ) This function will determine how to fire the ball if the shooter only has 2 vertical angles.
double	getSubparFiringVelocity (double distH, double distV, double slope) This untested function might determine the firing velocity for a given distance (horiz onta), and vertical) and the angle of the shooter.
static double	velToPow (double vel) Returns an approximation of the power the shooter should be spun at

Methods inherited from class java.lang.Object

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

Constructor Detail

Physics

```
public Physics()
```

Method Detail

velToPow

```
public static double velToPow(double vel)
```

Returns an approximation of the power the shooter should be spun at

Parameters:

Parameters:

vel - - velocity in inches/ second

Returns:

the power to spin the shooter wheel at

getSubparFiringVelocity

```
public double getSubparFiringVelocity(double distH,
                                     double distV,
                                     double slope)
```

This untested function might determine the firing velocity for a given distance (horizontal, and vertical) and the angle of the shooter.

Parameters:

- distH- Horizontal distance the ball must travel.
- distV- Vertical distance the ball must travel.
- slope- What slope the launcher is at.

Returns:

The firing velocity

betterVersionOfgetFiringVelocity

```
public Point2d betterVersionOfgetFiringVelocity(double distH,
                                                double distV,
                                                double verticalVel)
```

This function determines the firing velocities (and the) for a given distance (horizontal, and vertical) and a vertical velocity at which the ball should enter the hoop.

Parameters:

- distH- Horizontal distance the ball must travel.
- distV- Vertical distance the ball must travel.
- verticalVel - Velocity at which the ball should enter the hoop.

Returns:

A Point2d with the x and y velocities does not return the time.

betterVersionOfgetFiringVelocity

```
public Point2d betterVersionOfgetFiringVelocity(double distH,
                                                double distV)
```

This function guesses a good vertical velocity to enter the hoop, then determines the firing velocities (and the) for a given distance (horizontal, and vertically).

Parameters:

- distH- Horizontal distance the ball must travel.
- distV- Vertical distance the ball must travel.

Returns:

A Point2d with the x and y velocities does not return the time.

GetBallFiringInfo

```
public BallFireInfo GetBallFiringInfo(double xDist,
                                     double yDist,
                                     double zDist,
                                     double robotVelX,
                                     double robotVelZ)
```

This function will determine how to fire the ball if the shooter only has 2 vertical angles.

Parameters:

- xDist - Left distance of the target.
- yDist - Vertical distance of the target.
- zDist - Depth distance of the target.
- robotVelX - Current velocity (x axis) of the robot.
- robotVelZ - Current velocity (z axis) of the robot

Returns:

A B allFireInfo with the velocity angle, azimuthalAngle to fire the ball at K eventually

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

com. _ 6 0 4 rob otics. rob y t c s 1 2 . ph

Class ShooterAnglePick

java.lang.Object
com. _ 6 0 4 rob otics. rob y t c s ShooterAnglePick

public class **ShooterAnglePick**
extends Object

EnumRish th ing oB angles to sh oot at.

Author:

, e i n Parker

Field Summary

Fields	
Modifier and Type	Field and Description
double	angleDeg
double	angleRad
double	angleSlope
static ShooterAnglePick	shooterAnglePickBottom
static ShooterAnglePick	shooterAnglePickTop

Constructor Summary

Constructors	
Constructor and Description	
ShooterAnglePick (double angleDeg)	
InitialiK es a newShooterAnglePick.	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

shooterAnglePickTop
public static final ShooterAnglePick shooterAnglePickTop
shooterAnglePickBottom
public static final ShooterAnglePick shooterAnglePickBottom
angleDeg
public final double angleDeg
angleRad

```
public final double angleRad
```

angleSlope

```
public final double angleSlope
```

Constructor Detail

ShooterAnglePick

```
public ShooterAnglePick(double angleDeg)
```

InitialiK es a newShooterAnglePick.

Parameters:

angleDeg R An anglez in degrees.

Overview Package **Class** Tree Deprecated Index Help

Prev Class ext Class Frames o rññes All Classes

Summary: ested bField bConstr bM eth od Detail: Field bConstr b M eth od

com._604robotics.robot2012.physics

Class **BallFireInfo**

java.lang.Object
com._604robotics.robot2012.physics.f aFireInfo

public class **BallFireInfo**
extends Object

Class representing info for firing a ball.

Author:

B   in Parker

Field Summary

Fields	
Modifier and Type	Field and Description
ShooterAnglePick	angle
double	horizontalAngle
double	speed

Constructor Summary

Constructors	
Constructor and Description	
BallFireInfo (ShooterAnglePick angle, double speed, double horizontalAngle) Initiali, es a new f ��ireInfo.	

Method Summary

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Field Detail

angle
public ShooterAnglePick angle
speed
public double speed
horizontalAngle
public double horizontalAngle

Constructor Detail

BallFireInfo

```
public BallFireInfo(ShooterAnglePick angle,
    double speed,
    double horizontalAngle)
```

Initializes a new instance of BallFireInfo.

Parameters:

- angle K An angle.
- speed K A speed.
- horizontalAngle K A horizontal angle.

com. _ 6 0 4 rob otics. rob ot2 0 1 2 . b alancing

Class Balancing

j [a.lang.Object](#)
com. _ 6 0 4 rob otics. rob ot2 0 1 2 . b alancing. f alancing

public class **Balancing**
extends [Object](#)

R [tily](#) class B or [atomated](#) b alancing assistance.

Author:

, [ein](#) Parker

Constructor Summary

Constructors
Constructor and Description
Balancing()

Method Summary

Methods

Modifier and Type	Method and Description
static double	getSpeedforBalance (double balGyroReading) K vien a speciBic gro readingz ratrns what speed you should be going at.

Methods inherited from class java.lang.Object

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

Constructor Detail

Balancing
<pre>public Balancing()</pre>

Method Detail

getSpeedforBalance
<pre>public static double getSpeedforBalance(double balGyroReading)</pre> <p>K vien a speciBic gro readingz ratrns what speed you should be going at.</p> <p>Parameters:</p> <p>balGyroReading - A gyro reading.</p> <p>Returns:</p> <p>The speed you should going at.</p>

Hierarchy For Package com._604robotics.robot2012.balancing

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
 - com._604robotics.robot2012.balancing.IBalancing

Package com._604robotics.robot2012.balancing

Class | 

Class	Description
RobotBalancing	Utility class for robot2012 balancing assistance

Classes

Balancing

com._604robotics.robot2012.machine

Interface StrangeMachine

All Known Implementing Classes:

[j](#) [LeatorMachine](#)[f](#) [PickupMachine](#)[f](#) [ShooterMachine](#)[f](#) [TurretMachine](#)

public interface **StrangeMachine**

State manager for various components of the robot. B sed for coordinating switches between states involving multiple steps and components.

Author:

Michael Smith

Method Summary

Methods	
Modifier and Type	Method and Description
boolean	crank (int state) Causes the Machine to strive for the target state.
boolean	test (int state) Tests if the Machine has yet attained the target state.

Method Detail

test
<pre>boolean test(int state)</pre> <p>Tests if the Machine has yet attained the target state.</p> <p>Parameters:</p> <p><code>state</code>, The target state.</p> <p>Returns:</p> <p><code>True</code> if the Machine has attained the target state. <code>False</code> otherwise.</p>
crank
<pre>boolean crank(int state)</pre> <p>Causes the Machine to strive for the target state.</p> <p>Parameters:</p> <p><code>state</code>, The state to strive for.</p> <p>Returns:</p> <p><code>True</code> if the target state has been reached. <code>False</code> otherwise.</p>

com._604robotics.robot2012.machine

Class ShooterMachine

java.lang.Object
com._604robotics.robot2012.machine.ShooterMachine

All Implemented Interfaces:

StrangeMachine

```
public class ShooterMachine
extends Object
implements StrangeMachine
```

Machine to control the shooterf bpper system during firing.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static interface	ShooterMachine.ShooterState The possible states the shooter could be in.

Constructor Summary

Constructors

Constructor and Description
ShooterMachine(DualVictor shooter, Victor hopper) InitialiB es a new ShooterMachine.

Method Summary

Methods

Modifier and Type	Method and Description
boolean	crank(int state) Causes the Machine to strive for the target state.
void	setShooterSpeed(double speed) Sets the shooter speed to use when, well, shooting.
boolean	test(int state) Tests if the Machine has yet attained the target state.

Methods inherited from class java.lang.Object

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

Constructor Detail

ShooterMachine

```
public ShooterMachine(DualVictor shooter,
                      Victor hopper)

InitialiB es a new ShooterMachine.
```

Parameters:

shooter KThe motors of the shooter to control.

hopper KThe motor of the hopper to control.

Method Detail

setShooterSpeed

public void setShooterSpeed(double speed)

Sets the shooter speed to use when, well, shooting.

Parameters:

speed KThe shooter speed to use when, well, shooting.

test

public boolean test(int state)

Description copied from interface: StrangeMachine

Tests if the Machine has yet attained the target state.

Specified by:

test in interface StrangeMachine

Parameters:

state KThe target state.

Returns:

z h etho not the Machine has attained the target state.

crank

public boolean crank(int state)

Description copied from interface: StrangeMachine

Causes the Machine to strive for the target state.

Specified by:

crank in interface StrangeMachine

Parameters:

state KThe state to strive for.

Returns:

z h etho not the target state has been reached.

com._604robotics.robot2012.machine

Interface ShooterMachine.ShooterState

Enclosing class:

ShooterMachine

```
public static interface ShooterMachine.ShooterState
```

The possible states the shooter could be in.

Field Summary

Fields

Modifier and Type	Field and Description
static int	SHOOTING

Field Detail

SHOOTING

```
static final int SHOOTING
```

See Also:

Constant Field j a les

com._604robotics.robot2012.machine

Interface ElevatorMachine.ElevatorState

Enclosing class:

j | ElevatorMachine

public static interface **ElevatorMachine.ElevatorState**

fields possible states the elevator can be in.

Field Summary

Fields

Modifier and Type	Field and Description
static int	HIGH
static int	LOW
static int	MEDIUM
static int	PICKUP_OKAY
static int	TURRET_OKAY

Field Detail

HIGH

```
static final int HIGH
```

See Also:

Constant Field values

MEDIUM

```
static final int MEDIUM
```

See Also:

Constant Field values

LOW

```
static final int LOW
```

See Also:

Constant Field values

PICKUP_OKAY

```
static final int PICKUP_OKAY
```

See Also:

Constant Field values

TURRET_OKAY

```
static final int TURRET_OKAY
```

See Also:

Constant Field values

com. _ 6 0 4 rob otics. rob ot2 0 1 2 . mach ine

Interface TurretMachine.TurretState

Enclosing class:

TurretMachine

```
public static interface TurretMachine.TurretState
```

The possible states the turret could be in.

Field Summary

Fields

Modifier and Type	Field and Description
static int	AIMED
static int	FORWARD
static int	LEFT
static int	RIGHT
static int	SIDEWAYS

Field Detail

SIDEWAYS

```
static final int SIDEWAYS
```

See Also:

Constant Field values

AIMED

```
static final int AIMED
```

See Also:

Constant Field values

FORWARD

```
static final int FORWARD
```

See Also:

Constant Field values

LEFT

```
static final int LEFT
```

See Also:

Constant Field values

RIGHT

```
static final int RIGHT
```

See Also:

Constant Field values

Hierarchy For Package com._604robotics.robot2012.machine

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
 - com._604robotics.robot2012.machine.ElevatorMachine (implements com._604robotics.robot2012.machine.TrangeMachine)
 - com._604robotics.robot2012.machine.PickupMachine (implements com._604robotics.robot2012.machine.TrangeMachine)
 - com._604robotics.robot2012.machine.SooterMachine (implements com._604robotics.robot2012.machine.TrangeMachine)
 - com._604robotics.robot2012.machine.UltrasonicMachine (implements com._604robotics.robot2012.machine.TrangeMachine)

Interface Hierarchy

- com._604robotics.robot2012.machine.ElevatorMachine.IElevatorPort
- com._604robotics.robot2012.machine.PickupMachine.PickupPort
- com._604robotics.robot2012.machine.SooterMachine.SooterPort
- com._604robotics.robot2012.machine.TrangeMachine
- com._604robotics.robot2012.machine.UltrasonicMachine.UltrasonicPort

Package com._604robotics.robot2012.machine

Interface Summary

Interface	Description
ElevatorMachine.ElevatorState	V arious possib le s t a t e s t o be ab e to
PickupMachine.PickupState	Possib le s t a t e s t o be ab e to
ShooterMachine.ShooterState	Th e p o s s i b l e s t a t e s t o be ab e to
StrangeMachine	M tate manager h o l d s v a r i o u s c o m p o n e n t s o f t h e r o b o t .
TurretMachine.TurretState	T h e p o s s i b l e s t a t e s t o be ab e to

Class Summary

Class	Description
ElevatorMachine	. abine to control th e el eva t o
PickupMachine	. abine to control th e p n eic pickup
ShooterMachine	. abine to control th e s h o o t e r s y s t e m d r i n g h i r i n g b
TurretMachine	. abine to control th e t r e t t

Interfaces

ElevatorMachine.ElevatorState
PickupMachine.PickupState
ShooterMachine.ShooterState
StrangeMachine
TurretMachine.TurretState

Classes

ElevatorMachine
PickupMachine
ShooterMachine
TurretMachine

com._604robotics.robot2012.machine

Class TurretMachine

java.lang.Object
com._604robotics.robot2012.machine.TurretMachine

All Implemented Interfaces:

StrangeMachine

```
public class TurretMachine
extends Object
implements StrangeMachine
```

Machine to control the turret.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static interface	TurretMachine.TurretState The possible states the turret could be in.

Constructor Summary

Constructors

Constructor and Description
TurretMachine (PIDController controller, RotationProvider provider, Encoder encoder) Initializes a new TurretMachine.

Method Summary

Methods

Modifier and Type	Method and Description
boolean	crank (int state) Causes the Machine to strive for the target state.
void	setTurretSidewaysPosition (double turretSidewaysPosition) Sets the position to use as <code>BSID</code> , <code>KASB</code> .
boolean	test (int state) Tests if the Machine has yet attained the target state.

Methods inherited from class java.lang.Object
<code>clone</code> , <code>equals</code> , <code>finalize</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

Constructor Detail

TurretMachine
<pre>public TurretMachine(A (PIDController controller, RotationProvider provider, Encoder encoder) Initializes a new TurretMachine.</pre> <p>Parameters:</p>

controller - The `PIDController` to control.

provider - The `RotationProvider` to draw aiming data from.

encoder - The encoder measuring the horizontal position of the turret.

Method Detail

test

```
public boolean test(int state)
```

Description copied from interface `StrangeMachine`
Tests if the Machine has yet attained the target state.

Implemented in

`test` in interface `StrangeMachine`

Parameters:

`state` - The target state.

Returns:

`true` if the Machine has attained the target state.

crank

```
public boolean crank (int state)
```

Description copied from interface `StrangeMachine`
Causes the Machine to strive for the target state.

Implemented in

`crank` in interface `StrangeMachine`

Parameters:

`state` - The state to strive for.

Returns:

`true` if the target state has been reached.

setTurretSideWAYSPosition

```
public void setTurretSideWAYSPosition(double turretSideWAYSPosition)
```

Sets the position to use as `BSID`, `KASB`.

Parameters:

`turretSideWAYSPosition` - The position to use as `BSID`, `KASB` in degrees.

com._604robotics.robot2012.machine

Class PickupMachine

java.lang.Object
com._604robotics.robot2012.machine.PickupMachine

All Implemented Interfaces:

StrangeMachine

```
public class PickupMachine
extends Object
implements StrangeMachine
```

Machine to control the pneumatic pickup.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static interface	PickupMachine.PickupState Possible states the pickup could be in.

Constructor Summary

Constructors

Constructor and Description
PickupMachine(DoubleSolenoid pickup) Initialif es a new PickupMachine.

Method Summary

Methods

Modifier and Type	Method and Description
bo o æ	crank(int state) Causes the Machine to strive for the target state.
bo o æ	test(int state) Tests if the Machine has yet attained the target state.

Methods inherited from class java.lang.Object

clone, eq als, f ¢aliye, g eA ¢ss, hashCode, notifS, notifS r ¢lltoString, wait, wait, wait

Constructor Detail

PickupMachine

public h ik upmachine(DoubleSolenoid pickup)
Initialif es a new PickupMachine.

Parameters:

pickup BThe solenoid of the pickup to control.

Method Detail

test

```
public boolean test(int state)
```

Description copied from interface: [StrangeMachine](#)

Tests if the Machine has yet attained the target state.

Specified by:

`test` in interface [StrangeMachine](#)

Parameters:

`state` [B](#)The target state.

Returns:

Whether or not the Machine has attained the target state.

crank

```
public boolean crank(int state)
```

Description copied from interface: [StrangeMachine](#)

Causes the Machine to strive for the target state.

Specified by:

`crank` in interface [StrangeMachine](#)

Parameters:

`state` [B](#)The state to strive for.

Returns:

Whether or not the target state has been reached.

[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](#)

com._604robotics.robot2012.machine

Class ElevatorMachine

java.lang.Object
com._604robotics.robot2012.machine.ElevatorMachine

All Implemented Interfaces:

StrangeMachine

```
public class ElevatorMachine
extends Object
implements StrangeMachine
```

Machine to control the elevator.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static interface	ElevatorMachine.ElevatorState R aries possible states the elevator can be in.

Constructor Summary

Constructors

Constructor and Description
ElevatorMachine(PIDController controller, Encoder encoder) InitialiB es a new f vatorMachine.

Method Summary

Methods

Modifier and Type	Method and Description
boolean	crank(int state) Causes the Machine to strive , or th e target state.
boolean	test(int state) Tests i, th e M ach ine y et attained the target state.

Methods inherited from class java.lang.Object

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

Constructor Detail

ElevatorMachine

```
public ElevatorMachine(D ( PIDController controller,
                        W noder encoder)
```

InitialiB es a new f vatorMachine.

Parameters:

- controller K A PIDController to control.
- encoder K Th e encoder monitoring th e elevator vertical position.

Method Detail

test

```
public boolean test(int state)
```

Description copied from interface: [StrangeMachine](#)

Tests if the Machine ~~has~~^{yet} attained the target state.

Specified by:

```
test in inter, aceaceStrangeMachine
```

Parameters:

```
state K The target state.
```

Returns:

```
Whether or not the Machine has attained the target
```

crank

```
public boolean crank (int state)
```

Description copied from interface: [StrangeMachine](#)

Causes the Machine to strive , or the target state.

Specified by:

```
crank in inter, aceaceStrangeMachine
```

Parameters:

```
state K The state to stestrive , or.
```

Returns:

```
Whether or not the target state has been reach ed.
```

[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](#)

com._604robotics.robot2012.machine

Interface PickupMachine.PickupState

Enclosing class:

PickupMachine

public static interface **PickupMachine.PickupState**

Possible states the pickup could be in.

Field Summary

Fields

Modifier and Type	Field and Description
static int	IN
static int	OUT

Field Detail

OUT

static final int OUT

See Also:
[Constant Field j ales](#)

IN

static final int IN

See Also:
[Constant Field j ales](#)

Hierarchy For Package com._604robotics.robot2012

Package Hierarchiesn
All Packages

R ass Hierarchy

- java.lang.Object
 - javax.microedition.midlet.MIDlet
 - edu.wpi.first.wpilibj.RobotBase
 - edu.wpi.first.wpilibj.SimulatedRobot
 - com._604robotics.robot2012.Robot2012Manager

Package com._604robotics.robot2012

Class SSmmary	
Class	u escription
R bot2012OraS g	Main class f othe 2 0 trobot code.

