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.**DualVictor** (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**

All Classes

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 yearï¿½s ï¿½robot problemï¿½. 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.

edu.wpi.first.wpilibj

Classes

GyroHax

Deprecated API

Contents

Overview Package Class Tree Deprecated Index Help

Prev Next Frames No Frames All Classes

All Classes

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

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 page lists the static final fields and their values.

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

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z

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.

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.

frc.vision

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:

Specified by:

`pidGet` in interface `PIDSource`

Returns:

The difference between the two encoder values.

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.

Overview Package Class Tree Deprecated Index Help

Prev Package Next Package Frames No Frames All Classes

com._604robotics.robot2012.autonomous

Classes

PIDDriveEncoderDifference
PIDDriveEncoderOutput
PIDDriveGyro

com. _ 6 0 4 rob otics. rob ustd2 com1 us2 . a

Class PIDDriveGyro

java.lang.Object
  com. _ 6 0 4 rob otics. rob ustd2 com1 us2.PIDDriveUyro

All Implemented Interfaces:

PIDOutput

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

Driving sh im Ror th yrogbased PIDBturning controller th ing.

Author:

M ich aSmith

Constructor Summary

Constructors

Constructor and Description
PIDDriveGyro(RobotDrive driveTrain) Initiali, es a new PIDDrieUyro(b ased on th evgin z ob otDrie.

Method Summary

Methods

Modifier and Type	Method and Description
void	**pidWrite**(double output) z rites th e utput Rrom th e PIDController to th e z obvet(Drith e Rorm oRuntvalue.

Methods inherited from class java.lang.Object

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

Constructor Detail

PIDDriveGyro

```
public PIDDriveGyro(RobotDrive driveTrain)
```

Initiali, es a new PIDDriveUyro(b ased on th evgin z ob otDrie.

Parameters:

driveTrain B Th e z obvet(Dob j ect to control.

Method Detail

pidWrite

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

W rites th e utput Rrom th e PIDController to th e z obvet(Drith e Rorm oRuntvalue.

Specified by:

pidWrite in interRace PIDOutput

Parameters:

`output` B Th eutput oR th e PIDController.

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
PIDDriveEncoderOutput(RobotDrive driveTrain) Initializes a new PIDDriveEncoderOutput.
PIDDriveEncoderOutput(RobotDrive driveTrain, boolean inversion) Initializes a new PIDDriveEncoderOutput.

Method Summary

Methods

Modifier and Type	Method and Description
void	**pidWrite**(double output) 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.

j etl od 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.

Hierarchy For Package com._604robotics.robot2012.nhysics

Package HierarchiesR
All Packages

l aCss Hierarchy

- java.lang.D**bject**
  - com._604robotics.robot2012.p6 0 si.csLaOOreElR dB
  - com._604robotics.robot2012.p6 0 si.cs**Physics**
  - com._604robotics.robot2012.p6 0 si.cs**Shooter**p gB**ePick**

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.

Overview Package Class Tree Deprecated Index Help

Prev Package Next Package Frames No Frames All Classes

com._604robotics.robot2012.physics

Classes

BallFireInfo
Physics
ShooterAnglePick

com._604robotics.robot2012.physics

Class Physics

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

```
public class Physics
extends Object
```

U sed f or deter mining launch velocities of the ball. It gives velocity as a function of displacement and final vertical velocity

Author:

K evin 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, t hen determines the firing velocities (and t he) f or a given distance h(oriz ontal), and vertical).
Point2d	**betterVersionOfgetFiringVelocity**(double distH, double distV, double verticalVel) This function determines the firing velocities (and t he) f or a given distance h(oriz ontal), 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 ontal), and vertical) and the angle of t he shooter.
static double	**velToPow**(double vel) R eturns 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)
```

R eturns 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 time) 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 time) 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/right 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, and horizontalAngle to fire the ball at K eventuallyz

com. _ 6 0 4 rob otics. rob otyS2cs0 1 2 . ph

Class ShooterAnglePick

java.lang.Object
com. _ 6 0 4 rob otics. rob otyS2cs0.Sho2terAnglePick

```
public class ShooterAnglePick
extends Object
```

EnumR ish th ing oB angles to sh oot at.

Author:

, win 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.

com._604robotics.robot2012.physics

Class BallFireInfo

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

```
public class BallFireInfo
extends Object
```

Class representing info for firing a ball.

Author:

Brian 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) Initializes a new BallFireInfo.

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)
```

Initiali, es a new f ailreInfo.

Parameters:

`angle` K An angle.

`speed` K A speed.

`horizontalAngle` K A h ori, ontal angle.

com._6047.robotics.robot2012.balancing

Class Balancing

java.lang.Object
  com._6047.robotics.robot2012.balancing.Balancing

```
public class Balancing
extends Object
```

Rutility class for automated balancing assistance.

Author:

Kevin Parker

Constructor Summary

Constructors

Constructor and Description
`Balancing()`

Method Summary

Methods

Modifier and Type	Method and Description
`static double`	`getSpeedforBalance(double balGyroReading)` Given a specific gyro reading, returns 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

```
public Balancing()
```

Method Detail

getSpeedforBalance

```
public static double getSpeedforBalance(double balGyroReading)
```

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

Parameters:

`balGyroReading` - A gyro reading.