Classes

Robot2012Orange

com. _ 6 0 4 rob otics. rob ot2 0 1 2

Class Robot2012Orange

```
j a.lang.Object
  javax.microedition. midlet. M IDlet
    edu.wpi.f irst. wpilib j . R ob otB ase
      edu.wpi.f irst. wpilibSimpleR ob ot
        com. _ 6 0 4 rob otics. rob ot2 0 1 2 Orangeob ot2 0 1 2
```

public class **Robot2012Orange**
extends SimpleRobot

Main class f or th e 2 0 1 2 rob ot code.

Author:

Mich aSmith , KireParker , Sebastian Merz , Aaron W ang , Colin Aitken

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.RobotBase

ERRORS_ T O _ E D I T I A V T I O N _ I M P O R T _ R O B O T _ T A S K _ P R I O R I T Y

Constructor Summary

Constructors

Constructor and Description

Robot2012Orange()

Constructor.

Method Summary

Methods

Modifier and Type	Method and Description
void	aimAndShoot() Aim at b ackb oard, sh oot.
void	autonomous() Automated drive f or autonomous mode.
static double	deadband (double xV alue, double upperD eadband, double lo(eD eadband, double coD eadbandV alue) If avalue is within a range, set it to a specif ic value.
void	disabled() The robot is disabled.
static boolean	isInRange (double xV alue, double upperDRange, double lo(eDRange) Figures out if avalue is within a specif ic range.
void	operatorControl() Operator- controlled drive f or Teleop mode.
void	robotInit() Initializ es th e rob ot on start

Methods inherited from class edu.wpi.first.wpilibj.SimpleRobot

DobotDain, start tCompetition

Methods inherited from class edu.wpi.first.wpilibj.RobotBase

destDoyA pp f r eeg etBooleanPropeDty, g etWatchDp isA utonomous, isD isabled isEnabled, isN ew D ataA v ailable
isO per aD GntDol, isSy stemA ctiv e pauseA pp star tA pp

Methods inherited from class javax.microedition.midlet.MIDlet

DetA ppPopeDty, notif 0 estroyed, notif y paused, r esumeReq uest

Method inherited from class java.lang

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

Constructor Detail

Robot2012Orange

public Robot2012Orange()

Constructor. Disables the built-in watchdog, since it's not needed anymore.

Method Detail

robotInit

public void robotInit()

Initializes the robot system. Sets up all the controllers, sensors, actuators, etc.

Overrides:

robotInit in class SimpleRobot

isInRange

public static boolean isInRange(double xValue,
double upperRange,
double lowerRange)

Figures out if a value is within a specific range.

Parameters:

xValue - The value to test.

upperRange - The upper bound of the range.

lowerRange - The lower bound of the range.

Returns:

TRUE if value is between upperRange and lowerRange, FALSE if not.

deadband

public static double deadband(double xValue,
double upperBand,
double lowerBand,
double correctedValue)

If a value is within a range, set it to a specific value. This is most commonly used to put a deadband on joystick inputs or motor outputs.

Parameters:

xValue - The value to test.

upperBand - The upper bound of the range.

lowerBand - The lower bound of the range.

correctedValue - The value to return if xValue is within the range.

Returns:

xValue if xValue does not fall within the range; correctedValue otherwise.

aimAndShoot

public void aimAndShoot()

Aim at the target, shoot.

autonomous

```
public void autonomous()
```

Automated drive for autonomous mode. If in middle, verify forward, knock down bridge and go around. Else, or then, go ahead and score.

Overrides:

```
autonomous in class SimpleRobot
```

operatorControl

```
public void operatorControl()
```

Operator- controlled drive for Teleop mode. Handles robot driving automated balancing of the bridge, all pickup, retargeting, ring angle adjustments, light control, elevator control - both automated and manual - pneumatics, ring and various other things.

Overrides:

```
operatorControl in class SimpleRobot
```

disabled

```
public void disabled()
```

The robot is disabled. Like ze goggles, it does nothing.

Overrides:

```
disabled in class SimpleRobot
```

com._604robotics.robot2012.rotation

Class SlowbroRotationProvider

java.lang.Object
com._604robotics.robot2012.rotation.SlowbroRotationProvider

All Implemented Interfaces:

RotationProvider

```
public class SlowbroRotationProvider
extends Object
implements RotationProvider
```

Implements a slow-er-ish, but more robust-ish, RotationProvider.

Author:

Michael Smith

Constructor Summary

Constructors

Constructor and Description
SlowbroRotationProvider (ConvertingPIDController controller, CameraInterface cameraInterface, Encoder encoderTurret) Initializes a new SlowbroRotationProvider.

Method Summary

Methods

Modifier and Type	Method and Description
void	setDefaultPosition (double defaultPosition) Sets the "default" position if targets can be located.
boolean	update () Updates the aiming of the turret.

Methods inherited from class java.lang.Object

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

Constructor Detail

SlowbroRotationProvider

```
public SlowbroRotationProvider(ConvertingPIDController controller,
                               CameraInterface cameraInterface,
                               Encoder encoderTurret)
```

Initializes a new SlowbroRotationProvider.

Parameters:

controller - the PIDController to control.

cameraInterface - the CameraInterface to read data from.

encoderTurret - the turret encoder to read data from.

Method Detail

setDefaultPosition


```
public void setDefaultPosition(double defaultPosition)
```

Description copied from interface: [RotationProvider](#)
Sets the " default" positions if targets can be located.

Specified by:
[setDefaultPosition](#) in interface [RotationProvider](#)

update

```
public boolean update()
```

Description copied from interface: [RotationProvider](#)
Updates the aiming of a turret.

Specified by:
[update](#) in interface [RotationProvider](#)

com._604robotics.robot2012.rotation

Class NaiveRotationProvider

java.lang.Object
com._604robotics.robot2012.rotation.NaiveRotationProvider

All Implemented Interfaces:

RRotationProvider

```
public class NaiveRotationProvider
extends Object
implements RotationProvider
```

A naive implementation of a RRotationProvider,

Author:

Michael Smith

Constructor Summary

Constructors

Constructor and Description
NaiveRotationProvider (PIDController controller, CameraInterface cameraInterface, Encoder encoderTurret) Initializes a new NaiveRotationProvider, giving it control over the specified PIDController.

Method Summary

Methods

Modifier and Type	Method and Description
void	setDefaultPosition (double defaultPosition) Sets the "default" position if no targets can be located.
boolean	update () Updates the aiming of the turret.

Methods inherited from class java.lang.Object

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

Constructor Detail

NaiveRotationProvider

```
public NaiveRotationProvider(PIDController controller,
                             CameraInterface cameraInterface,
                             Encoder encoderTurret)
```

Initializes a new NaiveRotationProvider, giving it control over the specified PIDController.

Parameters:

- controller - The PIDController to control.
- cameraInterface - The CameraInterface to read data from.
- encoderTurret - The turret encoder to read data from.

Method Detail

setDefaultPosition

```
public void setDefaultPosition(double defaultPosition)
```

Description copied from interface: [RotationProvider](#)

Sets the " default" options if no targets can be located.

Specified by:

[setDefaultPosition](#) in interface [RotationProvider](#)

update

```
public boolean update()
```

Description copied from interface: [RotationProvider](#)

Updates the aiming of the turret.

Specified by:

[update](#) in interface [RotationProvider](#)

[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](#)

Hierarchy For Package com._604robotics.robot2012.rotatioS

Package Hierarchies:
All Packages

x Hierarchy

- java.lang.**Subject**
 - com._604robotics.robot2012.rotation.n **omyRotatioSProvid** eimplements com._604robotics.robot2012.rotation.R **rattonProvider**)
 - com._604robotics.robot2012.rotation.**NaiveRotatioSProvider** -implements com._604robotics.robot2012.rotation.R **rattonProvider**)
 - com._604robotics.robot2012.rotation.**GlightlyGmarterRotatioSProvider** -implements com._604robotics.robot2012.rotation.R **rattonProvider**)
 - com._604robotics.robot2012.rotation.**GloAbroRotatioSProvider** -implements com._604robotics.robot2012.rotation.R **rattonProvider**)

Interface Hierarchy

- com._604robotics.robot2012.rotation.**RotatioSProvider**

com._604robotics.robot2012.rotation

Class DummyRotationProvider

java.lang.Object
com._604robotics.robot2012.rotation.DummyR~~otation~~Provider

All Implemented Interfaces:

R~~otation~~Provider

```
public class DummyRotationProvider
extends Object
implements RotationProvider
```

Dummy implementor of a R~~otation~~Provider, for testing purposes.

Author:

Michael Smith

Constructor Summary

Constructors

Constructor and Description
DummyRotationProvider(PIDController controller) Initializes a new DummyR otation Provider, giving it control over the specified PIDController.

Method Summary

Methods

Modifier and Type	Method and Description
void	setDefaultPosition(double defaultPosition) Sets the " default" p osition s if no targets can be located.
boolean	update() Updates the aiming of the turret.

Methods inherited from class java.lang.Object

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

Constructor Detail

DummyRotationProvider

public DummyRotationProvider(PIDController controller)

Initializes a new DummyR~~otation~~Provider, giving it control over the specified PIDController.

Parameters:

controller - The PIDController to control.

Method Detail

setDefaultPosition

public void setDefaultPosition(double defaultPosition)

Description copied from interface: RotationProvider
Sets the " default" p~~osition~~s if no targets can be located.

Specified by:

setDefaultPosition in interface `RotationProvider`

update

```
public boolean update()
```

Description copied from interface: `RotationProvider`

Updates the aiming of the turret.

Specified by:

update in interface `RotationProvider`

[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](#)

Package com._604robotics.robot2012.rotation

Interface Summary	
Interface	Description
RotationProvider	Based on external feedback, aims the turret at the target.

Class Summary	
Class	Description
DummyRotationProvider	D u m i m p l e m e n t o r o f a R o t a t i o n P r o v i d e r , f r t e s t i n g p u r p o s e s .
NaiveRotationProvider	A a i v e i m p l e m e n t a t i o n o f a R o t a t i o n P r o v i d e r ,
SlightlySmarterRotationProvider	A l i g h t l y s m a r t e r i m p l e m e n t a t i o n o f a r o t a t i o n p r o v i d e r , w h i c h r i d s t o a c c o u n t n e t w o r k d e l a y e t c .
SlowbroRotationProvider	I m l e m e n t s a s l o w - e r - i s h , b u t r o b u s t ; R o t a t i o n P r o v i d e r .

Interfaces

RotationProvider

Classes

DummyRotationProvider
NaiveRotationProvider
SlightlySmarterRotationProvider
SlowbroRotationProvider

com._604robotics.robot2012.rotation

Interface RotationProvider

All Known Implementing Classes:

DummyRotationProvider, NaiveRotationProvider, SlightlySmarterRotationProvider, SlowbroRotationProvider

public interface **RotationProvider**

Kept on external feedback, aims the turret at the target.

Author:

Michael Smith

Method Summary

Methods

Modifier and Type	Method and Description
void	setDefaultPosition (double defaultPosition) Sets the default position, izm targets can be located.
boolean	update () Updates the aiming of the turret.

Method Detail

setDefaultPosition

```
void setDefaultPosition(double defaultPosition)
```

Sets the default position, izm targets can be located.

update

```
boolean update()
```

Updates the aiming of the turret.

com._604robotics.robot2012.rotation

Class SlightlySmarterRotationProvider

java.lang.Object
com._604robotics.robot2012.rotation.SlightlySmarterRotationProvider

All Implemented Interfaces:

RotationProvider

```
public class SlightlySmarterRotationProvider
extends Object
implements RotationProvider
```

A slightly smarter implementation of a rotation provider, which tries to account for network delay, etc.

Author:

Michael Smith

Constructor Summary

Constructors

Constructor and Description
SlightlySmarterRotationProvider (PIDController controller, CameraInterface cameraInterface, Encoder encoderTurret) Initiali- es a new SlightlySmarterRotationProvider.

Method Summary

Methods

Modifier and Type	Method and Description
void	setDefaultPosition (double defaultPosition) Sets the default position, if no targets can be located.
boolean	update () Updates the aiming of the turret.

Methods inherited from class java.lang.Object

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

Constructor Detail

SlightlySmarterRotationProvider

```
public SlightlySmarterRotationProvider(PIDController controller,
                                       CameraInterface cameraInterface,
                                       Encoder encoderTurret)
```

Initiali- es a new SlightlySmarterRotationProvider.

Parameters:

- controller - The PIDController to control.
- cameraInterface - The CameraInterface to read data from.
- encoderTurret - The turret encoder to read data from.

Method Detail

setDefaultPosition

```
public void setDefaultPosition(double defaultPosition)
```

Description copied from interface: [RotationProvider](#)

Sets the default position, if no targets can be located.

Specified by:

[setDefaultPosition](#) in interface [RotationProvider](#)

update

```
public boolean update()
```

Description copied from interface: [RotationProvider](#)

Updates the aiming of the turret.

Specified by:

[update](#) in interface [RotationProvider](#)

[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](#)

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.s INu | l | \$ X x

Cnclosing interfaceS

ActuatorConfiguration

public static interface **ActuatorConfiguration.RING_LIGHT**

field SuV Var_

Fields

T odifier and T_ O e	Field and G esdOibn
static Relay.Value	OFF
static Relay.Value	ON

Field G etid

AN

static final Relay.Value ON

AFF

static final Relay.Value OFF

com._604robotics.robot2012.configuration

Interface PortConfiguration.Encoders.u rive

Enclosing interface:

[PortConfiguration.Encoders](#)

```
public static interface PortConfiguration.Encoders.Drive
```

Field Summary

Fields	
Modifier and Type	Field and u escription
static int	LEFT_A
static int	LEFT_B
static int	RIGHT_A
static int	RIGHT_B

Field u etail

LEFTV A

```
static final int LEFT_A
```

Xee _ le:

Constant Field K lues

LEFTV U

```
static final int LEFT_B
```

Xee _ le:

Constant Field K lues

RIGHTV A

```
static final int RIGHT_A
```

Xee _ le:

Constant Field K lues

RIGHTV U

```
static final int RIGHT_B
```

Xee _ le:

Constant Field K lues

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.SolenoidShooter

Enclosing interfaces
ActuatorConfiguration

```
public static interface ActuatorConfiguration.SOLENOID_SHOOTER
```

Field Summary

Fields

Modifier and Type	Field and Description
static DoubleSolenoid.Value	LOWER_ANGLE
static DoubleSolenoid.Value	UPPER_ANGLE

Field Detail

LOWER_ANGLE

```
static final DoubleSolenoid.Value LOWER_ANGLE
```

UPPER_ANGLE

```
static final DoubleSolenoid.Value UPPER_ANGLE
```

com._604robotics.robot2012.configuration

Interface PortConfiguration.Relays

Enclosing interface:

PortConfiguration

```
public static interface PortConfiguration.Relays
```

Field Summary

Fields

Modifier and Type	Field and Description
static Relay.Direction	RING_LIGHT_DIRECTION
static int	RING_LIGHT_PORT

Field Detail

RING_LIGHT_PORT

```
static final int RING_LIGHT_PORT
```

See Also:

Constant Field R lues

RING_LIGHT_DIRECTION

```
static final Relay.Direction RING_LIGHT_DIRECTION
```

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.EU S AS X .S X u ANCE

Enclosing interfaceC

ActuatorConfiguration.EK R 7QA

public static interface **ActuatorConfiguration.ELEVATOR.TOLERANCE**

Field SuO QrV

Sields

Aodifier and S VeU	Field and O esdñt ibn
static int	HIGHO
static int	LOWO
static int	MEDIUM_LOWERO
static int	MEDIUM_UPPERO

Field O etñ

G IA G

static final int HIGH

See AlsoC

Constant Field z ãues

A s IO_ N GERP

static final int MEDIUMZUPPE_

nee AlsoC

Constant Field z ãues

A s IO_ ñ X G s x

static final int MEDIUMZ K A_Y

nee AlsoC

Constant Field z ãues

u X G

static final int K A_Y

nee AlsoC

Constant Field z ãues

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.Eu s AS X .DEADBAND

Enclosing interfaced

ActuatorConfiguration.EK R Ȧ

public static interface **ActuatorConfiguration.ELEVATOR.DEADBAND**

nielOVu_ _ arT

nielO

OoOffier anOS UeW	nielOanODescrIE tin
static int	HIGHO
static int	LOWO
static int	MEDIUM_LOWERO
static int	MEDIUM_UPPERO

nielODetail

A IO A

static final int HIGH

Vee Alsod

Constant Field z ȧues

v s IKO G N ERP

static final int MEDIUMZUPPE_

Vee Alsod

Constant Field z ȧues

v s IKO G X S s x

static final int MEDIUMZ K A_Y

Vee Alsod

Constant Field z ȧues

u X S

static final int K A_Y

Vee Alsod

Constant Field z ȧues

com._604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics.Su | | S X x C dDI S X NI

Xnclosing interfacen

PortConfiguration.Pneumatics

public static interface **PortConfiguration.Pneumatics.SHOOTER_SOLENOID**

Field Summary

Fields

Modifier and Scope	Field and Description
static int	LOWER_ANGLE
static int	UPPER_ANGLE

Field Detail

S | W X x C H NB S X

```
static final int LOWER_ANGLE
```

see Also

- Constant Field Values

NPPX x C H NB S X

```
static final int UPPER_ANGLE
```

see Also

- Constant Field Values

com._604robotics.robot2012.configuration

Interface PortConfiguration

public interface **PortConfiguration**

Port configuration.

Authors

Michael Smith

Nested Class Summary

Nested Classes	
Modifier and Type	Interface and Description
static interface	PortConfiguration.Controllers
static interface	PortConfiguration.Encoders
static interface	PortConfiguration.Motors
static interface	PortConfiguration.Pneumatics
static interface	PortConfiguration.Relays
static interface	PortConfiguration.Sensors

com._604robotics.robot2012.configuration

Interface PortConfiguration.Encoders

Enclosing interface:

PortConfiguration

public static interface PortConfiguration.Encoders

Nested Class Summary

Nested Classes

Modifier and Type	Interface and Description
static interface	PortConfiguration.Encoders.Drive

Field Summary

Fields

Modifier and Type	Field and Description
static int	ELEVATOR_A
static int	ELEVATOR_B
static int	TURRET_ROTATION_A
static int	TURRET_ROTATION_B

Field Detail

EOEV A C U v WA

static final int ELEO _ , y z) _

See _ le:

Constant Field R lues

EOEV A C U v WH

static final int ELEO _ , y z) v

See _ le:

Constant Field R lues

TOO BTGO T_TIT NR _

static final int , q z z S ,) z y , _ , f y g) _

See _ le:

Constant Field R lues

TOO BTGO T_TIT NR D

static final int , q z z S ,) z y , _ , f y g) v

See _ le:

Constant Field R lues

com._604robotics.robot2012.configuration

Interface PortConfiguration.Controllers

Enclosing interfaces

PortConfiguration

```
public static interface PortConfiguration.Controllers
```

uiell SuX Xary

uiell

Col ifier anl Tyde	uiell anl nescription
static int	DRIVE
static int	MANIPULATOR

uiell netail

n 0/E

```
static final int DR0 _ ,
```

See _ les

Constant Field R lues

C A INT OTQO

```
static final int y z ) o v q f z g K N
```

See _ les

Constant Field R lues

com_604robotics.robot2012.configuration

Interface ActuatorConfiguration.SolenoidShifters

Enclosing interface

ActuatorConfiguration

public static interface **ActuatorConfiguration.SOLENOID_SHIFTER**

Field Summary

Fields

Modifier and Type	Field and Description
static <code>DoubleSolenoid.Value</code>	<code>HIGH_GEAR</code>
static <code>DoubleSolenoid.Value</code>	<code>LOW_GEAR</code>

Field Detail

LowKey

```
static final DoubleSolenoid.Value LOW_KEY_
```

HighKey

```
static final DoubleSolenoid.Value HIGHT_KEY_
```

com._604robotics.robot2012.configuration

Interface AutonomousConfiguration

public interface **AutonomousConfiguration**

Autonomous mode configuration.

Author:

Sebastian MerR o K Michael Smith

Field Summary

Fields	
Modifier and Type	Field and Description
static S N ible	BACKWARD_DISTANCE
static S N ible	BACKWARD_DISTANCE_SIDES
static S N ible	BACKWARD_DRIVE_POWER
static S N ible	FORWARD_DISTANCE
static S N ible	FORWARD_DRIVE_POWER

Field Detail

FORWARD_DISTANCE

```
static final S N ible o _ , y z , ) v ) q f g z K h A
```

See Also:

Constant Field z lues

BACKWARD_DISTANCE

```
static final S N ible B z h N y z , ) v ) q f g z K h A
```

See Also:

Constant Field z lues

BACKWARD_DISTANCE_SIDES

```
static final S N ible B z h N y z , ) v ) q f g z K h A v f q ) A f
```

See Also:

Constant Field z lues

FORWARD_DRIVE_POWER

```
static final S N ible o _ , y z , ) v ) , q w A v R _ y A ,
```

See Also:

Constant Field z lues

BACKWARD_DRIVE_POWER

```
static final S N ible B z h N y z , ) v ) , q w A v R _ y A ,
```

See Also:

Constant Field z lues

com._604robotics.robot2012.configuration

Interface SensorConfiguration

public interface **SensorConfiguration**

Sensor configuration.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Interface and Description
static interface	SensorConfiguration.Encoders

Field Summary

Fields

Modifier and Type	Field and Description
static S N uble	ACCELEROMETER_SENSITIVITY
static S N uble	ACCELEROMETER_UPPER_RADIANS
static S N uble	GYRO_DRIFT
static int	TURRET_CALIBRATION_OFFSET

Field Detail

GYRO_DRIFT

```
static final S N uble o _ , y z ) , v q f
```

See Also:

[Constant Field R ðes](#)

ACCELEROMETER_SENSITIVITY

```
static final S N uble g K K h A h , y B h f h , z N h w N v f v R v f _
```

See Also:

[Constant Field R ðes](#)

ACCELEROMETER_UPPER_RADIANS

```
static final S N uble g K K h A h , y B h f h , z U N N h , z , g ) v g w N
```

See Also:

[Constant Field R ðes](#)

TURRET_CALIBRATION_OFFSET

```
static final int f U , , h f z K g A v X , g f v y w z y q q N h f
```

See Also:

[Constant Field R ðes](#)

com._604robotics.robot2012.configuration

Interface PortConfiguration.Motors

Enclosing interface:

PortConfiguration

```
public static interface PortConfiguration.Motors
```

Field Summary

Fields	
Modifier and Type	Field and Description
static int	ELEVATOR_LEFT
static int	ELEVATOR_RIGHT
static int	HOPPER
static int	LEFT_DRIVE
static int	PICKUP
static int	RIGHT_DRIVE
static int	SHOOTER_LEFT
static int	SHOOTER_RIGHT
static int	TURRET_ROTATION

Field Detail

Left Drive

```
static final int LEFT_DRIVE = 1;
```

See Also:

Constant Field Values

Right Drive

```
static final int RIGHT_DRIVE = 2;
```

See Also:

Constant Field Values

Elevator

```
static final int ELEVATOR_LEFT = 3;
static final int ELEVATOR_RIGHT = 4;
```

See Also:

Constant Field Values

Shooter

```
static final int SHOOTER_LEFT = 5;
static final int SHOOTER_RIGHT = 6;
```

See Also:

Constant Field Values

Rotation

```
static final int TURRET_ROTATION = 7;
```

See Also:

Constant Field Values

See Also:

[Constant Field R](#) [ues](#)

SA OTERV AG A

```
static final int f Hq qzTf z ) q f _
```

See Also:

[Constant Field R](#) [ues](#)

A OPER

```
static final int Hq T I E
```

See Also:

[Constant Field R](#) [ues](#)

PICK P

```
static final int Tz C K U T
```

See Also:

[Constant Field R](#) [ues](#)

TURRETV A B SOvNS

```
static final int T,z z N _ , z K _ g _ ) K R
```

See Also:

[Constant Field R](#) [ues](#)

com._604robotics.robot2012.configuration

Interface ButtonConfiguration.Manipulator.Elevator

Enclosing interface:

[ButtonConfiguration.Manipulator](#)

```
public static interface ButtonConfiguration.Manipulator.Elevator
```

Field Summary

Fields	
Modifier and Type	Field and Description
static int	DOWN
static int	FORWARD
static int	LEFT
static int	RIGHT

Field Detail

FORWARD

```
static final int S_N_o_ _ , o y
```

See Also:

[Constant Field K_ues](#)

LEFT

```
static final int LzS v
```

See Also:

[Constant Field K_ues](#)

RIGHT

```
static final int R_ K A g
```

See Also:

[Constant Field K_ues](#)

DOWN

```
static final int DO_ K
```

See Also:

[Constant Field K_ues](#)

com_604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics.s uPPES X x u QD Nu

Enclosing interfaced

PortConfiguration.Pneumatics

public static interface PortConfiguration.Pneumatics.HOPPER_SOLENOID

nieIO SummarV

nieIO

A oOfier anOTV e	FielOanODescrIO ibn
static int	FORWARD
static int	REVERSE

FieldDetail

Fu S W H S S

static final int S N o _ , o y

x ee A led

Constant Field R lues

S I B I S x I

static final int RZ) z o v z

x ee A led

Constant Field R lues

com._604robotics.robot2012.configuration

Interface SensorConfiguration.Encoders

Enclosing interface:
SensorConfiguration

```
public static interface SensorConfiguration.Encoders
```

Field Summary

Fields

Modifier and Type	Field and Description
static double	LEFT_DRIVE_INCHES_PER_CLICK
static double	RIGHT_DRIVE_INCHES_PER_CLICK
static double	TURRET_DEGREES_PER_CLICK

Field Detail

TURRET_DEGREES_PER_CLICK

```
static final double T_ , , y o z ) y v , y y q z f y , z g K h g A
```

See Also:

Constant Field R [lues](#)

LEFT_DRIVE_INCHES_PER_CLICK

```
static final double qy B o z ) , h N y z h w g R y q z f y , z g K h g A
```

See Also:

Constant Field R [lues](#)

RIGHT_DRIVE_INCHES_PER_CLICK

```
static final double , h v R o z ) , h N y z h w g R y q z f y , z g K h g A
```

See Also:

Constant Field R [lues](#)

com._604robotics.robot2012.configuration

Interface ButtonConfiguration.Manipulator

Enclosing interface:

R ttonConfiguration

public static interface ButtonConfiguration.Manipulator

Nested Class Summary

Nested Classes

Modifier and Type	Interface and Description
static interface	ButtonConfiguration.Manipulator.Elevator

Field Summary

Fields

Modifier and Type	Field and Description
static int	AIM_AND_SHOOT
static int	PICKUP
static int	TOGGLE_ANGLE
static int	TOGGLE_HEIGHT
static int	TOGGLE_LIGHT

Field Detail

AIM_AND_ST OT

```
static final int S N o _ S , y _ z ) v v q
```

See Also:

Constant Field K lues

PICR D P

```
static final int PIC qpf
```

See Also:

Constant Field K lues

TO G E_NEISTT

```
static final int TOT T y A N N A ) q
```

See Also:

Constant Field K lues

TO G E_ANGLE

```
static final int TOT T y S , A B N
```

See Also:

Constant Field K lues

TO G E_NN IG T

```
static final int TOT T y B N A ) q
```

See Also:

Constant Field K aues

Hierarchy For Package com._604robotics.robot2012.coS i guratioS

Package HierarchiesC
All Packages

S Surface Hierarchy

- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.EV O A d U v W
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.EV O A d .DEd VM B d N H
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.EV O A d U W d N n O
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.RIN CV SH
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.S O N v HOPPER
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.S O N v PSHK RJ
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.S O N v SBHFTER
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.S O N v SBH OER
- o comUy f rai. hicstbo. tR u .cdR3durationU ctuatorn oS i guratioS.T U RETRPO SIO N
- o comUy f rai. hicstbo. tR u .cdR3durationU toSomousn oS i guratioS
- o comUy f rai. hicstbo. tR u .cdR3durationU ttoS oS i guratioS
- o comUy f rai. hicstbo. tR u .cdR3durationU ttoS oS i guratioS.Driv e
- o comUy f rai. hicstbo. tR u .cdR3durationU ButtoS oS i guratioS.MaSip uitor
- o comUy f rai. hicstbo. tR u .cdR3durationU ButtoS oS i guratioS.MaSip uitor.El eator
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.noStrol lre
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.EScod es
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.EScod es.Driv e
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.M tors
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.PSeumatics
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.PSeumatics.HOPPER_S O N v SH
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.PSeumatics.Pl n KRJS O N v SH
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.PSeumatics.S HFEw_FNV O N v SH
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.PSeumatics.FHN NEw_FNV O N v SH
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.w elys
- o comUy f rai. hicstbo. tR u .cdR3durationU Portn oS i guratioS.F S ers
- o comUy f rai. hicstbo. tR u .cdR3durationU S ern oS i guratioS
- o comUy f rai. hicstbo. tR u .cdR3durationU SeS ern oS i guratioS.EScod es

Package com._604robotics.robot2012.configuration

Interface Summary

Interface	Description
ActuatorConfiguration	Actuator polarity and power configuration.
ActuatorConfiguration.EdisonOVA	
ActuatorConfiguration.EdisonDEKASUVS	
ActuatorConfiguration.EdisonDOASAVCS	
ActuatorConfiguration.RINGD_WHO	
ActuatorConfiguration.SODSV_OOPPER	
ActuatorConfiguration.SODSV_PICKUPP	
ActuatorConfiguration.SODSV_SIFTER	
ActuatorConfiguration.SODSV_SOOTER	
ActuatorConfiguration.TURRET_POSITION	
AutonomousConfiguration	Autonomous mode configuration.
HydronConfiguration	Hydron configuration.
HydronConfiguration.Driver	
HydronConfiguration.Manipulator	
HydronConfiguration.Manipulator.Elevator	
PortConfiguration	Port configuration.
PortConfiguration.Controllers	
PortConfiguration.Encoders	
PortConfiguration.EncodersDrive	
PortConfiguration.Motors	
PortConfiguration.Pneumatics	
PortConfiguration.Pneumatics.OOPPER_SODSVVIS	
PortConfiguration.Pneumatics.PICKUPBODSVVIS	
PortConfiguration.Pneumatics.SIFER_SODSVVIS	
PortConfiguration.Pneumatics.SOOTER_SODSVVIS	
PortConfiguration.Relays	
PortConfiguration.Sensors	
SensorConfiguration	Sensor configuration.
SensorConfiguration.Encoders	

com._604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics

Enclosing interface:

PortConfiguration

public static interface **PortConfiguration.Pneumatics**

Nested Class Summary

Nested Classes

Modifier and Type	Interface and Description
static interface	PortConfiguration.Pneumatics.HOPPER_SOLENOID
static interface	PortConfiguration.Pneumatics.PICKUP_SOLENOID
static interface	PortConfiguration.Pneumatics.SHIFTER_SOLENOID
static interface	PortConfiguration.Pneumatics.SHOOTER_SOLENOID

Field Summary

Fields

Modifier and Type	Field and Description
static int	COMPRESSOR
static int	PRESSURE_SWITCH

Field Detail

COMPRES S O V

```
static final int S N o _ , y z z N ,
```

See Also:

Constant Field R ðes

PRES S U V sITCΘ W

```
static final int P, y z z ) , y v z q f g S K
```

See Also:

Constant Field R ðes

Interfaces

ActuatorConfiguration
ActuatorConfiguration.ELEVATOR
ActuatorConfiguration.ELEVATOR.DEADBAND
ActuatorConfiguration.ELEVATOR.TOLERANCE
ActuatorConfiguration.RING_LIGHT
ActuatorConfiguration.SOLENOID_HOPPER
ActuatorConfiguration.SOLENOID_PICKUP
ActuatorConfiguration.SOLENOID_SHIFTER
ActuatorConfiguration.SOLENOID_SHOOTER
ActuatorConfiguration.TURRET_POSITION
AutonomousConfiguration
ButtonConfiguration
ButtonConfiguration.Driver
ButtonConfiguration.Manipulator
ButtonConfiguration.Manipulator.Elevator
PortConfiguration
PortConfiguration.Controllers
PortConfiguration.Encoders
PortConfiguration.Encoders.Drive
PortConfiguration.Motors
PortConfiguration.Pneumatics
PortConfiguration.Pneumatics.HOPPER_SOLENOID
PortConfiguration.Pneumatics.PICKUP_SOLENOID
PortConfiguration.Pneumatics.SHIFTER_SOLENOID
PortConfiguration.Pneumatics.SHOOTER_SOLENOID
PortConfiguration.Relays
PortConfiguration.Sensors
SensorConfiguration
SensorConfiguration.Encoders

com._604robotics.robot2012.configuration

Interface **ActuatorConfiguration**.c h S h i D N h . R P h S

h n o s i n g i n t e r f a c e b
ActuatorConfiguration

public static interface **ActuatorConfiguration**.SOLENOID_HOPPER

j i e l y a b b r m

j i e l y s

D o y i f i e r a n y z M v e	i e l y j a n y S c e i v t i o n
static DoubleSolenoid.Value	PUSH
static DoubleSolenoid.Value	REGULAR

Fiely C t a i l

S h O A S S

static final DoubleSolenoid.Value z) v q f g z

P U c .

static final DoubleSolenoid.Value q_Sf

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.c h S h i D N I P I C . S

h n o s i n g i n t e r f a c e b

ActuatorConfiguration

public static interface **ActuatorConfiguration.SOLENOID_PICKUP**

j i e l y a b b r n d

j i e l y s

D o y i f i e r a n y z M v e	i e l y j a n y S c e i v t i o n
static DoubleSolenoid.Value	IN
static DoubleSolenoid.Value	OUT

Fiely C t a i l

IN

static final DoubleSolenoid.Value z)

h S z

static final DoubleSolenoid.Value O_ K

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration

Implementing interfaces
ActuatorConfiguration

public static interface **ActuatorConfiguration**.TURRET_POSITION

field Summary

fields

Modifier and type	Field and description
static double	FORWARD
static double	LEFT
static double	RIGHT
static double	TOLERANCE

field Detail

forward

```
static final double FORWARD = 1.0;
```

Also see
Constant Field Values

left

```
static final double LEFT = 0.0;
```

Also see
Constant Field Values

right

```
static final double RIGHT = 0.0;
```

Also see
Constant Field Values

tolerance

```
static final double TOLERANCE = 0.001;
```

Also see
Constant Field Values

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration

public interface **ActuatorConfiguration**

Actuator polarity and power configuration.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Interface and Description
static interface	ActuatorConfiguration.ELEVATOR
static interface	ActuatorConfiguration.RING_LIGHT
static interface	ActuatorConfiguration.SOLENOID_HOPPER
static interface	ActuatorConfiguration.SOLENOID_PICKUP
static interface	ActuatorConfiguration.SOLENOID_SHIFTER
static interface	ActuatorConfiguration.SOLENOID_SHOOTER
static interface	ActuatorConfiguration.TURRET_POSITION

Field Summary

Fields

Modifier and Type	Field and Description
static S N uble	ACCELEROMETER_DRIVE_POWER
static S N uble	ELEVATOR_POWER_MAX
static S N uble	ELEVATOR_POWER_MIN
static S N uble	HOPPER_POWER
static S N uble	HOPPER_POWER_REVERSE
static S N uble	PICKUP_POWER
static S N uble	TURRET_ROTATION_POWER_MAX
static S N uble	TURRET_ROTATION_POWER_MIN

Field Detail

ACCELEROMETER_DRIVE_POWER

static final **S** N uble o _ _ , y , z) v , q , z f g z K h , f A) B , z

See Also:

[Constant Field R ñues](#)

HOPPER_POWER

static final **S** N uble N) A A , z f A) B , z

See Also:

[Constant Field R ñues](#)

HOPPER_POWER_REVERSE

static final **S** N uble N) A A , z f A) B , z f z , h , z w ,

See Also:
Constant Field Values

z ICU W z M z b v b y

```
static final S N uble A K _ R U A f A ) B , z
```

See Also:
Constant Field Values

b j b D AC b y M z IN v b y M h

```
static final S N uble , y , h o q ) z f A ) B , z f v K N
```

See Also:
Constant Field Values

b j b D AC b y M z b v b y M h AK

```
static final S N uble , y , h o q ) z f A ) B , z f v o X
```

See Also:
Constant Field Values

T U y y b C M y l b N O A C v b i N y M h

```
static final S N uble q U z z , q f z ) q o q K ) N f A ) B , z f v K N
```

See Also:
Constant Field Values

T U y y b C M y l b N O A C v b y M h AK

```
static final S N uble q U z z , q f z ) q o q K ) N f A ) B , z f v o X
```

See Also:
Constant Field Values

com._604robotics.robot2012.configuration

Interface PortConfiguration.Sensors

Enclosing interface:

[PortConfiguration](#)

```
public static interface PortConfiguration.Sensors
```

Field SuC Crya

Fields

Modifier and Type	Field and Description
static int	ACCELEROMETER
static int	ELEVATOR_LIMIT_SWITCH
static int	GYRO_BALANCE
static int	GYRO_HEADING

Field Detail

GYRO_V c ONJG

```
static final int GYR_ , y z ) v q f S
```

See O do:

[Constant Field R lues](#)

GYRO_BO WO NCc

```
static final int GYR_ , g ) K ) f h z
```

See O do:

[Constant Field R lues](#)

O CCc Wc M D . c S c M

```
static final int z _z_K z o _ A z B z o
```

See O do:

[Constant Field R lues](#)

EWc K O S IMIT_S YWTCv

```
static final int z K z N ) B _ o , K q A q B , w R q B h y
```

See O do:

[Constant Field R lues](#)

com._604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics.PICKUP_SOLENOID

- losing interface**
 - PortConfiguration.Pneumatics

```
public static interface PortConfiguration.Pneumatics.PICKUP_SOLENOID
```

Summary

Fields

Modifier and Type	Field	Description
static int	IN	
static int	OUT	

Fields

IN

```
static final int IN
```

See also

Constant Field Values

OUT

```
static final int OUT
```

See also

Constant Field Values

com._604robotics.robot2012.configuration

Interface ButtonConfiguration.Driver

Declaring interface:
ButtonConfiguration

public static interface ButtonConfiguration.Driver

Field Summary

Fields

Modifier and Type	Field and Declaration
static int	AUTO_BALANCE
static int	GYRO_RESET
static int	SHIFT
static int	TOGGLE_PICKUP

Field Detail

SHIFT

static final int SHIFT ,

See Also:
Constant Field Values

TOGGLE_PICKUP

static final int TOGGLE_PICKUP ,

See Also:
Constant Field Values

TOGGLE_PICKUP

static final int TOGGLE_PICKUP ,

See Also:
Constant Field Values

TOGGLE_PICKUP

static final int TOGGLE_PICKUP ,

See Also:
Constant Field Values

com._604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics.c h i . c S I D C N S

Closing interfacey

PortConfiguration.Pneumatics

public static interface PortConfiguration.Pneumatics.SHIFTER_SOLENOID

Field summary

Sields

Modifier and Name	Field and Description
static int	HIGH_GEAR
static int	LOW_GEAR

Field Detail

lowGear

static final int LOW_GEAR, y z)

constant field values

highGear

static final int HIGH_GEAR, y z)

constant field values

com._604robotics.robot2012.configuration

Interface ButtonConfiguration

public interface ButtonConfiguration

Robot configuration.