Returns:

The speed you should going at.

Hierarchy For Package com._604robotics.robot2012.balancing

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Dbject
  - com._604robotics.robot2012.balancing.l alancing

Overview Package Class Tree Deprecated Index Help

Prev ext Frames mraNes All Classes

Package com._604robotics.robot2012.balancing

Class	Summary

Class	Description
Balancing	Utility class for automated balancing assistance.

com._604robotics.robot2012.balancing

Classes

Balancing

com._604robotics.robot2012.machine

Interface StrangeMachine

All Known Implementing Classes:

j lcatorMachinef PickupMachinef ShooterMachinef TurretMachine

---

```
public interface StrangeMachine
```

State manager for various components of the robot. B sedor 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

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

Tests if the Machine has yet attained the target state.

Parameters:

state , The target state.

Returns:

K h ethoenot the Machine has attained the target state.

crank

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

Causes the Machine to strive for the target state.

Parameters:

state , The state to strive for.

Returns:

K h ethoenot the target state has been reached.

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 esa 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 esa 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 ethoeenot 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 ethoeenot 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 ales

com._604robotics.robot2012.machine

Interface ElevatorMachine.ElevatorState

Enclosing class:

j levatorMachine

---

public static interface **ElevatorMachine. ElevatorState**

f arious 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 f alues

MEDIUM

static final int MEDIUM

See Also:

Constant Field f alues

LOW

static final int LOW

See Also:

Constant Field f alues

PICKUP_OKAY

static final int PICKUP_OKAY

See Also:

Constant Field f alues

TURRET_OKAY

static final int TURRET_OKAY

See Also:

Constant Field f alues

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 possib le states th urret could b e 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 j ales

AIMED

`static final int AIMED`

See Also:

Constant Field j ales

FORWARD

`static final int FORWARD`

See Also:

Constant Field j ales

LEFT

`static final int LEFT`

See Also:

Constant Field j ales

RIGHT

`static final int RIGHT`

See Also:

Constant Field j ales

Hierarchy For Package com._604robotics.robot2012.machine

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
  - com._604robotics.robot2012.mac6 ine.EvolvingMachine (implements com._604robotics.robot2012.mac6 ine4 trangeb ac6 2ne
  - com._604robotics.robot2012.mac6 ine.PickupMachine (implements com._604robotics.robot2012.mac6 ine4 trangeb ac6 2ne
  - com._604robotics.robot2012.mac6 ine.ShooterMachine (implements com._604robotics.robot2012.mac6 ine4 trangeb ac6 2ne
  - com._604robotics.robot2012.mac6 ine.TurretMachine (implements com._604robotics.robot2012.mac6 ine4 trangeb ac6 2ne

Interface Hierarchy

- com._604robotics.robot2012.mac6 ine.IndexingMachine.IndexingState
- com._604robotics.robot2012.mac6 ine.PickupMachine.PickupState
- com._604robotics.robot2012.mac6 ine.shooterMachine.shooterState
- com._604robotics.robot2012.mac6 ine.strangeMachine
- com._604robotics.robot2012.mac6 ine.TurretMachine.TurretState

Package com._604robotics.robot2012.machine

Interface Summary

Interface	Description
ElevatorMachine.ElevatorState	Various possible states that can be entered.
PickupMachine.PickupState	Possible states the pickup could be put in.
ShooterMachine.ShooterState	The possible states that could be entered.
StrangeMachine	A state manager for various components of the robot.
TurretMachine.TurretState	The possible states that could be entered.

Class Summary

Class	Description
ElevatorMachine	A machine to control the elevator.
PickupMachine	A machine to control the pneumatic pickup.
ShooterMachine	A machine to control the shooter compressor system during firing.
TurretMachine	A machine to control the turret.

com._604robotics.robot2012.machine

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 SIDE, KNOB.
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

TurretMachine

```
public TurretMachine(AbstractController controller,
                     RotationProvider provider,
                     Encoder encoder)
```

Initializes a new TurretMachine.

Parameters:

`controller` - The PIDController to control.

`prov der` - The - otationProvider to draw aiming data from.

`encoder` - The encoder measuring the horif ontal position of the turret.

Method Detail

test

`public boolean test(int state)`

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

I pecified B O n

`test` in interface `StrangeMachine`

Parameters:

`state` - The target state.

Returnsn

K h ethoeenot the Machine has attained the target state.

crank

`public boolean crank (int state)`

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

I pecified B O n

`crank` in interface `StrangeMachine`

Parameters:

`state` - The state to strive for.

Returnsn

K h ethoeenot the target state has been reached.

setTurretI idewaOsPosition

`public v kdisetD urretSidewaysAosition(double turretSidewaysAosition)`

Sets the position to use as BSID, K ASB .

Parameters:

`turretSidewaysAosition` - The position to use as BSID, K ASB 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 pick up) Initialif esa new PickupMachine.