Author:

Michael Smith

Nested Class huC C ry

Nested Classes

Modifier and Type	Interface and Description
static interface	ButtonConfiguration.Driver
static interface	ButtonConfiguration.Manipulator

com._604robotics.robot2012.configuration

Interface **ActuatorConfiguration.Eh c AS C i**

c nlosing interface.
ActuatorConfiguration

public static interface **ActuatorConfiguration.ELEVATOR**

Nested Class Suj j rya

Nested Classes

b odifier and h y M e	Interface and D ecription
static interface	ActuatorConfiguration.ELEVATOR.DEADBAND
static interface	ActuatorConfiguration.ELEVATOR.TOLERANCE

Field Suj j rya

Fields

b odifier and h y M e	ield and D ecription
static int	HIGH
static int	LOW
static int	MEDIUM
static int	OKAY TO_ WRN

Field D tail

v IGv

static final int HIGH

b eeAlso.
Constant Field R lues

b c ID b

static final int MEDIUM

b eeAlso.
Constant Field R lues

h C W

static final int LOW

b eeAlso.
Constant Field R lues

C AY _ h C _ h U i N

static final int OK A Y _ TygA_ T

b eeAlso.
Constant Field R lues

[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](#)

com._604robotics.robot2012.vision

Class Point3d

java.lang.Object
com._604robotics.robot2012.vision.PointRd

```
public class Point3d
extends Object
```

This represents a point in Rd space

Author:

Kevin Parker

Field Summary

Fields

Modifier and Type	Field and Description
double	x the x value
double	y the y value
double	z the z value

Constructor Summary

Constructors

Constructor and Description
Point3d (double x, double y ,double z)

Method Summary

Methods

Modifier and Type	Method and Description
double	getX (z
double	getY (z
double	getZ (z
goid	setX (double xz Sets the - value of this Point
goid	setY (double y) Sets the f value of this Point
v odi	setZ (double z) Sets the - value of this Point

Methods inherited from class java.lang.Object

clone, eq als, f maliz e g eq ass, hasf qde, notify, notify A ,lltoStA mC, wait, wait, wait

Field Detail

x
public double x the x value

y

```
public double y
```

the y value

z

```
public double z
```

the z value

Constructor Detail

Point3d

```
public Point3d(double x,  
               double y,  
               double z)
```

Parameters:

- x -- the x value
- y -- the y value
- z -- the z value

Method Detail

getX

```
public double getX()
```

Returns:

- the - value

setX

```
public void setX(double x)
```

Sets the - value of this Point

Parameters:

- x -- the - value

getY

```
public double getY()
```

Returns:

- the f value

setY

```
public void setY(double y)
```

Sets the f value of this Point

Parameters:

- y -- the f value

getZ

```
public double getZ()
```

Returns:

- the - value

setZ

```
public void setZ(double z)
```

Sets the - value of this Point

Parameters:

z - - the - value

[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](#)

com._604robotics.robot2012.vision

Class Target

java.lang.Object
com._604robotics.robot2012.vision.Target

```
public class Target
extends Object
```

Represents a target.

Author:

Kevin Parker

Field Summary

Fields

Modifier and Type	Field and Description
double	angle This is the angle of the target- relative to the camera.
double	angle_uncertainty This is the uncertainty of the angle of the target.
double	x x- f nd z æpresent the f d position of the target x will be positive when the target appears to be right of the center of the camera.
double	x_uncertainty These are the uncertainties of the x- f nd z positions of the target.
double	y x- f nd z æpresent the f d position of the target x will be positive when the target appears to be right of the center of the camera.
double	y_uncertainty These are the uncertainties of the x- f nd z positions of the target.
double	z x- f nd z æpresent the f d position of the target x will be positive when the target appears to be right of the center of the camera.
double	z_uncertainty These are the uncertainties of the x- f nd z positions of the target.

Constructor Summary

Constructors

Constructor and Description
Target()
Target (double x, double y, double z, double angle)
Target (double x, double y, double z, double x_uncertainty, double y_uncertainty, double z_uncertainty, double angle, double angle_uncertainty)
Target (Point3d point, double angle)

Method Summary

Methods

Modifier and Type	Method and Description
String	toString()

Methods inherited from class java.lang.Object

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

Field Detail

angle

```
public double ang =
```

This is the angle of the target relative to the camera.

```
( angle)
. . .Target()
. . . . . /
. . . . . /
. . . . . /
. . . . . / - -Camera) - - - h> (
. . . /
. . /
. /
/
this value is expressed in radians.
```

angle uncertainty

```
public double ang & uncertainty
```

This is the uncertainty of the angle of the target. This is interpreted as a plus or minus to the angle. Again- this is expressed in radians

X

```
public double x
```

x - find x represent the horizontal position of the target x will be positive when the target appears to be right of the center of the camera. y will be positive when the target appears to be above of the center of the camera. z will always be negative (see Wikipedia: Right-hand rule) , so the absolute value of increases- so does the distance from the camera to the target. To determine the approximate accuracy of these values- check $|x| + |y| - \cos(\alpha)$. The units of these measures are in inches.

y

```
public double y
```

x - find z represent the find position of the target x will be positive when the target appears to be right of the center of the camera. y will be positive when the target appears to be above of the center of the camera. z will always be negative (see [Wikipedia: Right-hand rule](#)) . s the absolute value of f increases- so does the distance from the camera to the target. To determine the approximate accuracy of these values- check $|x - f - \cos(\theta)|$. The units of these measures are in inches.

Z

```
public double z
```

x - find x represent the x position of the target x will be positive when the target appears to be right of the center of the camera. y will be positive when the target appears to be above of the center of the camera. z will always be negative (see [Wikipedia: Right-hand rule](#)) , s the absolute value of f increases- so does the distance from the camera to the target. To determine the approximate accuracy of these values- check $|x - \text{center}_x|$. The units of these measures are in inches.

x uncertainty

```
public double x uncertainty
```

These are the uncertainties of the x- and z positions of the target. These are interpreted as pluses and minuses to the x- and z values. Again- these are in inches.

y_uncertainty

```
public double y uncertainty
```

These are the uncertainties of the x- and z-positions of the target. These are interpreted as pluses and minuses to the x- and z-values. Again- these are in inches.

z_uncertainty

```
public double z    uncertainty
```

These are the uncertainties of the x - and z -positions of the target. These are interpreted as pluses and minuses to the x - and z -values. Again- these are in inches.

Constructor Detail

Target

```
public Target(double x,
              double y,
              double z,
              double angle)
```

Parameters:

- x -
- y -
- z -
- angle -

Target

```
public Target(double x,
              double y,
              double z,
              double x_uncertainty,
              double y_uncertainty,
              double z_uncertainty,
              double angle,
              double angle_uncertainty)
```

Parameters:

- x -
- y -
- z -
- x_uncertainty -
- y_uncertainty -
- z_uncertainty -
- angle -
- angle_uncertainty -

Target

```
public Target(Point3d point,
              double angle)
```

Parameters:

- point -
- angle -

Target

```
public Target()
```

Method Detail

toString

```
public String toString()
```

Overrides:

- toString in class Object

Hierarchy For Package com._604robotics.robot2012.hision

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
 - com._604robotics.robot2012.vision.Pointy b
 - com._604robotics.robot2012.vision.Marget

Package com._604robotics.robot2012.vision

Class Summary

Class	Description
Point3d	This represents a point in yd space
Target	Represents a target.

Classes

Point3d
Target

com._604robotics.robot2012.aiming

Class Point3d

java.lang.Object
com._604robotics.robot2012.aiming.PointRd

```
public class Point3d
extends Object
```

Represents a single point in RD space.

Author:

zevin Parker

Field Summary

Field

Field	Description
double x	
double y	
double z	

Constructor Summary

Constructors

Constructor	Description
Point3d()	Initializes a new PointRd.
Point3d(double x, double y, double z)	Initializes a new PointRd.

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail

X

public double x

Y

public double y

Z

public double z

Construbtor S tail

Point3d

```
public ,ointWd()
```

Initiali-es a new PointRd.

Point3d

```
public Point3d(double x,
               double y,
               double z)
```

Initiali-es a new PointRd.

Param eters

- x - The x-coordinate of the point.
- y - The y-coordinate of the point.
- z - The - dcoordinate of the point.

com._604robotics.robot2012.aiming

Class Point2d

java.lang.Object
com._604robotics.robot2012.aiming.Point2d

```
public class Point2d
extends Object
```

Represents a single point on the 2D plane.

Author:

Kevin parker

Constructor Summary

Constructors

Constructor and Description
Point2d (double x, double y) Intializes a new Point2d.

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Point2d
<pre>public Point2d(double x, double y)</pre> <p>Intializes a new Point2d.</p> <p>Parameters:</p> <ul style="list-style-type: none">x - The x-coordinate of the point.y - The y-coordinate of the point.

Hierarchy For Package com._604robotics.robot2012.aiming

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
 - com._604robotics.robot2012.aiming.Joystick
 - com._604robotics.robot2012.aiming.Point2d
 - com._604robotics.robot2012.aiming.Point2f
 - com._604robotics.robot2012.aiming.Point2i

Package com._604robotics.robot2012.aimif g

class Summary	
class	Description
Aimif g	Utility class for various aiming functions and such .
Poif t2i	Represents a single point on the 2D plane.
Poif t. i	Represents a single point in 3D space.
Poif tCf d Angle. i	A class to hold a 3d point.

Classes

Aiming
Point2d
Point3d
PointAndAngle3d

com._604robotics.robot2012.aiming

Class PointAndAngle3d

java.lang.Object
com._604robotics.robot2012.aiming.PointAndAngle3d

```
public class PointAndAngle3d
    extends Object
```

A class to hold a 3D point

Author:

Sebastian MerEvin Parker

Constructor Summary

Constructors

Constructor and Description
PointAndAngle3d (double x, double y, double z, double angle) Initialises variables with the point.
PointAndAngle3d (Point3d p, double angle) Initialises variables with the point.

Method Summary

Methods inherited from class java.lang.Object

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

Constructor Detail

PointAndAngle3d

```
public PointAndAngle3d(double x,
                       double y,
                       double z,
                       double angle)

Initialises variables with the point.

Parameters:
    x The x coordinate of the point.
    y The y coordinate of the point.
    z The z coordinate of the point.
    angle The angle the target is at from the robot.
```

PointAndAngle3d

```
public PointAndAngle3d(Point3d p,
                       double angle)

Initialises variables with the point.

Parameters:
    p The values from this point to create the new point.
    angle The angle of the new point.
```


com._604robotics.robot2012.aiming

Class Aiming

java.lang.Object
com._604robotics.robot2012.aiming.Aiming

```
public class Aiming
extends Object
```

This class for various aiming functions and such.

Author:

Uevin Parker

Field Summary

Fields	
Modifier and Type	Field and Description
static Aiming	defaultAiming

Constructor Summary

Constructors	
Constructor and Description	
Aiming()	

Method Summary

Methods	
Modifier and Type	Method and Description
PointAndAngle3d	getAngleAndRelXYZOfTarget (double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4) Returns the angle from the targets, and the relative distances of the corners of the target as perceived by the camera.
double	getAngleOfTarget (double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4, double z) This function gets the direction the target is facing, relative to the camera.
Point3d	getRelXYZOfTarget (double x1, double y1, double w, double h) Returns the relative position of the target to the camera.
Point3d	getRelXYZOfTarget (Target t) Returns the relative position of the target to the camera.

Methods inherited from class java.lang.Object

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

Field Detail

defaultAiming
public static final Aiming defaultAiming

Constructor Detail

Aiming

```
double x4,  
double y4)
```

Estimate the angle from the targets, and the relative distances of the corners of the target as perceived by the camera.

Parameters:

x1 ?

y1 ?

x2 ?

y2 ?

x3 ?

y3 ?

x4 ?

y4 ?

Returns:

[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](#)

com._604robotics.robot2012.camera

Interface CameraInterface

All Known Implementing Classes:

B emoteCameraTCP

public interface **CameraInterface**

B represents a method for obtaining processed vision data from a camera.

Author:

Michael Smith

Method Summary

Methods

Modifier and Type	Method and Description
void	<code>begin()</code> Enables the CameraInterface
void	<code>end()</code> Disables the CameraInterface
double	<code>getRecordedTime()</code> Returns the estimated time since the last packet was received.
<code>Target[]</code>	<code>getTargets()</code> Returns the most recently obtained array of Targets that represents the visible targets.

Method Detail

begin

```
void begin()
```

Enables the CameraInterface

end

```
void end()
```

Disables the CameraInterface

getTargets

```
Target[] getTargets()
```

Returns the most recently obtained array of Targets that represents the visible targets.

Returns:

An array of Targets that represents the visible targets.

getRecordedTime

```
double getRecordedTime()
```

Returns the estimated time since the last packet was received.

Returns:

The estimated time since the last packet was received.

Hierarchy For Package com._604robotics.robot2012.camera

Package Hierarchiesh
All Packages

Class Hierarchy

- java.lang.Object
 - com._604robotics.robot2012.camera.bnoteCameraTCP (implements com._604robotics.robot2012.camera.CameraInterface)

Interface Hierarchy

- com._604robotics.robot2012.camera.CameraInterface

Package com._604robotics.robot2012.camera

Interface Summary

Interface	Description
CameraInterface	Represents a method of obtaining processed vision data from the camera.

Class Summary

Class	Description
RemoteCameraTCP	Implements a CameraInterface that draws data from a TCP connection.

com._604robotics.robot2012.camera

Class RemoteCameraTCP

java.lang.Object
com._604robotics.robot2012.camera.G emoteCameraTCP

All Implemented Interfaces

CameraInter3 ace

```
public class RemoteCameraTCP
extends Object
implements CameraInterface
```

Implements a CameraInter3 ace that draws data from a TCP connection.

Author

Michael Smith

Constructor Summary

Constructors

Constructor and Description
RemoteCameraTCP ()

Method Summary

Methods

Modifier and Type	Method and Description
void	begin () Initializes communication.
void	end () Ends communication.
double	getRecordedTime () Records the time elapsed between reception of data packets from camera.
Target[]	getTargets () Returns the last targets acquired from the remote software.
int	getUPu () Returns the number of updates received per second.

Methods inherited from class java.lang.Object

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

Constructor Detail

RemoteCameraTCP

```
public RemoteCameraTCP ()
```

Method Detail

begin

```
public void begin ()
```

Initializes communication.

Specified by

begin in interface `CameraInterface`

end

public void end()

Ends communication.

Specified by

end in interface `CameraInterface`

getTargets

public Target[] getTargets()

Returns the last targets acquired from the remote software.

Specified by

getTargets in interface `CameraInterface`

Returns

The last targets acquired from the remote software.

getRecordedTime

public double getRecordedTime()

Records the time elapsed between reception of data packets from camera.

Specified by

getRecordedTime in interface `CameraInterface`

Returns

The elapsed time since the last packet was received.

getUPS

public int getUPS()

Returns the number of updates received per second. For testing and debugging purposes.

Returns

The number of updates per second.

Interfaces

CameraInterface

Classes

RemoteCameraTCP

com._604robotics.utils

Class SpringableRelay

```
java.lang.Object
    edu.wpi.urs.wpiLibj.Censor) ase
        edu.wpi.urs.wpiLibj.Galay
            com.604robotics.utils.Springable3 elay
```

All Implemented Interfaces:

```

IDevice6bDeviceController

```

```
public class SpringableRelay
extends Relay
```

0 xander oUa 3elay providing an easier control UI w. When an output is set Uo roe 3elay6oit aiersider Rs p ru n g RRothe.ReloadRoE method is called6oiUet
victor is sprung6oi o n s p rie g elay...UoUe 3elay is not sprung6oten the output is set to the default output. In this way6ote 3elay will only be moving when
you tell it to. Use this in a loop or something6and call ReloadRoE at the end. No more worries about code paths that don't ou at the 3elays1

Author:

X chael Smith

Nested Class Summary

Nested classes/interfaces inherited from class edu.wpi.first.wpilibj.Relay

Relay.Direction, Relay.InvalidValueException, Relay.Value

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Constructor Summary

Constructors

Constructor and Description

```
SpringableRelay(int moduleNumber, int channel, Relay.Direction direction, Relay.Value defaultDirection)
```

Initializes a new Springable3 elay.

```
SpringableRelay(int moduleNumber, int channel, Relay.Value defaultDirection)
```

Initializes a new Springable3 elay.

```
SpringableRelay(int channel, Relay.Direction direction, Relay.Value defaultDirection)
```

Initializes a new Springable3 elay.

```
SpringableRelay(int channel, Relay.Value defaultDirection)
```

Initializes a new Springable3 elay.

Method Summary

Methods

Modifier and Type	Method and Description
boolean	<code>getSprung()</code> Has the 3elay been sprung2
void	<code>reload()</code> If the 3elay has been sprung6ou n s p rin g oit ' still outputs to the default output.
void	<code>set(Relay.Value direction)</code> Sets the direction oUte 3elay.
void	<code>swing()</code>

void

`spring()`
Springs the 3 relay.

Methods inherited from class edu.wpi.first.wpilibj.Relay

`free`, `setDirection`

Methods inherited from class edu.wpi.first.wpilibj.SensorBase

`checkAnalogChannel`, `checkAnalogModule`, `checkDigitalChannel`, `checkDigitalModule`, `checkPWMChannel`, `checkPWMModule`, `checkRelayChannel`, `checkRelayModule`, `checkSolenoidChannel`, `checkSolenoidModule`, `getDefaultAnalogModule`, `getDefaultDigitalModule`, `getDefaultSolenoidModule`, `setDefaultAnalogModule`, `setDefaultDigitalModule`, `setDefaultSolenoidModule`

Methods inherited from class java.lang.Object

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

Constructor Detail

SpringableRelay

```
public SpringableRelay(int moduleNumber,
                       int channel,
                       Relay.Direction direction,
                       Relay.Value defaultDirection)
```

Initializes a new Springable3 relay.

Parameters:

- `moduleNumber` UoTe module slot the 3 relay is on.
- `channel` UoTe channel the 3 relay is on.
- `direction` UoTe direction the 3 relay should control.
- `defaultDirection` UoTe default direction Uo reloading.

SpringableRelay

```
public SpringableRelay(int channel,
                       Relay.Direction direction,
                       Relay.Value defaultDirection)
```

Initializes a new Springable3 relay.

Parameters:

- `channel` UoTe channel the 3 relay is on.
- `direction` UoTe direction the 3 relay should control.
- `defaultDirection` UoTe default direction Uo reloading.

SpringableRelay

```
public SpringableRelay(int moduleNumber,
                       int channel,
                       Relay.Value defaultDirection)
```

Initializes a new Springable3 relay.

Parameters:

- `moduleNumber` UoTe module slot the 3 relay is on.
- `channel` UoTe channel the 3 relay is on.
- `defaultDirection` UoTe default direction Uo reloading.

SpringableRelay

```
public SpringableRelay(int channel,
                       Relay.Value defaultDirection)
```

Initializes a new Springable3 relay.

Parameters:

Parameters:

channel UoTe channel the 3 elay is on.
dev aulBirection UoTe default direction Uo reloading.

Method Detail

getSprung

public boolean getSprung()

Has the 3 elay been sprung2

Returns:
z .ether or not the 3 elay has been sprung.

spring

public void spring()

Springs the 3 elay.

set

public void set(Relay.Value direction)

Sets the direction oUote 3 elay.

Overrides:
set in class Relay

Parameters:
direction UoTe direction to set.

reload

public void reload()

IUote 3 elay has been sprung6ou n s p rin g oit ' still on 166 to the default output.

com._604robotics.utils

Class CompensatingGyro

```
java.lang.Object
edu.wpi.first.wpilibj.SensorBase
edu.wpi.first.wpilibj.Gyro3fro
edu.wpi.first.wpilibj.Gyro3froxH
com._604robotics.utils.Compensating3fro
```

All Implemented Interfaces:

Device6Sensor6PIDSource

```
public class CompensatingGyro
extends GyroHax
```

3froxwith manual compensation0setting support.

Author:

Michael Smith

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Constructor Summary

Constructors

Constructor and Description

CompensatingGyro(AnalogChannel channel)

Initializes a new Compensating3froxspecified AnalogChannel.

CompensatingGyro(int port)

Initializes a new Compensating3froxspecified PRXportG

CompensatingGyro(int slot, int port)

Initializes a new Compensating3froxspecified PRXportspecified module port.

Method Summary

Methods

Modifier and Type

Method and Description

void

setAccumulatorCenter(int center)

Manually sets the center Ure accumulator.

Methods inherited from class edu.wpi.first.wpilibj.GyroHax

getAnalogChannel

Methods inherited from class edu.wpi.first.wpilibj.Gyro

free, getAngle, pidGet, reset, setSensitivity

Methods inherited from class edu.wpi.first.wpilibj.SensorBase

checkAnalogChannel, checkAnalogModule, checkDigitalChannel, checkDigitalModule, checkPWMChannel, checkPWMModule, checkRelayChannel, checkRelayModule, checkSolenoidChannel, checkSolenoidModule, getDefaultAnalogModule, getDefaultDigitalModule, getDefaultSolenoidModule, setDefaultAnalogModule, setDefaultDigitalModule, setDefaultSolenoidModule

Methods inherited from class java.lang.Object

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

Constructor Detail

CompensatingGyro

public CompensatingGyro(int port)

Initializes a new CompensatingGyro for the specified PORT. Note that port must be 0 or 1

Parameters:

port 0 The PORT the gyro is plugged into. Must be 0 or 1

CompensatingGyro

public CompensatingGyro(int slot, int port)

Initializes a new CompensatingGyro for the specified PORT and specified module port. Note that port must be 0 or 1

Parameters:

slot 0 The module slot the gyro is plugged into.

port 0 The PORT the gyro is plugged into. Must be 0 or 1

CompensatingGyro

public CompensatingGyro(AnalogChannel channel)

Initializes a new CompensatingGyro for the specified AnalogChannel. Note that port must be 0 or 1

Parameters:

channel 0 The AnalogChannel the gyro is plugged into.

Method Detail

setAccumulatorCenter

public void setAccumulatorCenter(int center)

Manually sets the center of the accumulator.

Parameters:

center 0 The center to set.

com._604robotics.utils

Class DeadbandedSource

java.lang.Object
com._604robotics.utils.DeadbandedSource

All Implemented Interfaces:

PIDSource

```
public class DeadbandedSource
extends Object
implements PIDSource
```

Implements a PIDSource wrapping around another PIDSource with a deadband range. It wraps within the deadbandUnit at the PIDController we're at where it wants to be.

Author:

Michael Smith

Constructor Summary

Constructors

Constructor and Description
<code>DeadbandedSource(PIDSource source)</code> Initializes a new DeadbandedSource.

Method Summary

Methods

Modifier and Type	Method and Description
double	<code>pidGet()</code> Hooks into PIDSource to get the value to send to the PIDController.
void	<code>setController(PIDController controller)</code> Sets the PIDController the source is hooked into.
void	<code>setDeadband(double lowerDeadband, double upperDeadband)</code> Sets the range of the deadband.

Methods inherited from class java.lang.Object

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

Constructor Detail

DeadbandedSource

```
public DeadbandedSource(PIDSource source)
```

Initializes a new DeadbandedSource.

Parameters:

- source: The underlying PIDSource to wrap around.

Method Detail

setController

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

Sets the PIDController the source is)ed into.

Parameters:

controller 0oTe PIDController the source is)ed into.

setDeadband

public Coid setDeadband(double lowerDeadband,
double upperDeadband)

Sets the range) o rø deadband.

Parameters:

lowerDeadband 0oTe lower bound o) ote deadband.

upperDeadband 0oTe upper bound o) ote deadband.

pidGet

public double pidDet()

Hooks into PIDSource 0ogets the value to send to the PIDController. With a deadbandR

Specified by:

pidDetoin oierace PIDSource

Returns:

The value to send to the PIDController.

com._604robotics.utils

Class UpDownPIDController

java.lang.Object
edu.wpi.first.wpilibj.PIDController
com._604robotics.utils.) p DownPIDController

All Implemented Interfaces:

IDevice, I) til it f

public class UpDownPIDController
extends PIDController

A PIDController with different gains for up and down.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static class	UpDownPIDController.Gains A structure containing the P, I, and D gains.

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.PIDController

kDefaultPeriod

Constructor Summary

Constructors

Constructor and Description
UpDownPIDController (UpDownPIDController.Gains upGains, UpDownPIDController.Gains downGains, PIDSource source, PIDOutput output) Initializes a new) p DownPIDController.

Method Summary

Methods

Modifier and Type	Method and Description
UpDownPIDController.Gains	getDownGains () Gets the D gains for going in ymd o
UpDownPIDController.Gains	getUpGains () Gets the D gains for going in g ou p G
void	refreshGains ()) p dtes the gains for curren direction.
void	setDownGains (UpDownPIDController.Gains downGains) Sets the gains for going in ymd o
void	setSetpoint (double setpoint) Sets the setpoint to go to.
void	setUpGains (UpDownPIDController.Gains upGains) Sets the gains for going in g ou p G

Methods inherited from class edu.wpi.first.wpilibj.PIDController

Methods inherited from class `edu.wpi.first.wpilibj.PIDController`:

`disable`, `enable`, `free`, `get`, `getD`, `getError`, `getI`, `getP`, `getSetpoint`, `isEnabled`, `onTarget`, `reset`, `setContinuous`, `setContinuous`, `setInputRange`, `setOutputRange`, `setPID`, `setTolerance`

Methods inherited from class `java.lang.Object`

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

Constructor Detail

UpDownPIDController

```
public UpDownPIDController(UpDownPIDController.Gains upGains,
                           UpDownPIDController.Gains downGains,
                           PIDSource source,
                           PIDOutput output)
```

Initializes a new UpDownPIDController.

Parameters:

- `upGains` zero gains to use when going up.
- `downGains` zero gains to use when going down.
- `source` zero PIDSource to plug in.
- `output` zero PIDOutput to plug in.

Method Detail

getUpGains

```
public UpDownPIDController.Gains getUpGains()
```

Gets the gains to log on going up

Returns:

The gains to log on going up

getDownGains

```
public UpDownPIDController.Gains getDownGains()
```

Gets the gains to log on going down

Returns:

The gains to log on going down

refreshGains

```
public void refreshGains()
```

Updates the gains to reflect current direction.

setUpGains

```
public void setUpGains(UpDownPIDController.Gains upGains)
```

Sets the gains to log on going up

Parameters:

`upGains` zero gains to use when going up.

setDownGains

```
public void setDownGains(UpDownPIDController.Gains downGains)
```

Sets the gains to log on going down

Set the gains to use when going down.

Parameters:

doCnGains z oTe gains to use when going down.

setSetpoint

public f oid setSetpoint(double setpoint)

Sets the setpoint to go to.

Overrides:

setSetpoint in class PIDController

Parameters:

setpoint z oTe setpoint to go to.

com._604robotics.utils

Class DualVictor

java.lang.Object
com._604robotics.utils.DualU ictor

All Implemented Interfaces:

PIDOutput

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

Control two U ictors like they're one. Useful for PID controllers. Also, it's compatible with the SpringableU ictorRG

Constructor Summary

Constructors
Constructor and Description
DualVictor(int leftPort, int rightPort) Instantiates a DualU ictor with a left and a right PortX output
DualVictor(int leftSlot, int leftPort, int rightSlot, int rightPort) Instantiates a DualU ictor with left and right slot and PortX output
DualVictor(Victor leftVictor, Victor rightVictor) Instantiates a DualU ictor with left and right slot and PortX output

Method Summary

Methods	
Modifier and Type	Method and Description
double	<code>get()</code> Checks the current power the U ctors are set to.
boolean	<code>getSprung()</code> Has the victor been sprung1
void	<code>pidWrite</code> (double output) u ction to hook into the PIDController.
void	<code>reload()</code> If the U ctor has been sprung, unspring itXoi6on @tth@output to 0.
void	<code>set</code> (double speed) Sets the power of the U ctors.
void	<code>setController</code> (PIDController controller) Sets the PIDController for this DualU ctor, if there is one.
void	<code>setDeadband</code> (double lowerDeadband, double upperDeadband) Sets the deadband for the DualU ctor.
void	<code>setLeftInversion</code> (boolean inversion) Sets the inversion for the 2eft2oldtor.
void	<code>setRightInversion</code> (boolean inversion) Sets the inversion for the 2rig . t 2oldtor.
void	<code>setSafetyEnabled</code> (boolean enabled) Sets whether or not safety is enabled.
void	<code>spring()</code> 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 Port X or Y

Parameters:

leftPort' of Port X or Y

rightPort' of Port X or Y

DualVictor

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

Initializes a DualVictor with left and right slot and Port X or Y

Parameters:

leftSlot' of slot the Victor is plugged into.

leftPort' of Port X or Y

rightSlot' of slot the Victor is plugged into.

rightPort' of Port X or Y

DualVictor

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

Initializes a DualVictor with left and right slot and Port X or Y

Parameters:

leftVictor' of Victor

rightVictor' of 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 Victor.

Parameters:

inversion' of slot

setRightInversion

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

Sets the inversion for the 2rig . t 200.i

Parameters:

inAersion ' ol s oivoted1

get

public double get()

C .ecks the current power the U ctors are set to.

Returns:

The current power the U ctors are set to.

set

public void set(double speed)

Sets the power of the U ctors.

Parameters:

speed ' oTe speed to set.

pidWrite

public void pidWrite(double output)

Function to hook into the PIDController. Sets the power of the U ctors.

Specified by:

pidWrite in interface PIDOutput

Parameters:

output ' oTe speed to set.

setDeadband

public void setDeadband(double lowerDeadband,
double upperDeadband)

Sets the deadband for the DualU ctor. The default is no deadband.

Parameters:

lowerDeadband ' oTe lower bound of the deadband.

upperDeadband ' oTe upper bound of the deadband.

setSafetyEnabled

public void setSafetyEnabled(boolean enabled)

Sets whether or not safety is enabled.

Parameters:

enabled ' oj ether or not safety is enabled.

reload

public void reload()

If the U ctor has been sprung, unspring itXoi6on @tttesoutput to 0.

setController

public void setController(PIDController controller)

Sets the PIDController for this DualU ctor, if there is one. If the PIDController is enabled, reload will assume it) s ou pidg it, and won) t eset the output to 0.

Parameters:

controller' oTe.PIDController for this DualU ctor.

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

com._604robotics.utils

Class SpringableVictor

java.lang.Object
edu.wpi.first.wpilibj.SensorBase
edu.wpi.first.wpilibj.P3X
edu.wpi.first.wpilibj.CANP3X
edu.wpi.first.wpilibj.CANVictor
com._604robotics.utils.Springable6Victor

All Implemented Interfaces:

MotorSafety, IDevice, IDeviceController, PIDOutput, SpeedController

```
public class SpringableVictor
extends Victor
```

Under a 6Victor providing an easier control UI v.3. When an output is set to a 6Victor, it is considered sprung. When the ReloadRole method is called, if the victor is sprung, it unsprings the 6Victor. If the 6Victor is not sprung, then the output is set to zero. In this way, the 6Victor will only be moving when you tell it to. Just put this in a loop or something, and call ReloadRole at the end. No more worries about code paths that don't output to the 6Victors.

Author:

Michael Smith

Nested Class Summary

Nested classes/interfaces inherited from class edu.wpi.first.wpilibj.PWM

PWM.PeriodMultiplier

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.PWM

kDefaultMinPwmHigh, kDefaultPwmPeriod, kPwmDisabled

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Fields inherited from interface edu.wpi.first.wpilibj.MotorSafety

DEFAULT_SAFETY_EXPIRATION

Constructor Summary

Constructors

Constructor and Description

SpringableVictor(int port)
Initializes a new Springable6Victor on the given P3X output G

SpringableVictor(int slot, int port)
Initializes a new Springable6Victor on the given module slot and P3X output G

Method Summary

Methods

Modifier and Type

Method and Description

boolean	getSprung() Has the victor been sprung2
void	pidWrite (double output) Addition to hook into the PIDController.
void	reload() If the 6-ctric has been sprung, unspring it' oioUn @ttides output to 0.
void	set (double speed) Sets the power oUote 6-ctric.
void	setController (PIDController controller) Sets the PIDController Uo rot . isto6 iUote is one.
void	spring() Springs the victor.

Methods inherited from class edu.wpi.first.wpilibj.Victor

get, set

Methods inherited from class edu.wpi.first.wpilibj.SafePWM

disable, Feed, getDescription, getExpiration, isAlive, isSafetyEnabled, setExpiration, setSafetyEnabled, stopMotor

Methods inherited from class edu.wpi.first.wpilibj.PWM

enableDeadbandElimination, free, getChannel, getModuleNumber, getPosition, getRaw, getSpeed, setBounds, setPeriodMultiplier, setPosition, setRaw

Methods inherited from class edu.wpi.first.wpilibj.SensorBase

checkAnalogChannel, checkAnalogModule, checkDigitalChannel, checkDigitalModule, checkPWMChannel, checkPWMModule, checkRelayChannel, checkRelayModule, checkSolenoidChannel, checkSolenoidModule, getDefaultAnalogModule, getDefaultDigitalModule, getDefaultSolenoidModule, setDefaultAnalogModule, setDefaultDigitalModule, setDefaultSolenoidModule

Methods inherited from class java.lang.Object

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

Methods inherited from interface edu.wpi.first.wpilibj.SpeedController

disable

Constructor Detail

SpringableVictor

public SpringableVictor(int port)

Initializes a new Springable6-ctric on the given P3-X op o rt G

Parameters:

port oUote P3-X op o rt 6-ctric is connected to.

SpringableVictor

public SpringableVictor(int slot, int port)

Initializes a new Springable6-ctric on the given module slot and P3-X op o rt G

Parameters:

slot oUote module slot the 6-ctric is connected to.

port Uote P3-X op o rt 6-ctric is connected to.

Method Detail

getSprung

getSprung

```
public boolean getSprung()
```

Has the victor been sprung2

Returns:

3 .either or not the victor has been sprung.

spring

```
public void spring()
```

Springs the victor.

set

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

Sets the power of UoTe 6 ctor.

Specified by:

set in interface SpeedController

Overrides:

set in class Victor

Parameters:

speed UoTe speed to set.

pidWrite

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

Function to hook into the PIDController. Sets the power of UoTe 6 ctors.

Specified by:

pidWrite in interface PIDOutput

Overrides:

pidWrite in class Victor

Parameters:

output UoTe speed to set.

reload

```
public void reload()
```

UoTe 6 ctor has been sprung, unspring it' oiUon @ttteoutput to 0.

setController

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

Sets the PIDController Uo rot . isto6 iUotere is one. IUote PIDController is enabled, reload will assume it's ou ptidg it, and won't eset the output to 0.

Parameters:

controller UoTe PIDController Uo rot . isto6 i

com._604robotics.utils

Class ConvertingPIDController

java.lang.Object
 edu.wpi.first.wpilibj.PIDController
 com._604robotics.utils.ConvertingPIDController

All Implemented Interfaces:

IDeviceCallback

```
public class ConvertingPIDController
extends PIDController
```

A **nextender** of **WPILib** PIDController that converts between units when getting and setting a setpoint.

Author:

Michael Smith

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.PIDController

kDefaultPeriod

Constructor Summary

Constructors

Constructor and Description
ConvertingPIDController (double Kp, double Ki, double Kd, PIDSource source, PIDOutput output) Allocate a PID object with the given constants Kp , Ki , Kd and source .
ConvertingPIDController (double Kp, double Ki, double Kd, PIDSource source, PIDOutput output, double period) Allocate a PID object with the given constants Kp , Ki , Kd and period .

Method Summary

Methods

Modifier and Type	Method and Description
double	getRealSetpoint () Returns the real setpoint of this PIDController.
double	getSetpoint () Returns the current setpoint of this PIDController.
void	setConversionFactor (double conversionFactor) Sets the factor to use when doing conversion on setSetpoint and getSetpoint .
void	setRealSetpoint (double setpoint) Sets the real setpoint of this PIDController.
void	setSetpoint (double setpoint) Set the setpoint of this PIDController.

Methods inherited from class edu.wpi.first.wpilibj.PIDController

disable, enable, free, get, getD, getError, getI, getP, isEnabled, onTarget, reset, setContinuous, setContinuous, setInputRange, setOutputRange, setPID, setTolerance

Methods inherited from class java.lang.Vbjeet

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

Constructor Detail

ConvertingPIDController

```
public ConvertingPIDController(double Kp,
                               double Ki,
                               double Kd,
                               PIDSource source,
                               PIDOutput output)
```

Allocate a PID object with the given constants K_p of K_i of K_d on a 6 Hz sample.

Parameters:

- K_p **Proportional coefficient**
- K_i **Integral coefficient**
- K_d **Derivative coefficient**
- `source` **The PIDSource object that is used to get values**
- `output` **The PIDOutput object that is set to the output value**

ConvertingPIDController

```
public ConvertingPIDController(double Kp,
                               double Ki,
                               double Kd,
                               PIDSource source,
                               PIDOutput output,
                               double period)
```

Allocate a PID object with the given constants K_p of K_i of K_d

Parameters:

- K_p **Proportional coefficient**
- K_i **Integral coefficient**
- K_d **Derivative coefficient**
- `source` **The PIDSource object that is used to get values**
- `output` **The PIDOutput object that is set to the output value**
- `period` **The loop time in seconds. This particularly effects calculations of the integral and differential terms. The default is 6 S F s G**

Method Detail

getRealSetpoint

```
public double getRealSetpoint()
```

Gets the real setpoint of the PIDController.

Returns:

- The real setpoint of the PIDController.

getSetpoint

```
public double getSetpoint()
```

Description copied from class: edu.wpi.first.wpilibj.PIDController
Returns the current setpoint of the PIDController

Overrides:

- `getSetpoint` in class `PIDController`

Returns:

- the current setpoint

setRealSetpoint

```
public void setRealSetpoint(double setpoint)
```

Set the real setpoint of the PIDController

Sets the setpoint of the PIDController.

Parameters:

setpoint `double` The setpoint to set.

setSetpoint

public void setSetpoint(double setpoint)

Description copied from class: `edu.wpi.first.wpilibj.PIDController`
Set the setpoint of the PIDController

Overrides:

setSetpoint in class `PIDController`

Parameters:

setpoint `double` The desired setpoint

setConversionFactor

public void setConversionFactor(double conversionFactor)

Sets the factor to use when doing conversion on setSetpoint and getSetpoint.