Method Summary

Methods

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

Methods inherited from class java.lang.Object

clone, eq als, f maliye, g eA ass, hashCode, notifS, notifS r ,lltoString, wait, wait, wait

Constructor Detail

PickupMachine

```
public h ik up achine(DoubleSolenoid pickup)
```

Initialif esa 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` BThe 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` BThe state to strive for.

Returns:

Whether or not the target state has been reached.

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 arios possible states the elevator can be in.

Constructor Summary

Constructors

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

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 yet attained the target state.

Methods inherited from class java.lang.Object

clone, equals, finalize, getY lass, hashYode, notifA, notifA S lltoString, r ait, r ait, r ait

Constructor Detail

ElevatorMachine

```
public w ehatorMachine(D ( Dnyroller controller,
                       w noder encoder)
```

InitialiB es a new f vetorMachine.

Parameters:

`controller` K A PIDController to control.

`encoder` K Th e encoder monitoring th evetorz s vertical position.

Method Detail

test

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

Description copied from interface: `StrangeMachine`
Tests i, th e M ach ine yet attained the target state.

Specified by:

`test` in inter, ace `StrangeMachine`

Parameters:

`state` K Th e target state.

Returns:

W h eth er or not th e M ach ine h as attained th state target

crank

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

Description copied from interface: `StrangeMachine`
Causes the Machine to strive , or th e target state.

Specified by:

`crank` in inter, ace `StrangeMachine`

Parameters:

`state` K Th e state to sti , or.

Returns:

W h eth er or not th e target state h as b een reach ed.

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.MI Det
    - edu.wpi.first.wpilibj.RobotBase
      - edu.wpi.first.wpilibj.Simp eRobot
        - com.M h .obotics.robot6 . 0R6bot2012O arGge

Package com._604robotics.robot2012

Class SSmmary

Class	u escription
R bot2012OraS g	Main class f othe 2 0 1ro2ot code.

Overview Package Class Tree Deprecated Index Help

Prev Package Next Package Frames No Frames All Classes

com._604robotics.robot2012

Classes

Robot2012Orange

com._6047robotics.robot2012

Class Robot2012Orange

java.lang.Object
  javax.microedition.midlet.MIDlet
    edu.wpi.first.wpilibj.RobotBase
      edu.wpi.first.wpilibj.SimpleRobot
        com._6047robotics.robot2012.Robot2012Orange

```
public class Robot2012Orange
extends SimpleRobot
```

Main class for the 2012 robot code.

Author:

Michael Smith, Kevin Parker, Sebastian Merz, Aaron Wang, Colin Aitken

Field Summary

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

ERROR_TO_EDESTINATION_mPORT_ROBOT_TASK_PRIORITY

Constructor Summary

Constructors

Constructor and Description
Robot2012Orange() Constructor.

Method Summary

Methods

Modifier and Type	Method and Description
void	**aimAndShoot**() Aim at backboard, shoot.
void	**autonomous**() Automated drive for autonomous mode.
static double	**deadband**(double xValue, double upperBound, double lowerBound, double coDirectedValue) If a value is within a range, set it to a specific value.
void	**disabled**() The robot is disabled.
static boolean	**isInRange**(double xValue, double upperRange, double lowerRange) Figures out if a value is within a specific range.
void	**operatorControl**() Operator-controlled drive for Teleop mode.
void	**robotInit**() Initializes the robot on start.

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

RobotMain, startCompetition

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

destroyApplicationRegistered, getBooleanProperty, getWatchdog, isAutonomous, isDisabled, isEnabled, isNewDataAvailable, isOperatorControl, isSystemActive, pauseApplication, startApplication

Methods inherited from class javax.microedition.midlet.MIDlet

getAppPropertyDouble, notifyDestroyed, notifyPaused, resumeRequest

Methods inherited from class java.lang.Object

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 on startup. 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 xValue 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 backboard, shoot.

autonomous

```
public void autonomous()
```

Automated drive for autonomous mode. If in middle, very far forward, knock down bridge, turn 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 for the bridge, all pickup, retrieval, aiming, firing, angle adjustments, light control, elevator control - both automated and manual - pneumatics, firing, and various other things.

Overrides:

operatorControl in class SimpleRobot

disabled

```
public void disabled()
```

The robot is disabled. Like zen gogglesz, this 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 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

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" 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`

com._604robotics.robot2012.rotation

Class NaiveRotationProvider

java.lang.Object
  com._604robotics.robot2012.rotation.NaiveR rotationProvider

All Implemented Interfaces:

R rotationProvider

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

A naive implementation of a R rotationProvider,

Author:

Michael Smith

Constructor Summary

Constructors

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

Method Summary

Methods

Modifier and Type	Method and Description
void	**setDefaultPosition**(double defaultPosition) Sets the "default" o pitions, 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(PID Sntroller controller,
                     CameraInterface cameraInterface,
                     E ncoderencoderTurret)
```

Initializes a new NaiveR rotationProvider, 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" orientation, 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`

Hierarchy For Package com._604robotics.robot2012.rotatioS

Package Hierarchies:
All Packages

x aSs Hierarchy

- java.lang.Sbject
  - com._604robotics.robot2012.rotation.n GmyRotatioSProvid er-implements 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)

GSterkace Hierarchy

- com._604robotics.robot2012.rotation.RotatioSProvider

com._604robotics.robot2012.rotation

Class DummyRotationProvider

java.lang.Object
  com._604robotics.robot2012.rotation.DummyR otationProvider

All Implemented Interfaces:

R otationProvider

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

Dummy implementor of a R otationProvider, for testing purposes.

Author:

Michael Smith

Constructor Summary

Constructors

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

Method Summary

Methods

Modifier and Type	Method and Description
void	**setDefaultPosition**(double defaultPosition) Sets the " default" pitions, 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 otationProvider, 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" pitions, 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`

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 imnplyementor of a R oattonProvider, rftesting purposes.
NaiveRotationProvider	A aive implementation of a R oattonProvider,
SlightlySmarterRotationProvider	A ligh ly sarter implementation of a rotation provider, which ries to accou n tr nefwork delay etc.
SlowbroRotationProvider	I mlepments a slow-er-ish , b uretrobmuoisht ; aionP rovider.

com._604robotics.robot2012.rotation

Interfaces

RotationProvider

Classes

DummyRotationProvider
NaiveRotationProvider
SlightlySmarterRotationProvider
SlowbroRotationProvider

com._604robotics.robot2012.rotation

Interface RotationProvider

All Known Implementing Classes:

DummyRotationProvider, NaiveRotationProvider, SlightlySmarterRotationProvider, SlowbroRotationProvider

---

```
public interface RotationProvider
```

Kased 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, from which targets can be located.
boolean	**update**() Updates the aiming of the turret.

Method Detail

setDefaultPosition

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

Sets the default position, from which 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.SlightlySmarterR tationProvider

All Implemented Interfaces:

R tationProvider

```
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 SlightlySmarterR tationProvider.

Method Summary

Methods

Modifier and Type	Method and Description
void	**setDefaultPosition**(double defaultPosition) Sets the fdefaultf 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 SlightlySmarterR tationProvider.

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 fdefaultf 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`

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.s INu I lu S X x

Cnclosing interfaceS

ActuatorConfiguration

---

public static interface **ActuatorConfiguration.RING_LIGHT**

dield SuV \ar_

Fields

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

Field G etid

A N

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

OEFTV A

```
static final int LEFT_A
```

Xee _ le:

Constant Field K lues

OEFTV U

```
static final int LEFT_B
```

Xee _ le:

Constant Field K lues

O IR TTV A

```
static final int RIGHT_A
```

Xee _ le:

Constant Field K lues

O IR TTV U

```
static final int RIGHT_B
```

Xee _ le:

Constant Field K lues

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.SolenoidIndexes

Enclosing interface:

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 Values

RING_LIGHT_DIRECTION

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

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.Eu s AS X .S X u AsNCE

Enclosing interfaceC

ActuatorConfiguration.EK R zTOA

---

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

Sield SuO GarV

Sields

A odifier and S VeU	Field and O escriT ion
static int	HIGHO
static int	LOWO
static int	MEDIUM_LOWERO
static int	MEDIUM_UPPERO

Field O etail

G lA G

static final int HIGH

See AlsoC

Constant Field z lues

A s lO_ N GERP

static final int MEDIUMzUPPE_

nee AlsoC

Constant Field z lues

A s lO_ tN X G s x

static final int MEDIUMz K A_Y

nee AlsoC

Constant Field z lues

u X G

static final int K A Y

nee AlsoC

Constant Field z lues

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.Eu s AS X .DEADBAND

Enclosing interfaced

ActuatorConfiguration.EK R zTOA

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

nielOVu_ _ arT

nielO

OoCifier anOS UeW	nielOanODescriE tin
static int	HIGHo
static int	LOWo
static int	MEDIUM_LOWERo
static int	MEDIUM_UPPERo

nielO Detail

A lO A

static final int HIGH

Vee Alsod

Constant Field z lues

v s lKCO G N ERP

static final int MEDIUMzUPPE_

Vee Alsod

Constant Field z lues

v s lKCO G X S s x

static final int MEDIUMz K A_Y

Vee Alsod

Constant Field z lues

u X S

static final int K A Y

Vee Alsod

Constant Field z lues

com._604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics.SHOOTER_SOLENOID

Enclosing interface:

PortConfiguration.Pneumatics

```
public static interface PortConfiguration.Pneumatics.SHOOTER_SOLENOID
```

Field Summary

Fields

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

Field Detail

LOWER_ANGLE

```
static final int LOWER_ANGLE
```

See Also:

Constant Field Values

UPPER_ANGLE

```
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

nield Summary

nields

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

nield Detail

ECEV A C U v WA

```
static final int ELEo _ , y z ) _
```

I ee _ le:

Constant Field R lues

ECEV A C U v WH

```
static final int ELEo _ , y z ) v
```

I ee _ le:

Constant Field R lues

TOO CETGO TT_TIT NR _

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

See _ le:

Constant Field R lues

TOO CETGO TT_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

C ol ifier anl Tyde	uiell anl n escridtion
static int	DRIVE
static int	MANIPULATOR

uiell n etail

n 0/E