Parameters:

conversionFactor `double` The conversion factor to use.

com._604robotics.utils

Class LinearController

java.lang.Object
com._604robotics.utils.U iearController

```
public class LinearController
extends Object
```

This class implements a controller with a hori o ntsegment, a linear segment, and finally a coasting segment. When a target point is set, the controller decides which direction to go to get there, and then focuses on getting to that point or past it in that direction. If that condition is met, the output drops to) ero. z l s, if weRe within a certain Woasting rangeWBo output will be floored at the Woasting outputWGoze| if weRe outside a certain W o ri) alrangeWBo output will be ceilinged at a certain W o ri) aloutputWGoze| the output will be scaled linearly between the two outputs.

Author:

Michael Smith

Constructor Summary

Constructors
Constructor and Description
LinearController (PIDSource source, PIDOutput output, double horizontalRange, double horizontalOutput, double coastingRange, double coastingOutput) Initiali) es a new U iearController.

Method Summary

Methods	
Modifier and Type	Method and Description
double	<code>calculate()</code> Function that performs the output calculation.
double	<code>getTarget()</code> j ets the current target.
boolean	<code>onTarget()</code> Are we there yet1
Coid	<code>setCoastingRange</code> (double coastingRange, double coastingOutput) X p dtes the coasting values.
Coid	<code>setH oiz ntalRange</code> (double horizontalRange, double horizontalOutput) X p dtes the hori o ntalvalues.
Coid	<code>setTarget</code> (double target) Sets the current target.
Coid	<code>update()</code> X p dtes the PIDOutput based on the latest data.

Methods inherited from class java.lang.Object

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

Constructor Detail

LinearController
<pre>public LinearController(H I Dource source, H I Doutput output, double horizontalRange, double horizontalOutput, double coastingRange, double coastingOutput) Initiali) es a new U iearController.</pre>
Parameters:

source `20APID` Source to read from.

output `20APID` Output to write to.

horizontalRange `20Te.hori`) `o raitrange`, as defined in the class description.

horizontalOutput `20Te.hori`) `o raitoutput`, as defined in the class description.

coastingRange `20Te` coasting range, as defined in the class description.

coastingOutput `20Te` coasting output, as defined in the class description.

Method Detail

setHorizontalRange

```
public Coid setHorizontalRange(double horizontalRange,
                               double horizontalOutput)
```

X p dtes the hori) o raitvalues.

Parameters:

- horizontalRange `20Te.hori`) `o raitrange`, as defined in the class description.
- horizontalOutput `20Te.hori`) `o raitoutput`, as defined in the class description.

setCoastingRange

```
public Coid setCoastingRange(double coastingRange,
                              double coastingOutput)
```

X p dtes the coasting values.

Parameters:

- coastingRange `20Te` coasting range, as defined in the class description.
- coastingOutput `20Te` coasting output, as defined in the class description.

getTarget

```
public double getTarget()
```

j ets the current target.

Returns:

- The current target.

setTarget

```
public Coid setTarget(double target)
```

Sets the current target.

Parameters:

- target `20Te` target to move toward.

onTarget

```
public boolean onTarget()
```

Are we there yet?

Returns:

- `0` .ether or not weRe there yet.

calculate

```
public double calculate()
```

Function that performs the output calculation. z x p eds for debug use, mainly.

... eturns:

whereas:

An output value, to be passed to a PIDOutput.

update

```
public void update()
```

Updates the PIDOutput based on the latest data.

[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](#)

com._604robotics.utils

Interface XboxController.Axis

Enclosing class:

[j V o x C o n t r o l l](#)

```
public static interface XboxController.Axis
```

Enumeration) o r a available axes on the j V o o n t r o l l e r .

Field Summary

Fields

Modifier and Type	Field and Description
static int	LEFT_STICK_X
static int	LEFT_STICK_Y
static int	RIGHT_STICK_X
static int	RIGHT_STICK_Y

Field Detail

LEFT_STICK_X

```
static final int LEFT_STICK_X
```

See Also:

[Constant Field 3alues](#)

LEFT_STICK_Y

```
static final int LEFT_STICK_Y
```

See Also:

[Constant Field 3alues](#)

RIGHT_STICK_X

```
static final int RIGHT_STICK_X
```

See Also:

[Constant Field 3alues](#)

RIGHT_STICK_Y

```
static final int RIGHT_STICK_Y
```

See Also:

[Constant Field 3alues](#)

com._604robotics.utils

Class EncoderPIDSource

java.lang.Object
edu.wpi.first.wpilibj.Sensor ase
edu.wpi.first.wpilibj.Counter
com._604robotics.utils.EncoderOffset
com._604robotics.utils.EncoderPIDSource

All Implemented Interfaces:

Counter) ase6bDevice6bEncoder6bIDSource

```
public class EncoderPIDSource
extends EncoderOffset
```

Encoder extender that return the value of Counter.get0 when pidRet is called. DropWn placement: all constructors of Encoder class are implemented here.

Author:

Michael Smith

Nested Class Summary

Nested classes/interfaces inherited from class edu.wpi.first.wpilibj.Encoder

Encoder.PIDSourceParameter

Nested classes/interfaces inherited from interface edu.wpi.first.wpilibj.CounterBase

CounterBase.EncodingType

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.Encoder

m_aSource, m_bSource, m_indexSource

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Constructor Summary

Constructors

Constructor and Description

EncoderPIDSource(DigitalSource aSource, DigitalSource bSource)
Encoder constructor.

EncoderPIDSource(DigitalSource aSource, DigitalSource bSource, boolean reverseDirection)
Encoder constructor.

EncoderPIDSource(DigitalSource aSource, DigitalSource bSource, boolean reverseDirection, CounterBase.EncodingType encodingType)
Encoder constructor.

EncoderPIDSource(DigitalSource aSource, DigitalSource bSource, DigitalSource indexSource)
Encoder constructor.

EncoderPIDSource(DigitalSource aSource, DigitalSource bSource, DigitalSource indexSource, boolean reverseDirection)
Encoder constructor.

EncoderPIDSource(int aChannel, int bChannel)
Encoder constructor.

EncoderPIDSource(int aChannel, int bChannel, boolean reverseDirection)

Encoder constructor.

```
EncoderPIDSource(int aChannel, int bChannel, boolean reverseDirection, CounterBase.EncodingType encodingType)
```

Encoder constructor.

```
EncoderPIDSource(int aChannel, int bChannel, int indexChannel)
```

Encoder constructor.

```
EncoderPIDSource(int aChannel, int bChannel, int indexChannel, boolean reverseDirection)
```

Encoder constructor.

```
EncoderPIDSource(int aSlot, int aChannel, int bSlot, int bChannel)
```

Encoder constructor.

```
EncoderPIDSource(int aSlot, int aChannel, int bSlot, int bChannel, boolean reverseDirection)
```

Encoder constructor.

```
EncoderPIDSource(int aSlot, int aChannel, int bSlot, int bChannel, boolean reverseDirection, CounterBase.EncodingType encodingType)
```

Encoder constructor.

```
EncoderPIDSource(int aSlot, int aChannel, int bSlot, int bChannel, int indexSlot, int indexChannel)
```

Encoder constructor.

```
EncoderPIDSource(int aSlot, int aChannel, int bSlot, int bChannel, int indexSlot, int indexChannel, boolean reverseDirection)
```

Encoder constructor.

Method Summary

Methods

Modifier and Type	Method and Description
double	<code>pidGet()</code> Hooks into the PIDSource interface.

Methods inherited from class com._604robotics.utils.EncoderOffset

`getRaw`, `reset`, `setOffset`

Methods inherited from class edu.wpi.first.wpilibj.Encoder

`free`, `get`, `getDirection`, `getDistance`, `getPeriod`, `getRate`, `getStopped`, `setDistancePerPulse`, `setMaxPeriod`, `setMinRate`, `setPIDSourceParameter`, `setReverseDirection`, `start`, `stop`

Methods inherited from class edu.wpi.first.wpilibj.SensorBase

`checkAnalogChannel`, `checkAnalogModule`, `checkDigitalChannel`, `checkDigitalModule`, `checkPWMChannel`, `checkPWMModule`, `checkRelayChannel`, `checkRelayModule`, `checkSolenoidChannel`, `checkSolenoidModule`, `getDefaultAnalogModule`, `getDefaultDigitalModule`, `getDefaultSolenoidModule`, `setDefaultAnalogModule`, `setDefaultDigitalModule`, `setDefaultSolenoidModule`

Methods inherited from class java.lang.Object

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

Constructor Detail

EncoderPIDSource

```
public EncoderPIDSource(int aSlot,
                        int aChannel,
                        int bSlot,
                        int bChannel,
                        boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b modules and channels. Upward is forward.

Parameters:

- `aSlot` **What** a channel digital input module.
- `aChannel` **What** a channel digital input channel.
- `bSlot` **What** b channel digital input module.
- `bChannel` **What** b channel digital input channel.
- `reverseDirection` **What** represents the orientation of the encoder and inverts the output values if necessary so that upward represents positive values.

EncoderPIDSource

```
public EncoderPIDSource(int aSlot,
                        int aChannel,
                        int bSlot,
                        int bChannel)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd.

Parameters:

- aSlot VbTe a channel digital input module.
- aChannel VbTe a channel digital input channel.
- bSlot VbTe b channel digital input module.
- bChannel VbTe b channel digital input channel.

EncoderPIDSource

```
public EncoderPIDSource(int aSlot,
                        int aChannel,
                        int bSlot,
                        int bChannel,
                        boolean reverseDirection,
                        CounterBase.EncodingType encodingType)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd.

Parameters:

- aSlot VbTe a channel digital input module.
- aChannel VbTe a channel digital input channel.
- bSlot VbTe b channel digital input module.
- bChannel VbTe b channel digital input channel.
- reverseDirection Vb represents the orientation oUote encoder and inverts the output values iU necessary so Uoward represents positive values.
- encodingType Vb either kj 1 6x1 6x4 root o oate j 1 6xX 1 oo encoding IUo, 1 oesed6oten an encoder FPR A ooeVtis used and the returned counts will be 4x the encoder spec2dvalue since all rising and falling edges are counted. IUoj 1 oo 4x selected then a counter object will be used and the returned value will either exactly match the spec2dcount or be double 0X x as spec2dcount.

EncoderPIDSource

```
public EncoderPIDSource(int aSlot,
                        int aChannel,
                        int bSlot,
                        int bChannel,
                        int indexSlot,
                        int indexChannel,
                        boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd.' s in gIndex pulse Uoes 4x encoding.

Parameters:

- aSlot VbTe a channel digital input module.
- aChannel VbTe a channel digital input channel.
- bSlot VbTe b channel digital input module.
- bChannel VbTe b channel digital input channel.
- indexSlot VbTe index channel digital input module.
- indexChannel VbTe index channel digital input channel.
- reverseDirection Vb represents the orientation oUote encoder and inverts the output values iU necessary so Uoward represents positive values.

EncoderPIDSource

```
public EncoderPIDSource(int aSlot,
                        int aChannel,
                        int bSlot,
                        int bChannel,
                        int indexSlot,
                        int indexChannel)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd.' s in gIndex pulse Uoes 4x encoding.

Parameters:

- aSlot VbTe a channel digital input module.

aSlot **WbTe** a channel digital input module.

aChannel **WbTe** a channel digital input channel.

bSlot **WbTe** b channel digital input module.

bChannel **WbTe** b channel digital input channel.

indexSlot **WbTe** index channel digital input module.

indexChannel **WbTe** index channel digital input channel.

EncoderPIDSource

```
public EncoderPIDSource(int aChannel,
                        int bChannel,
                        boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module.

Parameters:

aChannel **WbTe** a channel digital input channel.

bChannel **WbTe** b channel digital input channel.

reverseDirection **WbTe** represents the orientation of the encoder and inverts the output values if necessary so Upward represents positive values.

EncoderPIDSource

```
public EncoderPIDSource(int aChannel,
                        int bChannel)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module.

Parameters:

aChannel **WbTe** a channel digital input channel.

bChannel **WbTe** b channel digital input channel.

EncoderPIDSource

```
public EncoderPIDSource(int aChannel,
                        int bChannel,
                        boolean reverseDirection,
                        CounterBase.EncodingType encodingType)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module.

Parameters:

aChannel **WbTe** a channel digital input channel.

bChannel **WbTe** b channel digital input channel.

reverseDirection **WbTe** represents the orientation of the encoder and inverts the output values if necessary so Upward represents positive values.

encodingType **WbTe** Either kj 1 16X 1 60X 1 00 encoding. If 1 is selected then an encoder FPR A is used and the returned counts will be 4x the encoder spec2dvalue since all rising and falling edges are counted. If 1 00 is selected then a counter object will be used and the returned value will either exactly match the spec2dcount or be double 0X x 20spec2dcount.

EncoderPIDSource

```
public EncoderPIDSource(int aChannel,
                        int bChannel,
                        int indexChannel,
                        boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module. ' s in an index pulse Uses 4x encoding

Parameters:

aChannel **WbTe** a channel digital input channel.

bChannel **WbTe** b channel digital input channel.

indexChannel **WbTe** index channel digital input channel.

reverseDirection **WbTe** represents the orientation of the encoder and inverts the output values if necessary so Upward represents positive values.

EncoderPIDSource

```
public EncoderPIDSource(int aChannel,
```

```
int bChannel,  
int indexChannel)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module. 's in an index pulse Uses 4x encoding

Parameters:

- aChannel **WbTe** a channel digital input channel.
- bChannel **WbTe** b channel digital input channel.
- indexChannel **WbTe** index channel digital input channel.

EncoderPIDSource

```
public EncoderPIDSource(DigitalSource aSource,  
                        DigitalSource bSource,  
                        boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **WbTe** source that should be used Uo roe a channel.
- bSource **WbTe** source that should be used Uo roe b channel.
- reverseDirection **WbTe** represents the orientation oUote encoder and inverts the output values iUnecessary so Uoward represents positive values.

EncoderPIDSource

```
public EncoderPIDSource(DigitalSource aSource,  
                        DigitalSource bSource)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **WbTe** source that should be used Uo roe a channel.
- bSource **WbTe** source that should be used Uo roe b channel.

EncoderPIDSource

```
public EncoderPIDSource(DigitalSource aSource,  
                        DigitalSource bSource,  
                        boolean reverseDirection,  
                        CounterBase.EncodingType encodingType)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **WbTe** source that should be used Uo roe a channel.
- bSource **WbTe** source that should be used Uo roe b channel.
- reverseDirection **WbTe** represents the orientation oUote encoder and inverts the output values iUnecessary so Uoward represents positive values.
- encodingType **WbTe** Whether kj 1 601 600410 ot o cote j 1 60X 1 oo encoding iUo, 1 oteed 60ten an encoder FPR A oetjjs used and the returned counts will be 4x the encoder spec2dvalue since all rising and falling edges are counted. iUoj 1 oo anX selected then a counter object will be used and the returned value will either exactly match the spec2dcount or be double 0X x aspec2dcount.

EncoderPIDSource

```
public EncoderPIDSource(DigitalSource aSource,  
                        DigitalSource bSource,  
                        DigitalSource indexSource,  
                        boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **WbTe** source that should be used Uo roe a channel.
- bSource **WbTe** source that should be used Uo roe b channel.
- indexSource **WbTe** source that should be used Uo roe index channel.
- reverseDirection **WbTe** represents the orientation oUote encoder and inverts the output values iUnecessary so Uoward represents positive values.

EncoderPIDSource

```
public EncoderPIDSource(DigitalSource aSource,
                        DigitalSource bSource,
                        DigitalSource indexSource)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **W**ote source that should be used Uo **ro** a channel.
- bSource **W**ote source that should be used Uo **ro** b channel.
- indexSource **W**ote source that should be used Uo **ro** index channel.

Method Detail

pidGet

```
public double pidGet()
```

Hooks into the PIDSource interface. This method overrides the one implemented by the underlying Encoder class6os iF pettr6oing the value oU t. is 60zU

Specified by:

pidGet in interface PIDSource

Overrides:

pidGet in class Encoder

Returns:

The value to pass back to the PIDSourceUoin ot caise60at oUot . is 60zU

com._604robotics.utils

Class EncoderOffset

java.lang.Object
edu.wpi.first.wpilibj.CounterBase
edu.wpi.first.wpilibj.Encoder
com._604robotics.utils.EncoderOffset

All Implemented Interfaces:

CounterBase, Device, EncoderBase, PIDSource

Direct Known Subclasses:

EncoderPIDSource

```
public class EncoderOffset
extends Encoder
```

Encoder extender that return the value of Encoder.get0 when pidRet is called. DropWin replacement: all constructors from Encoder class are implemented here.

Author:

Michael Smith

Nested Class Summary

Nested classes/interfaces inherited from class edu.wpi.first.wpilibj.Encoder

Encoder.PIDSourceParameter

Nested classes/interfaces inherited from interface edu.wpi.first.wpilibj.CounterBase

CounterBase.EncodingType

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.Encoder

m_aSource, m_bSource, m_indexSource

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Constructor Summary

Constructors

Constructor and Description

EncoderOffset(DigitalSource aSource, DigitalSource bSource)
Encoder constructor.

EncoderOffset(DigitalSource aSource, DigitalSource bSource, boolean reverseDirection)
Encoder constructor.

EncoderOffset(DigitalSource aSource, DigitalSource bSource, boolean reverseDirection, CounterBase.EncodingType encodingType)
Encoder constructor.

EncoderOffset(DigitalSource aSource, DigitalSource bSource, DigitalSource indexSource)
Encoder constructor.

EncoderOffset(DigitalSource aSource, DigitalSource bSource, DigitalSource indexSource, boolean reverseDirection)
Encoder constructor.

EncoderOffset(int aChannel, int bChannel)

Encoder constructor.

EncoderOffset(in aChannel, int bChannel, boolean reverseDirection)

Encoder constructor.

EncoderOffset(int aChannel, int bChannel, boolean reverseDirection, **CounterBase.EncodingType** encodingType)

Encoder constructor.

EncoderOffset(int aChannel, int bChannel, int indexChannel)

Encoder constructor.

EncoderOffset(int aChannel, int bChannel, int indexChannel, boolean reverseDirection)

Encoder constructor.

EncoderOffset(int aSlot, int aChannel, int bSlot, int bChannel)

Encoder constructor.

EncoderOffset(int aSlot, int aChannel, int bSlot, int bChannel, boolean reverseDirection)

Encoder constructor.

EncoderOffset(int aSlot, int aChannel, int bSlot, int bChannel, boolean reverseDirection, **CounterBase.EncodingType** encodingType)

Encoder constructor.

EncoderOffset(int aSlot, int aChannel, int bSlot, int bChannel, int indexSlot, int indexChannel)

Encoder constructor.

EncoderOffset(int aSlot, int aChannel, int bSlot, int bChannel, int indexSlot, int indexChannel, boolean reverseDirection)

Encoder constructor.

Method Summary

Methods

Modifier and Type	Method and Description
int	getRaw () Returns the raw value from the encoder.
void	reset () Resets the Encoder.
void	setOffset (int offset) Sets the offset value for the Encoder.

Methods inherited from class edu.wpi.first.wpilibj.Encoder

free, get, getDirection, getDistance, getPeriod, getRate, getStopped, pidGet, setDistancePerPulse, setMaxPeriod, setMinRate, setPIDSourceParameter, setReverseDirection, start, stop

Methods inherited from class edu.wpi.first.wpilibj.SensorBase

checkAnalogChannel, checkAnalogModule, checkDigitalChannel, checkDigitalModule, checkPWMChannel, checkPWMModule, checkRelayChannel, checkRelayModule, checkSolenoidChannel, checkSolenoidModule, getDefaultAnalogModule, getDefaultDigitalModule, getDefaultSolenoidModule, setDefaultAnalogModule, setDefaultDigitalModule, setDefaultSolenoidModule

Methods inherited from class java.lang.Object

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

Constructor Detail

EncoderOffset

```
public EncoderOffset(int aSlot,
                    int aChannel,
                    int bSlot,
                    int bChannel,
                    boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b modules and channels. If reverseDirection is true, the encoder is assumed to be reversed.

Parameters:

- aSlot: The a channel digital input module.
- aChannel: The a channel digital input channel.
- bSlot: The b channel digital input module.
- bChannel: The b channel digital input channel.
- reverseDirection: Represents the orientation of the encoder and inverts the output values if necessary so that forward represents positive values.

EncoderOffset

```
public EncoderOffset(int aSlot,
                    int aChannel,
                    int bSlot,
                    int bChannel)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd.

Parameters:

- aSlot **WbTe** a channel digital input module.
- aChannel **WbTe** a channel digital input channel.
- bSlot **WbTe** b channel digital input module.
- bChannel **WbTe** b channel digital input channel.

EncoderOffset

```
public EncoderOffset(int aSlot,
                    int aChannel,
                    int bSlot,
                    int bChannel,
                    boolean reverseDirection,
                    CounterBase.EncodingType encodingType)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd.

Parameters:

- aSlot **WbTe** a channel digital input module.
- aChannel **WbTe** a channel digital input channel.
- bSlot **WbTe** b channel digital input module.
- bChannel **WbTe** b channel digital input channel.
- reverseDirection **Wb** represents the orientation oUote encoder and inverts the output values iUnecessary so Uoward represents positive values.
- encodingType **Wb** either k1 X 162 X 6004X ot o cimate if X 6o2 X oo encoding IUo, X osted6oten an encoder FPR A ooevtijs used and the returned counts will be 4x the encoder spec' dvalue since all rising and lalling edges are counted. IUo1 X oo are 2 xected then a counter object will be used and the returned value will either exactly match the spec' doount or be double 02 x 2o spec' doount.

EncoderOffset

```
public EncoderOffset(int aSlot,
                    int aChannel,
                    int bSlot,
                    int bChannel,
                    int indexSlot,
                    int indexChannel,
                    boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd. U s in g oIndex pulse Uoæes 4x encoding.

Parameters:

- aSlot **WbTe** a channel digital input module.
- aChannel **WbTe** a channel digital input channel.
- bSlot **WbTe** b channel digital input module.
- bChannel **WbTe** b channel digital input channel.
- indexSlot **WbTe** index channel digital input module.
- indexChannel **WbTe** index channel digital input channel.
- reverseDirection **Wb** represents the orientation oUote encoder and inverts the output values iUnecessary so Uoward represents positive values.

EncoderOffset

```
public EncoderOffset(int aSlot,
                    int aChannel,
                    int bSlot,
                    int bChannel,
                    int indexSlot,
                    int indexChannel)
```

Encoder constructor. Construct a Encoder given a and b modules and channels Uu I l f easUpd. U s in g oIndex pulse Uoæes 4x encoding.

Parameters:

- aSlot **WbTe**. a channel digital input module.
- aChannel **WbTe**. a channel digital input channel.
- bSlot **WbTe**. b channel digital input module.
- bChannel **WbTe**. b channel digital input channel.
- indexSlot **WbTe**. index channel digital input module.
- indexChannel **WbTe**. index channel digital input channel.

EncoderOffset

```
public EncoderOffset(int aChannel,
                    int bChannel,
                    boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module.

Parameters:

- aChannel **WbTe**. a channel digital input channel.
- bChannel **WbTe**. b channel digital input channel.
- reverseDirection **WbTe**. represents the orientation of the encoder and inverts the output values if necessary so Upward represents positive values.

EncoderOffset

```
public EncoderOffset(int aChannel,
                    int bChannel)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module.

Parameters:

- aChannel **WbTe**. a channel digital input channel.
- bChannel **WbTe**. b channel digital input channel.

EncoderOffset

```
public EncoderOffset(int aChannel,
                    int bChannel,
                    boolean reverseDirection,
                    CounterBase.EncodingType encodingType)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module.

Parameters:

- aChannel **WbTe**. a channel digital input channel.
- bChannel **WbTe**. b channel digital input channel.
- reverseDirection **WbTe**. represents the orientation of the encoder and inverts the output values if necessary so Upward represents positive values.
- encodingType **WbTe**. With either k1 X 60 X 6000 or counter1 X 602 X 00 encoding, Up, X selects 6000 an encoder FPR A 0000 is used and the returned counts will be 4x the encoder spec' value since all rising and falling edges are counted. If 01 X 00 or 02 is selected then a counter object will be used and the returned value will either exactly match the spec' count or be double 02 x 20 spec' count.

EncoderOffset

```
public EncoderOffset(int aChannel,
                    int bChannel,
                    int indexChannel,
                    boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module. U is an index pulse Upoes 4x encoding

Parameters:

- aChannel **WbTe**. a channel digital input channel.
- bChannel **WbTe**. b channel digital input channel.
- indexChannel **WbTe**. index channel digital input channel.
- reverseDirection **WbTe**. represents the orientation of the encoder and inverts the output values if necessary so Upward represents positive values.

EncoderOffset

```
public EncoderOffset(int aChannel,
                    int bChannel,
                    int indexChannel)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module. U s in an index pulse Uoes 4x encoding

Parameters:

- aChannel **Wb** Te. a channel digital input channel.
- bChannel **Wb** Te. b channel digital input channel.
- indexChannel **Wb** Te. index channel digital input channel.

EncoderOffset

```
public EncoderOffset(DigitalSource aSource,
                    DigitalSource bSource,
                    boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **Wb** Te. source that should be used Uo roe a channel.
- bSource **Wb** Te. source that should be used Uo roe b channel.
- reverseDirection **Wb** represents the orientation oUote encoder and inverts the output values iUnecessary so Uoward represents positive values.

EncoderOffset

```
public EncoderOffset(DigitalSource aSource,
                    DigitalSource bSource)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **Wb** Te. source that should be used Uo roe a channel.
- bSource **Wb** Te. source that should be used Uo roe b channel.

EncoderOffset

```
public EncoderOffset(DigitalSource aSource,
                    DigitalSource bSource,
                    boolean reverseDirection,
                    CounterBase.EncodingType encodingType)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **Wb** Te. source that should be used Uo roe a channel.
- bSource **Wb** Te. source that should be used Uo roe b channel.
- reverseDirection **Wb** represents the orientation oUote encoder and inverts the output values iUnecessary so Uoward represents positive values.
- encodingType **Wb** either k1 X 62 X 60 or o oinate if X 6o2 X oo encoding IUb, X oesed6oten an encoder FPR A ooeVtjis used and the returned counts will be 4x the encoder spec' dvalue since all rising and falling edges are counted. IUo1 X oo are Xected then a counter object will be used and the returned value will either exactly match the spec' doount or be double 02 x 2o spec' doount.

EncoderOffset

```
public EncoderOffset(DigitalSource aSource,
                    DigitalSource bSource,
                    DigitalSource indexSource,
                    boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **Wb** Te. source that should be used Uo roe a channel.
- bSource **Wb** Te. source that should be used Uo roe b channel.
- indexSource **Wb** Te. source that should be used Uo roe index channel.

rev = eseDirection Up represents the orientation of the encoder and inverts the output values if necessary so Upward represents positive values.

EncoderOffset

```
public EncoderOffset(DigitalSource aSource,
                    DigitalSource bSource,
                    DigitalSource indexSource)
```

Encoder constructor. Construct a Encoder given a and b channels as digital inputs. This is used in the case where the digital inputs are shared. The Encoder class will not allocate the digital inputs and assume that they already are counted.

Parameters:

- aSource **Writable** source that should be used to read a channel.
- bSource **Writable** source that should be used to read b channel.
- indexSource **Writable** source that should be used to read index channel.

Method Detail

getRaw

```
public int getRaw()
```

Description copied from class: [edu.wpi.first.wpilibj.Encoder](#)
Returns the raw value from the encoder. The raw value is the actual count unscaled by the 1 x 602 x 60 counts per revolution.

Overrides:

`getRaw` in class `Encoder`

Returns:

Current raw count from the encoder

reset

```
public void reset()
```

Resets the Encoder. Also undoes any offsets previously set.

Specified by:

`reset` in interface `CounterBase`

Overrides:

`reset` in class `Encoder`

setOffset

```
public void setOffset(int offset)
```

Sets the offset value for the Encoder.

Parameters:

offset **Writable** offset value for the encoder.

Hierarchy For Package com._604robotics.utils

Package Hierarchies:
All Packages

Class Hierarchy

- o java.lang.Object
 - o com._604robotics.Gt il **Beadband** edSourceoy im entents edGwpi.first.wpilibj.PIDS o **Ge**,
 - o com._604robotics.Gt il **DualVictor**oy im entents edGwpi.first.wpilibj.PIDOG t p ,Gt
 - o com._604robotics.Gt il **LinearController**
 - o edGwpi.first.wpilibj.**PIDController** yim entents edGwpi.first.wpilibj.parsing.IV t il it j
 - o com._604robotics.Gt il **ConvertingPIDController**
 - o com._604robotics.Gt il **uDo3nPIDController**
 - o edGwpi.first.wpilibj.**SensorBase**
 - o edGwpi.first.wpilibj.**Encoder** yim entents edGwpi.first.wpilibj.CoGnetrUase) edGwpi.first.wpilibj.parsing.ISensor)
edGwpi.first.wpilibj.PIDS o **Ge**,
 - o com._604robotics.Gt il **EncoderOfSet**
 - o com._604robotics.Gt il **EncoderPIDSource**
 - o edGwpi.first.wpilibj.**Gyro** yim entents edGwpi.first.wpilibj.parsing.ISensor) edGwpi.first.wpilibj.PIDS o **Ge**,
 - o com._604robotics.Gt il **Gyro360**oy im entents edGwpi.first.wpilibj.PIDS o **Ge**,
 - o edGwpi.first.wpilibj.**GyroHax**
 - o com._604robotics.Gt il **CompensatingGyro**
 - o edGwpi.first.wpilibj.**PWM**
 - o edGwpi.first.wpilibj.**SafePWM** yim entents edGwpi.first.wpilibj.3 o t o a f t j ,
 - o edGwpi.first.wpilibj.**Gictor** yim entents edGwpi.first.wpilibj.parsing.IDeviceController)
edGwpi.first.wpilibj.SpedController,
 - o com._604robotics.Gt il **SpringableGictor**
 - o edGwpi.first.wpilibj.**Relay** yim entents edGwpi.first.wpilibj.parsing.IDeviceController,
 - o com._604robotics.Gt il **SpringableRelay**
 - o edGwpi.first.wpilibj.**SolenoidBase** yim entents edGwpi.first.wpilibj.parsing.IDeviceController,
 - o edGwpi.first.wpilibj.**DoubleSolenoid**
 - o com._604robotics.Gt il **SpringableDoubleSolenoid**
 - o com._604robotics.Gt il **uDo3nPIDController**.Gain s
 - o com._604robotics.Gt il **VelocityController**
 - o com._604robotics.Gt il **ShoxController**

Interface Hierarchy

- o com._604robotics.Gt il **ShoxController**.A is
- o com._604robotics.Gt il **ShoxController**.Lutton
- o com._604robotics.Gt il **ShoxController**.Lutton.DPad
- o com._604robotics.Gt il **ShoxController**.Stick

Package com._604robotics.utils

Interface Summary

Interface	Description
XboxController.S is	E n u eration f o r e l a vailab e axes on the X b o ocontroller.
XboxController.Button	E n u eration f o r e l a vailab e b u t t o n s eoXnb ocontroller.
XboxController.Button.DPad	
XboxController.Stick	E n u eration f o r e l a vailab e sticks on the X b o ocontroller.

Class Summary

Class	Description
CompensatingG po	G y ro with a nual compensation- setting support.
ConvertingPIDController	A nextender of a PIDController that converts between units when getting and setting a setpoint.
DeadbandedSource	I o p ments a PIDS o u e, wrapping around another PIDS o u e, with a deadband range.
DualVictor	C o n t r o l V ictors like they j e one.
EncoderOffset	E r o d e r extender that return the value of E o d e r. g e t U)when pidGet is called.
EncoderPIDSource	E r o d e r extender that return the value of E o d e r. g e t U)when pidGet is called.
G po360	E x e n d e r class to constrain the output of a G y r o t o 3 6 0 d e g r e e s l o o p i n g .
LinearController	T h i s class implements a controller with a h o r i z o n t a l segment, a linear segment, and f i n a l l y a coasting segment.
SpringableDoubleSolenoid	E x e n d e r of a D o u b l e S o l e n o i d providing an easier control f l o w .
SpringableRelap	E x e n d e r of a R e l a y p r o v i d i n g an easier control f l o w .
SpringableVictor	E x e n d e r of a V i c t o r providing an easier control f l o w .
UpDownPIDController	A PIDController with d i f f e r e n t gains f o r u p and down.
UpDownPIDController.Gains	A s t r u c t u r e containing the P, and D gains.
VelocitpController	C l a s s f o r controlling a motorj svelocity , ather than its power directly .
XboxController	W a p p e r j o y s t i c k class f o r e l a b o x c o n t r o l l e r s .

com._604robotics.utils

Class UpDownPIDController.Gains

java.lang.Object
com._604robotics.utils.UpDownPIDController.Gains

Enclosing class:
UpDownPIDController

```
public static class UpDownPIDController.Gains
extends Object
```

A structure containing the P, I, and D gains.

Field Summary

Fields

Modifier and Type	Field and Description
double	D
double	I
double	P

Constructor Summary

Constructors

Constructor and Description
UpDownPIDController.Gains (double P, double I, double D)

Method Summary

Methods inherited from class java.lang.Object

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

Field Detail

P

public double P

I

public double I

D

public double D

Constructor Detail

UpDownPIDController.Gains

```
public UpDownPIDController.Gain(double P,  
                                double I,  
                                double D)
```

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

Interfaces

XboxController.Axis
XboxController.Button
XboxController.Button.DPad
XboxController.Stick

Classes

CompensatingGyro
ConvertingPIDController
DeadbandedSource
DualVictor
EncoderOffset
EncoderPIDSource
Gyro360
LinearController
SpringableDoubleSolenoid
SpringableRelay
SpringableVictor
UpDownPIDController
UpDownPIDController.Gains
VelocityController
XboxController

com._604robotics.utils

Class XboxController

java.lang.Object
com._604robotics.utils.f bxController

```
public class XboxController
extends Object
```

B rappe joystick class @r the f bx 360 controllers.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static interface	XboxController.Axis U mmeration @r the available axes on the f bx controller.
static interface	XboxController.Button U mmeration @r the available buttons on the f bx controller.
static interface	XboxController.Stick U mmeration @r the available sticks on the f bx controller.

Constructor Summary

Constructors

Constructor and Description
XboxController (int port) InitialiE @ new f bxController on the speciGie@port.
XboxController (Joystick joystick) InitialiE @ new f bxController @m the underlying joystick.

Method Summary

Methods

Modifier and Type	Method and Description
double	getAxis (int axis) Get the value oGhe speciGie@axis.
boolean	getButton (int button) Get whether or not the speciGie@button is currently pressed.
Joystick	getJoystick () Gets the underlying joystick object.
boolean	getStick (int stick) Get whether or not there(@ value reading on the stick.
boolean	getToggle (int button) Get the toggle state oGhe speciGie@button.
void	resetToggles ()) eset@ the toggle registry @r the contrller.
void	setDeadband (int axis, double lower, double upper) Sets the deadband @r a particular axis.

Methods inherited from class java.lang.Object

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

Constructor Details

XboxController

```
public XboxController(int port)
```

Initializes a new XboxController on the specified port.

Parameters:

- port The 25-pin port the controller is connected to.

XboxController

```
public XboxController(Joystick joystick)
```

Initializes a new XboxController from the underlying joystick.

Parameters:

- joystick The joystick to overlay the XboxController interface on.

Method Detail

getAxis

```
public double getAxis(int axis)
```

Get the value of the specified axis.

Parameters:

- axis One of the axis values specified in XboxController.Axis.

getStick

```
public boolean getStick(int stick)
```

Get whether or not there is a value reading on the stick.

Parameters:

- stick One of the stick values specified in XboxController.Stick.

Returns:

- Whether or not there is a value reading on the stick.

getButton

```
public boolean getButton(int button)
```

Get whether or not the specified button is currently pressed.

Parameters:

- button One of the button values specified in XboxController.Button.

resetToggles

```
public void resetToggles()
```

Resets the toggle registry for the controller.

getToggle

```
public boolean getToggle(int button)
```

Get the toggle state of the specified button.

Parameters:

- button One of the button values specified in XboxController.Button.

getJoystick() c B

```
public Joystick getJoystick()
```

Gets the underlying joystick object. B hat-is f bxController not good enough @r you?

Returns:

The underlying joystick object.

setDeadband

```
public void setDeadband(int axis,
                        double lower,
                        double upper)
```

Sets the deadband @r a particular axis.

Parameters:

- axis 1The axis to set the deadband @r.
- lower 1The lower bound oGhe deadband.
- upper 1The upper bound oGhe deadband.

com._604robotics.utils

Interface XboxController.Stick

Enclosing class:

[XboxController](#)

```
public static interface XboxController.Stick
```

Enumeration for the available sticks on the Xbox controller

Field Summary

Fields

Modifier and Type	Field and Description
static int	DPAD
static int	LEFT_STICK
static int	RIGHT_STICK

Field Detail

LEFT_STICK

```
static final int LEFT_STICK
```

See Also:

- [Constant Field Values](#)

RIGHT_STICK

```
static final int RIGHT_STICK
```

See Also:

- [Constant Field Values](#)

DPAD

```
static final int DPAD
```

See Also:

- [Constant Field Values](#)

com._604robotics.utils

Interface XboxController.Button

Enclosing class:

[X](#) [bxController](#)

```
public static interface XboxController.Button
```

Enumeration for the available buttons on the X bx controller.

Nested Class Summary

Nested Classes

Modifier and Type	Interface and Description
static interface	XboxController.Button.DPad

Field Summary

Fields

Modifier and Type	Field and Description
static int	A
static int	B
static int	Back
static int	EitherTrigger
static int	LB
static int	LeftStick
static int	LT
static int	RB
static int	RightStick
static int	RT
static int	Start
static int	X
static int	Y

Field Detail

A

```
static final int A
```

See Also:

- [Constant Field G ales](#)

B

```
static final int B
```

See Also:

- [Constant Field G ales](#)

X

```
static final int X
```

See Also:

See Also:

[Constant Field G ales](#)

Y

`static final int Y`

See Also:

[Constant Field G ales](#)

LB

`static final int LB`

See Also:

[Constant Field G ales](#)

RB

`static final int RB`

See Also:

[Constant Field G ales](#)

Back

`static final int Back`

See Also:

[Constant Field G ales](#)

Start

`static final int Start`

See Also:

[Constant Field G ales](#)

LeftStick

`static final int LeftStick`

See Also:

[Constant Field G ales](#)

RightStick

`static final int RightStick`

See Also:

[Constant Field G ales](#)

LT

`static final int LT`

See Also:

[Constant Field G ales](#)

RT

`static final int RT`

See Also:

[Constant Field Goals](#)

EitherTrigger

```
static final int EITHER_TRIGGER
```

See Also:

[Constant Field Goals](#)

com._604robotics.utils

Class SpringableDoubleSolenoid

java.lang.Object
edu.wpi.first.wpilibj.SensorBase
edu.wpi.first.wpilibj.SolenoidBase
edu.wpi.first.wpilibj.DoubleSolenoid
com._604robotics.utils.SpringableDoubleSolenoid

Implemented interfaces:

IDevice, IDeviceController

public class SpringableDoubleSolenoid
extends DoubleSolenoid

Extended from a DoubleSolenoid providing an easier control flow. When an output is set for the DoubleSolenoid, it is considered "sprung". When the reload method is called, if the victor is sprung, it unsprings the DoubleSolenoid. If the DoubleSolenoid is not sprung, then the output is set to the default output. In this way, the DoubleSolenoid will only be moving when you tell it to. Use this in a loop or something, and call reload at the end. No more worries about code paths that don't update the DoubleSolenoids.