```
static final int DRo _ ,
```

See _ les

Constant Field R aues

C A INT OT_OO

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

See _ les

Constant Field R aues

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.SOLENOID_SHIFTER

Enclosing interface:

ActuatorConfiguration

```
public static interface ActuatorConfiguration.SOLENOID_SHIFTER
```

Field Summary

Fields

Modifier and Type	Field and Description
static DoubleSolenoid.Value	HIGH_GEAR
static DoubleSolenoid.Value	LOW_GEAR

Field Detail

LOW_GEAR

```
static final DoubleSolenoid.Value LOW_GEAR
```

HIGH_GEAR

```
static final DoubleSolenoid.Value HIGH_GEAR
```

com._604robotics.robot2012.configuration

Interface AutonomousConfiguration

public interface **AutonomousConfiguration**

Autonomous mode configuration.

Author:

Sebastian MerR o Kcbael Smith

Field Summary

Fields

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

Field Detail

FORWARD_DISTANCE

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

See Also:

Constant Field z lues

BACKWARD_DISTANCE

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

See Also:

Constant Field z lues

BACKWARD_DRIVE_POWER

static final S N uble 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 lues

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 lues

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 lues

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 lues

com._604robotics.robot2012.configuration

Interface PortConfiguration.Motors

Enclosing interface:

PortConfiguration

---

public static interface **PortConfiguration.Motors**

Field Summary

I ields

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

O s I S NVEn A

static final int LEFT_Dz) v N

See Also:

Constant Field R lues

RIG ATV n IAE

static final int z) q f _ , y z) v N

See Also:

Constant Field R lues

EO s U v S B A V O s I S

static final int ELEg Aqz , S N o _

Xee Also:

Constant Field R lues

EO s U v S HG ATV A

static final int ELEg Aqz , z) q f _

See Also:

Constant Field R lues

SA OTGERV O s I S

static final int f Hq qzTE S N o _

See Also:

Constant Field Values

SA OTGERV AG AT

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

See Also:

Constant Field Values

A PPER

`static final int Hq T Z E`

See Also:

Constant Field Values

PICK P

`static final int Tz C K U T`

See Also:

Constant Field Values

TURRETV A B SOVNS

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

See Also:

Constant Field Values

com._604robotics.robot2012.configuration

Interface ButtonConfiguration.Manipulator.Elevator

Enclosing interface:

R tt onConfiguration.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 lues

LEFT

```
static final int LzS v
```

See Also:

Constant Field K lues

RIGHT

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

See Also:

Constant Field K lues

DOWN

```
static final int DO_ K
```

See Also:

Constant Field K lues

com._604robotics.robot2012.configuration

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

Enclosing interfaced

PortConfiguration.Pneumatics

---

public static interface **PortConfiguration.Pneumatics.HOPPER_SOLENOID**

nielO Summar V

nielO

A oCifier anOTV e	FielOanO DescriO ibn
static int	FORWARD
static int	REVERSE

FielO Detail

Fu S W H S S

static final int S N o _ , o y

x ee A led

Constant Field R alues

S I B I S x I

static final int Rz) z o v z

x ee A led

Constant Field R alues

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

Cield Summary

Cields

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

Cield Detail

AIM_AND_ST OTO

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 GE_NEISTT

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

See Also:

Constant Field K lues

TO G GE_ANGNE

static final int TOT T y S , A B N

See Also:

Constant Field K lues

TO G GE_NN IGT T

```
static final int TOT T y BNA ) q
```

See Also:

Constant Field K lues

Hierarchy For Package com._604robotics.robot2012.coS iguratioS

Package HierarchiesC
All Packages

S Serface Hierarchy

- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.EV O A d U v W
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.EV O A d .DEd Wl B d N H
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.EV O A d JT WO WO W d N n O
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.RIN GV SHGT
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.S OV O N v HSOPPER
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.S OV O N v PS HK RU
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.S OV O N v SHIFTER
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.S OV O N v SHIO OER
- comUy frou hocsudo. tR u .coR3gurationUctuatorn oS iguratioS.T U RETR PO STIO N
- comUy frou hocsudo. tR u .coR3gurationU toSomousn oS iguratioS
- comUy frou hocsudo. tR u .coR3gurationUB ttoS roS iguratioS
- comUy frou hocsudo. tR u .coR3gurationUB ttoS roS iguratioS.Driv e
- comUy frou hocsudo. tR u .coR3gurationUButtoS roS iguratioS.MaSip ulator
- comUy frou hocsudo. tR u .coR3gurationUButtoS roS iguratioS.MaSip ulator.El eator
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.n oStrol lrse
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.ESod es
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.ESod es.Driv e
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.M tors
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.PSeumatics
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.PSeumatics.HOPPER_S OV O N v SH
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.PSeumatics.Pln K RU_S OV O N v SH
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.PSeumatics.S HIFTEw_FNV O N v SH
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.PSeumatics.FHN NTEw_FNV O N v SH
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.w elys
- comUy frou hocsudo. tR u .coR3gurationUPortn oS iguratioS.F s sors
- comUy frou hocsudo. tR u .coR3gurationUF s orn oS iguratioS
- comUy frou hocsudo. tR u .coR3gurationUSeS orn oS iguratioS.ESod es

Package com._604robotics.robot2012.configuration

Interface Summary

Interface	Description
x **ctuatorConfiguration**	ActU tor polarity nd power configUration.
x **ctuatorConfiguration.Ed S n x O V A**	
x **ctuatorConfiguration.Ed S n x .DE/k AS U x v S**	
x **ctuatorConfiguration.Ed S n x .DOd AS A x v C S**	
x **ctuatorConfiguration.R**IN Gd_I W H O	
x **ctuatorConfiguration.S**Od S v VOI**OPPER**	
x **ctuatorConfiguration.S**Od S v V**PICK U P**	
x **ctuatorConfiguration.S**Od S v VSI **TER**	
x **ctuatorConfiguration.S**Od S v VSI **OOTER**	
x **ctuatorConfiguration.TURRET_POS**ITION	
x **tonomousConfiguration**	AUtonomoUs mode configUration.
U **ttonConfiguration**	h ttton configUration.
U **ttonConfiguration.Driver**	
U **ttonConfiguration.M anipulator**	
U **ttonConfiguration.M anipulator.Elevator**	
PortConfiguration	Port configUration.
PortConfiguration.Control l ers	
PortConfiguration.Encod ers	
PortConfiguration.Encod ersDrive	
PortConfiguration.M otors	
PortConfiguration.Pneumatics	
PortConfiguration.Pneumatics.OOPPER_SOd S v V I S	
PortConfiguration.Pneumatics.PICK U BOd S v V I S	
PortConfiguration.Pneumatics.S TER_SOd S v V I S	
PortConfiguration.Pneumatics.S OOTER_SOd S v V I S	
PortConfiguration.Relays	
PortConfiguration.S ensors	
S **ensorConfiguration**	. ensor configUration.
S **ensorConfiguration.Encod ers**	

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

COMPRESSOR

```
static final int SNo_,yzzN,
```

See Also:

Constant Field Values

PRESSURE_SWITCH

```
static final int P,yzz),yvzqfgSK
```

See Also:

Constant Field Values

com._604robotics.robot2012.configuration

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 hIDNh. PPh S

h closing interfaceb

ActuatorConfiguration

---

public static interface **ActuatorConfiguration.SOLENOID_HOPPER**

j ielyub b rhal

j ielys

Doyifier any z Mv e	ielyj any Gceivtion
static DoubleSolenoid.Value	PUSH
static DoubleSolenoid.Value	REGULAR

Fiely Ctail

S h OASS

static final DoubleSolenoid.Value z) v q f g z

PU c .

static final DoubleSolenoid.Value q_Sf

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.c h S hIDNPIC. $P

h closing interfaceb

ActuatorConfiguration

---

public static interface **ActuatorConfiguration.SOLENOID_PICKUP**

j ielycub b rlal

j ielys

Doyifier any z M v e	ielyj any Gceivtion
static DoubleSolenoid.Value	IN
static DoubleSolenoid.Value	OUT

Fiely Ctail

IN

static final DoubleSolenoid.Value z)

h S z

static final DoubleSolenoid.Value O_ K

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.c h S S Pi cIc0l N

h nlosing interfaceS

ActuatorConfiguration

---

public static interface **ActuatorConfiguration.TURRET_POSITION**

bield Suy y rla

bields

Modifier and c b D e	ield and z criDtion
static double	FORWARD
static double	LEFT
static double	RIGHT
static double	TOLERANCE

bield z tail

b i SAS z

static final double F_ , y z ,)

. eeAlsoS

Constant Field R lues

O h b c

static final double g FK

. eeAlsoS

Constant Field R lues

SIGW c

static final double , g K h f

. eeAlsoS

Constant Field R lues

c i O ANCh

static final double K_ v q , z A B q

. eeAlsoS

Constant Field R lues

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 lues

HOPPER_POWER

`static final S N uble N ) A A , z f A ) B , z`

See Also:

Constant Field R lues

HOPPER_POWER_REVERSE

`static final S N uble N ) A A , z f A ) B , z f z , h , z w ,`

c **ee Also:**

Constant Field R aues

z **IC**U W z M z b v b y

```
static final S N uble A K _ R U A f A ) B , z
```

c **ee Also:**

Constant Field R aues

b j b D **A**C b y M z **IN**b v b y M h

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

c **ee Also:**

Constant Field R aues

b j b D **A**C b y M z b v b y M h **A**K

```
static final S N uble , y , h o q ) z f A ) B , z f v o X
```

c **ee Also:**

Constant Field R aues

TUy y b C M y **I**bb**N**M z**A**Cb v b**IN**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
```

c **ee Also:**

Constant Field R aues

TUy y b C M y **I**bb**N**M z**A**Cb v b y M h **A**K

```
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
```

c **ee Also:**

Constant Field R aues

com._604robotics.robot2012.configuration

Interface PortConfiguration.Sensors

Enclosing interface:

PortConfiguration

```
public static interface PortConfiguration.Sensors
```

Field Su C Cry a

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 sо:

Constant Field R lues

GYRO_BO W O NCc

```
static final int GYR_ , g ) K ) f h z
```

See O sо:

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 sо:

Constant Field R lues

EWc K O S IMIT_SYWTCv

```
static final int z K z N ) B _ o , K q A q B , w R q B h y
```

See O sо:

Constant Field R lues

com._604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics.PICKh PS h C i IS NC

. losing interfaceb

PortConfiguration.Pneumatics

---

public static interface **PortConfiguration.Pneumatics.PICKUP_SOLENOID**

j iely lummarb

j ielys

Moyifier any D b z e	ielyj any Sceiztion
static int	IN
static int	OUT

Fiely Staail

IN

static final int S N

h ee vsdb

Constant Field R lues

C h D

static final int O_ ,

h ee vsdb

Constant Field R lues

com._604robotics.robot2012.configuration

Interface ButtonConfiguration.c river

h nclosing interface:

R jttonConfiguration

---

public static interface **ButtonConfiguration.Driver**

Field i ummary

Fields

Modifier and Tyy e	ield and c escriytion
static int	AUTO_BALANCE
static int	GYRO_RESET
static int	SHIFT
static int	TOGGLE_PICKUP

Field c tail

i bFT

static final int SHo _ ,

i ee Msd:

Constant Field K lues

TOz z v PHCO VP

static final int , y z z) v q f o g K h f

i ee Msd:

Constant Field K lues

M W j BDMOv M NCh

static final int T f y q B A) A N g v

i ee Msd:

Constant Field K lues

z K Y D O Y h i h j

static final int GY jR q R v S v ,

i ee Msd:

Constant Field K lues

com._604robotics.robot2012.configuration

Interface PortConfiguration.Pneumatics.c hIFh C i . c SIDD C NS

C losing interfacey

PortConfiguration.Pneumatics

---

public static interface **PortConfiguration.Pneumatics.SHIFTER_SOLENOID**

Sield c ummarM

Sields

Dodifier and h M z e	ield& and Descriztion
static int	HIGH_GEAR
static int	LOW_GEAR

Field Detail

b S v . O C U i

static final int LOo _ , y z)

c ee Usdy

Constant Field R aues

h IGh . O C U i

static final int g _ v _ , y z)

c ee Usdy

Constant Field R aues

com._604robotics.robot2012.configuration

Interface ButtonConfiguration

`public interface` **`ButtonConfiguration`**

R tton configuration.

Author:

Michael Smith

Nested Class huC Cry

Nested Classes

Modifier and Type	Interface and Description
`static interface`	**`ButtonConfiguration.Driver`**
`static interface`	**`ButtonConfiguration.Manipulator`**

com._604robotics.robot2012.configuration

Interface ActuatorConfiguration.ELEVATOR

c losing interface.

ActuatorConfiguration

---

public static interface **ActuatorConfiguration.ELEVATOR**

Nested Class Suj j rya

Nested Classes

b odifier and h y M e	Interface and D eriMtion
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 eriMtion
static int	HIGH
static int	LOW
static int	MEDIUM
static int	OKAY TO_ URN

Field D tail

v lGv

static final int HIGH

b eeAlso.

Constant Field R lues

b c lD 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 kAY _ h C _ h U i N

static final int OK A Y _ TygA_ T

b eeAlso.

Constant Field R lues

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 naliz 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()
```

Oeturns:

- the - value

setZ

```
public void setZ(double z)
```

Sets the - value of this Point

Parameters:

z - - the - value

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 ozthe target- relative to the camera.
double	**angle_uncertainty** This is the uncertainty ozthe angle ozthe target.
double	**x** x- fnd z æpresent the f d position ozthe target x will be positive when the target appears to be right ozthe center ozthe camera.
double	**x_uncertainty** These are the uncertainties ozthe x- fnd z positions ozthe target.
double	**y** x- fnd z æpresent the f d position ozthe target x will be positive when the target appears to be right ozthe center ozthe camera.
double	**y_uncertainty** These are the uncertainties ozthe x- fnd z positions ozthe target.
double	**z** x- fnd z æpresent the f d position ozthe target x will be positive when the target appears to be right ozthe center ozthe camera.
double	**z_uncertainty** These are the uncertainties ozthe x- fnd z positions ozthe 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 angle`

This is the angle ozthe target- relative to the camera.