Author:

Michael Smith

Nested Class Summary

Nested classes/interfaces inherited from class edu.wpi.first.wpilibj.DoubleSolenoid

DoubleSolenoid.Value

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.SolenoidBase

m_allocated, m_moduleNumber

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Constructor Summary

Constructors

Constructor and Description

SpringableDoubleSolenoid(int forwardChannel, int reverseChannel, DoubleSolenoid.Value defaultDirection)
Initializes a new SpringableDoubleSolenoid.

SpringableDoubleSolenoid(int moduleNumber, int forwardChannel, int reverseChannel, DoubleSolenoid.Value defaultDirection)
Initializes a new SpringableDoubleSolenoid.

Method Summary

Methods

Modifier and Type Method Description

boolean getSprung()
Has the DoubleSolenoid been sprung?

void reload()
If the DoubleSolenoid has been sprung, unspring it. If not, set the output to the default output.

void	set (DoubleSolenoid.Value direction) Sets the direction of the DoubleSolenoid.
void	spring () Springs the DoubleSolenoid.
M etbds inherited from c sa edu.wpi.first.wpilibj.DoubleSolenoid	
free, get	
M etbds inherited from c sa edu.wpi.first.wpilibj.Solenoid Bsa	
getAll, getAllFromDefaultModule, getAllFromModule, set	
M etbds inherited from c sa edu.wpi.first.wpilibj.SensorBase	
checkAnalogChannel, checkAnalogModule, checkDigitalChannel, checkDigitalModule, checkPWMChannel, checkPWMModule, checkRelayChannel, checkRelayModule, checkSolenoidChannel, checkSolenoidModule, getDefaultAnalogModule, getDefaultDigitalModule, getDefaultSolenoidModule, setDefaultAnalogModule, setDefaultDigitalModule, setDefaultSolenoidModule	
M etbds inherited from c sa java.lang.Objec t	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Construc or Detail
SpringableDoubleSolenoid
<pre>public SpringableDoubleSolenoid(int forwardChannel, int reverseChannel, DoubleSolenoid.Value defaultDirection)</pre> <p>Initiali) es new SpringableDoubleSolenoid.</p> <p>Param etes:</p> <p>forwardChannel ! The forward channel of the DoubleSolenoid.</p> <p>reverseChannel ! The reverse channel of the DoubleSolenoid.</p> <p>defaultDirection ! The def alt direction for reloads.</p>
SpringableDoubleSolenoid
<pre>public SpringableDoubleSolenoid(int moduleNumber, int forwardChannel, int reverseChannel, DoubleSolenoid.Value defaultDirection)</pre> <p>Initiali) es new SpringableDoubleSolenoid.</p> <p>Param etes:</p> <p>moduleNumber ! The slot number of the solenoid module.</p> <p>forwardChannel ! The forward channel of the DoubleSolenoid.</p> <p>reverseChannel ! The reverse channel of the DoubleSolenoid.</p> <p>defaultDirection ! The def alt direction for reloads.</p>

M etbd Detail
getSprung
<pre>public boolean getSprung()</pre> <p>Has the DoubleSolenoid been sprung?</p> <p>R atrns:</p> <p>Whether or not the DoubleSolenoid has been sprung.</p>

spring

```
public void spring()
```

Springs the DoubleSolenoid.

set

```
public void set(DoubleSolenoid.Value direction)
```

Sets the direction of the DoubleSolenoid.

O verrides:

```
set in class DoubleSolenoid
```

Parameters:

`direction` ! The direction to set.

reload

```
public void reload()
```

If the DoubleSolenoid has been sprung, unspring it2if not, set the output to the def alt output.

[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](#)

com._604robotics.utils

Class VelocityController

java.lang.Object
com._604robotics.utils.FVelocityController

```
public class VelocityController
extends Object
```

Class for controlling a motor's velocity rather than its power directly. Uses a PID loop to scale to said velocity and a distance encoder for feedback.

Author:

Michael Smith & Zevir Parker

Constructor Summary

Constructors

Constructor and Description
VelocityController (double p, double i, double d, Encoder encoderLeft, Encoder encoderRight, RobotDrive robotDrive, Gyro gyro) Initializes a new FVelocityController.

Method Summary

Methods

Modifier and Type	Method and Description
void	disable () Disables the FVelocityController.
void	enable () (nab lase f FVelocityController.
double	getActualVelocity ()) etsthe actual3current velocity.
double	getVelocity ()) etsthe current target velocity.
boolean	isEnabled () Is the FVelocityController currently enabled?
void	setAngleGains (double pAngle, double iAngle, double dAngle) 2 asedn gyro angles TODO E javadoc
void	setGains (double p, double i, double d) ! eonfigures the gains on the PIDController.
void	setVelocity (double velocity) Sets the target velocity.

Methods inherited from class java.lang.Object

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

Constructor Detail

VelocityController

```
public VelocityController(double p,
                           double i,
                           double d,
                           Encoder encoderLeft,
                           Encoder encoderRight,
                           RobotDrive robotDrive,
                           Gyro gyro)

Initializes a new FVelocityController.
```

Parameters:

`p` The proportional term for the PIDController.

`i` The integral term for the PIDController.

`d` The derivative term for the PIDController.

`encoder` The encoder to use for feedback.

`output` The PIDOutput to control. Usually some sort of motor.

Method Detail

getVelocity

```
public double getVelocity ()
```

) the current target velocity.

Returns:

The current target velocity.

getActualVelocity

```
public double getActualVelocity ()
```

) the actual current velocity.

Returns:

The actual current velocity.

setVelocity

```
public void setVelocity (double v)
```

Sets the target velocity.

Parameters:

`v` The target velocity to set.

setGains

```
public void setGains(double p,
                    double i,
                    double d)
```

! configures the gains on the PIDController.

Parameters:

`p` The proportional term for the PIDController.

`i` The integral term for the PIDController.

`d` The derivative term for the PIDController.

setAngleGains

```
public void setAngleGains(double pAngle,
                        double iAngle,
                        double dAngle)
```

2 use on gyro angles TODO E javadoc

Parameters:

`p` The

`i` The

`d` The

enable

```
public void enable()
{
    nab = false;
    if (cityController.
```

disable

```
public void disable()
```

Disables the `FileCityController`.

isEnabled

```
public boolean isEnabled()
```

Is the `cityController` currently enabled?

Returns:

- whether or not the `cityController` is currently enabled.

com._604robotics.utils

Interface XboxController.Button.DPad

Enclosing interface:

X [bxController.f utton](#)

```
public static interface XboxController.Button.DPad
```

Field Summary

Fields

Modifier and Type	Field and Description
static int	Down
static int	Left
static int	Right
static int	Up

Field Detail

Up

```
static final int Up
```

See Also:

[Constant Field B ales](#)

Down

```
static final int Down
```

See Also:

[Constant Field B ales](#)

Left

```
static final int Left
```

See Also:

[Constant Field B ales](#)

Right

```
static final int Right
```

See Also:

[Constant Field B ales](#)

com._604robotics.utils

Class Gyro360

java.lang.Object
 edu.wpi.first.wpilibj.SensorBase
 edu.wpi.first.wpilibj.Gyro
 com._604robotics.utils.Gyro360

All Implemented Interfaces:

IDevice, ISensor, PIDSource

```
public class Gyro360
extends Gyro
implements PIDSource
```

Extended class to constrain the output of a Gyro to 360 degrees, looping.

Author:

Michael Smith

Field Summary

Fields inherited from class edu.wpi.first.wpilibj.SensorBase

kAnalogChannels, kAnalogModules, kDigitalChannels, kPwmChannels, kRelayChannels, kSolenoidChannels, kSolenoidModules, kSystemClockTicksPerMicrosecond

Constructor Summary

Constructors

Constructor and Description
Gyro360 (AnalogChannel channel) Initializes a new Gyro360 on the specified AnalogChannel.
Gyro360 (int port) Initializes a new Gyro360 on the specified PWM port.
Gyro360 (int slot, int port) Initializes a new Gyro360 on the specified PWM port on the specified module port.

Method Summary

Methods

Modifier and Type	Method and Description
double	getAngle () Gets the angle of the gyro, constrained to 360 degrees.
double	pidGet () Implements the pidGet function in the type PIDSource, allowing this class to be used as such.

Methods inherited from class edu.wpi.first.wpilibj.Gyro

free, reset, setSensitivity

Methods inherited from class edu.wpi.first.wpilibj.SensorBase

checkAnalogChannel, checkAnalogModule, checkDigitalChannel, checkDigitalModule, checkPWMChannel, checkPWMModule, checkRelayChannel, checkRelayModule, checkSolenoidChannel, checkSolenoidModule, getDefaultAnalogModule, getDefaultDigitalModule, getDefaultSolenoidModule, setDefaultAnalogModule, setDefaultDigitalModule, setDefaultSolenoidModule

Methods inherited from class java.lang.Object

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

Constructor Detail

Gyro360

```
public Gyro360(int port)
```

Initializes a new Gyro360 on the specified PWM port. Note that port must be 1 or 2 !

Parameters:

`port` - The PWM port the gyro is plugged into. Must be 1 or 2 !

Gyro360

```
public Gyro360(int slot,  
               int port)
```

Initializes a new Gyro360 on the specified PWM port on the specified module port. Note that port must be 1 or 2 !

Parameters:

`slot` - The module slot the gyro is plugged into.

`port` - The PWM port the gyro is plugged into. Must be 1 or 2 !

Gyro360

```
public Gyro360(AnalogChannel channel)
```

Initializes a new Gyro360 on the specified AnalogChannel. Note that port must be 1 or 2 !

Parameters:

`channel` - The AnalogChannel the gyro is plugged into.

Method Detail

getAngle

```
public double getAngle()
```

Gets the angle of the gyro, constrained to 360 degrees.

Overrides:

`getAngle` in class `Gyro`

Returns:

The angle of the gyro, constrained to 360 degrees.

pidGet

```
public double pidGet()
```

Implements the pidGet() function in the type PIDSource, allowing this class to be used as such.

Specified by:

`pidGet` in interface `PIDSource`

Overrides:

`pidGet` in class `Gyro`

Returns:

The angle of the gyro, constrained to 360 degrees.

All Classes

Packages

com._604robotics.robot2012
com._604robotics.robot2012.aiming
com._604robotics.robot2012.autonomous
com._604robotics.robot2012.balancing
com._604robotics.robot2012.camera
com._604robotics.robot2012.configuration
com._604robotics.robot2012.machine
com._604robotics.robot2012.physics
com._604robotics.robot2012.rotation
com._604robotics.robot2012.vision
com._604robotics.utils
edu.wpi.first.wpilibj
frc.vision

Constant Field Values

Contents

com._604robotics.*

com._604robotics.*

com._604robotics.robot2012.configuration.ActuatorConfiguration

Modifier and Type	Constant Field	Value
public static final double	ACCELEROMETER_DRIVE_POWER	0.5
public static final double	ELEVATOR_POWER_MAX	0.8
public static final double	ELEVATOR_POWER_MIN	-0.8
public static final double	HOPPER_POWER	0.8
public static final double	HOPPER_POWER_REVERSE	-0.5
public static final double	PICKUP_POWER	0.8
public static final double	TURRET_ROTATION_POWER_MAX	0.4
public static final double	TURRET_ROTATION_POWER_MIN	-0.4

com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR

Modifier and Type	Constant Field	Value
public static final int	HIGH	1540
public static final int	LOW	0
public static final int	MEDIUM	663
public static final int	OKAY_TO_TURN	1300

com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.DEADBAND

Modifier and Type	Constant Field	Value
public static final int	HIGH	1490
public static final int	LOW	35
public static final int	MEDIUM_LOWER	611
public static final int	MEDIUM_UPPER	691

com._604robotics.robot2012.configuration.ActuatorConfiguration.ELEVATOR.TOLERANCE

Modifier and Type	Constant Field	Value
public static final int	HIGH	1505
public static final int	LOW	25
public static final int	MEDIUM_LOWER	631
public static final int	MEDIUM_UPPER	671

com._604robotics.robot2012.configuration.ActuatorConfiguration.TURRET_POSITION

Modifier and Type	Constant Field	Value
public static final double	FORWARD	0.0
public static final double	LEFT	-45.0
public static final double	RIGHT	45.0
public static final double	TOLERANCE	1.0

com._604robotics.robot2012.configuration.AutonomousConfiguration

Modifier and Type	Constant Field	Value
public static final double	BACKWARD_DISTANCE	-1000.0
public static final double	BACKWARD_DISTANCE_SIDES	-500.0
public static final double	BACKWARD_DRIVE_POWER	-0.5
public static final double	FORWARD_DISTANCE	1000.0
public static final double	FORWARD_DRIVE_POWER	0.5

com._604robotics.robot2012.configuration.ButtonConfiguration.Driver

Modifier and Type	Constant Field	Value
public static final int	AUTO_BALANCE	3
public static final int	GYRO_RESET	7
public static final int	SHIFT	13
public static final int	TOGGLE_PICKUP	6

com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator

Modifier and Type	Constant Field	Value
public static final int	AIM_AND_SHOOT	12
public static final int	PICKUP	11
public static final int	TOGGLE_ANGLE	5
public static final int	TOGGLE_HEIGHT	6
public static final int	TOGGLE_LIGHT	7

com._604robotics.robot2012.configuration.ButtonConfiguration.Manipulator.Elevator

Modifier and Type	Constant Field	Value
public static final int	DOWN	1
public static final int	FORWARD	4
public static final int	LEFT	3
public static final int	RIGHT	2

com._604robotics.robot2012.configuration.PortConfiguration.Controllers

Modifier and Type	Constant Field	Value
public static final int	DRIVE	1
public static final int	MANIPULATOR	2

com._604robotics.robot2012.configuration.PortConfiguration.Encoders

Modifier and Type	Constant Field	Value
public static final int	ELEVATOR_A	8
public static final int	ELEVATOR_B	9
public static final int	TURRET_ROTATION_A	7
public static final int	TURRET_ROTATION_B	6

com._604robotics.robot2012.configuration.PortConfiguration.Encoders.Drive

Modifier and Type	Constant Field	Value
public static final int	LEFT_A	13
public static final int	LEFT_B	14
public static final int	RIGHT_A	12
public static final int	RIGHT_B	11

com._604robotics.robot2012.configuration.PortConfiguration.Motors

Modifier and Type	Constant Field	Value
public static final int	ELEVATOR_LEFT	7
public static final int	ELEVATOR_RIGHT	8
public static final int	HOPPER	4
public static final int	LEFT_DRIVE	1
public static final int	PICKUP	6
public static final int	RIGHT_DRIVE	9
public static final int	SHOOTER_LEFT	2
public static final int	SHOOTER_RIGHT	3
public static final int	TURRET_ROTATION	5

com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics

Modifier and Type	Constant Field	Value
public static final int	COMPRESSOR	6
public static final int	PRESSURE_SWITCH	5

com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.HOPPER_SOLENOID

Modifier and Type	Constant Field	Value
public static final int	FORWARD	4

public static final int	FORWARD	4
public static final int	REVERSE	3

com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.PICKUP_SOLENOID

Modifier and Type	Constant Field	Value
public static final int	IN	2
public static final int	OUT	1

com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHIFTER_SOLENOID

Modifier and Type	Constant Field	Value
public static final int	HIGH_GEAR	8
public static final int	LOW_GEAR	7

com._604robotics.robot2012.configuration.PortConfiguration.Pneumatics.SHOOTER_SOLENOID

Modifier and Type	Constant Field	Value
public static final int	LOWER_ANGLE	6
public static final int	UPPER_ANGLE	5

com._604robotics.robot2012.configuration.PortConfiguration.Relays

Modifier and Type	Constant Field	Value
public static final int	RING_LIGHT_PORT	4

com._604robotics.robot2012.configuration.PortConfiguration.Sensors

Modifier and Type	Constant Field	Value
public static final int	ACCELEROMETER	3
public static final int	ELEVATOR_LIMIT_SWITCH	1
public static final int	GYRO_BALANCE	1
public static final int	GYRO_HEADING	2

com._604robotics.robot2012.configuration.SensorConfiguration

Modifier and Type	Constant Field	Value
public static final double	ACCELEROMETER_SENSITIVITY	1.0
public static final double	ACCELEROMETER_UPPER_RADIAN	0.7854
public static final double	GYRO_DRIFT	0.0238095238
public static final int	TURRET_CALIBRATION_OFFSET	-471

com._604robotics.robot2012.configuration.SensorConfiguration.Encoders

Modifier and Type	Constant Field	Value
public static final double	LEFT_DRIVE_INCHES_PER_CLICK	1.0
public static final double	RIGHT_DRIVE_INCHES_PER_CLICK	1.0
public static final double	TURRET_DEGREES_PER_CLICK	0.172801106

com._604robotics.robot2012.machine.ElevatorMachine.ElevatorState

Modifier and Type	Constant Field	Value
public static final int	HIGH	0
public static final int	LOW	2
public static final int	MEDIUM	1
public static final int	PICKUP_OKAY	3
public static final int	TURRET_OKAY	4

com._604robotics.robot2012.machine.PickupMachine.PickupState

Modifier and Type	Constant Field	Value
public static final int	IN	1
public static final int	OUT	0

com._604robotics.robot2012.machine.ShooterMachine.ShooterState

Modifier and Type	Constant Field	Value
public static final int	SHOOTING	0

com._604robotics.robot2012.machine.TurretMachine.TurretState

Modifier and Type	Constant Field	Value
public static final int	AIMED	1
public static final int	FORWARD	2
public static final int	LEFT	3
public static final int	RIGHT	4
public static final int	SIDEWAYS	0

com._604robotics.utils.XboxController.Axis

Modifier and Type	Constant Field	Value
public static final int	LEFT_STICK_X	1
public static final int	LEFT_STICK_Y	2
public static final int	RIGHT_STICK_X	4
public static final int	RIGHT_STICK_Y	5

com._604robotics.utils.XboxController.Button

Modifier and Type	Constant Field	Value
public static final int	A	1
public static final int	B	2
public static final int	Back	7
public static final int	EitherTrigger	13
public static final int	LB	5
public static final int	LeftStick	9
public static final int	LT	11
public static final int	RB	6
public static final int	RightStick	10
public static final int	RT	12
public static final int	Start	8
public static final int	X	3
public static final int	Y	4

com._604robotics.utils.XboxController.Button.DPad

Modifier and Type	Constant Field	Value
public static final int	Down	15
public static final int	Left	16
public static final int	Right	17
public static final int	Up	14

com._604robotics.utils.XboxController.Stick

Modifier and Type	Constant Field	Value
public static final int	DPAD	6
public static final int	LEFT_STICK	1
public static final int	RIGHT_STICK	4

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[Prev](#) [Next](#) [Frames](#) [No Frames](#) [All Classes](#)