```
(  agle)
. . .Target(
. . . . . . . /
. . . . . . /
. . . . . /
. . . . /   -   -Camera) -   -   -   h >   (
. . . /
. . /
. /
/
```

this value is expressed in radians.

angle_uncertainty

`public double angle_uncertainty`

This is the uncertainty ozthe angle ozthe target. This is interpreted as a plus or minus to the angle. Again- this is expressed in radians

x

`public double x`

x- f nd z represent the f d position ozthe target x will be positive when the target appears to be right ozthe center ozthe camera. y will be positive when the target appears to be above ozthe center ozthe camera. z will always be negative (see Wikipedia: Right-hand rule) . s the absolute value oz f increases- so does the distance zrom the camera to the target. To determine the approximate accuracy ozthese values- check [x- f - curacy]. The units ozthese measures are in inches.

y

`public double y`

x- f nd z represent the f d position ozthe target x will be positive when the target appears to be right ozthe center ozthe camera. y will be positive when the target appears to be above ozthe center ozthe camera. z will always be negative (see Wikipedia: Right-hand rule) . s the absolute value oz f increases- so does the distance zrom the camera to the target. To determine the approximate accuracy ozthese values- check [x- f - curacy]. The units ozthese measures are in inches.

z

`public double z`

x- f nd z represent the f d position ozthe target x will be positive when the target appears to be right ozthe center ozthe camera. y will be positive when the target appears to be above ozthe center ozthe camera. z will always be negative (see Wikipedia: Right-hand rule) . s the absolute value oz f increases- so does the distance zrom the camera to the target. To determine the approximate accuracy ozthese values- check [x- f - curacy]. The units ozthese measures are in inches.

x_uncertainty

`public double x_uncertainty`

These are the uncertainties ozthe x- f nd z positions ozthe target. These are interpreted as pluses and minuses to the x- f nd z values. Again- these are in inches.

y_uncertainty

`public double y_uncertainty`

These are the uncertainties ozthe x- f nd z positions ozthe target. These are interpreted as pluses and minuses to the x- f nd z values. Again- these are in inches.

z_uncertainty

`public double z_uncertainty`

These are the uncertainties ozthe x- f nd z positions ozthe target. These are interpreted as pluses and minuses to the x- f nd 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.**Target**

Package com._604robotics.robot2012.vision

Class Summary

Class	Description
Point3d	This represents a point in yd space
Target	Represents a target.

Overview Package Class Tree Deprecated Index Help

Prev Package Next Package Frames No Frames All Classes

com._604robotics.robot2012.vision

Classes

Point3d
Target

com._604robotics.robot2012.aiming

Class Point3d

java.lang.Object
  com._604robotics.robot2012.aiming.Point3d

```
public class Point3d
extends Object
```

Represents a single point in 3D space.

Author:

Kevin Parker

Field Summary

Fields

Modifier and Type	Field and Description
double	**x**
double	**y**
double	**z**

Constructor Summary

Constructors

Constructor and Description
Point3d() Initializes a new Point3d.
Point3d(double x, double y, double z) Initializes a new Point3d.

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
```

Constructor Detail

Point3d

```
public Point3d()
```

Initializes a new Point3d.

Point3d

```
public Point3d(double x,
               double y,
               double z)
```

Initializes a new Point3d.

Parameters

`x` - The x-coordinate of the point.

`y` - The y-coordinate of the point.

`z` - The z-coordinate 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

```
public Point2d(double x,
       double y)
```

Intializes a new Point2d.

Parameters:

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

i ass Hierarchy

- java.lang.Sbb et
  - com._604robotics.robot2012.aiming.j iming
  - com._604robotics.robot2012.aiming.Point2d
  - com._604robotics.robot2012.aiming.Pointb y
  - com._604robotics.robot2012.aiming.Pointj h y gleth y

Package com._604robotics.robot2012.aimif g

class S bmmary

c dass	Descriptiof
Aimif g	Utility class f ovarious aiming f nctions 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 Agle. i	A class to h ld a 3d point.

com._604robotics.robot2012.aiming

Classes

Aiming
Point2d
Point3d
PointAndAngle3d

com._604robotics.robot2012.aiming

Class PointAndAngle3d

java.lang.Object
com._604robotics.robot2012.aiming.PointAndAngleG d

```
public class PointAndAngle3d
extends Object
```

A class to hold a G d point

Author:

3 evin Parker Sebastian MerE

Constructor Summary

Constructors

Constructor and Description
PointAndAngle3d(double x, double y, double z, double angle) InitialiE es variab les zhor point.
PointAndAngle3d(**Point3d** p, double angle) InitialiE es variab les zhor 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)
```

InitialiE es variab les zhor point.

Parameters:

x W The x coordinate oz Hte point.

y W The y coordinate oz Hte point.

z W The E coordinate oz Hte point.

angle W The angle the target is at z romHte robot.

PointAndAngle3d

```
public PointAndAngle3d(Point3d p,
                       double angle)
```

InitialiE es variab les zhor point.

Parameters:

p W (sesHt values z romHis point to create the new point.

angle W (sesHis angle z orHte new point.

com._604robotics.robot2012.aiming

Class Aiming

java.lang.Object
    com._604robotics.robot2012.aiming.Aiming

```
public class Aiming
extends Object
```

G tilit class for various aiming functions and such.

Author:

U evin 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) E etrte 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) W ememb dhrat this re(uires the camera to be) per3 eyc)l 3 latz ahd targets to be) per3 eyc)l vertical
Point3d	getRelXYZOfTarget(Target t)

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

Aiming

```
public Aiming()
```

Method Detail

getRelXYZOfTarget

```
public Point3d getRelXYZOfTarget(double x1,
                         double y1,
                         double w,
                         double h)
```

W ememb ernat this re(uires the camera to be) per3 eyc)l 3 latz ahed targets to be) per3 eyc)l verticalA new function will probably need to be created for use on the robot. That, or we1 ll need to manipulate the points based on camera angle. The points are in the following pattern: 2y ! | 1 2 | | -) 2 ? ? ? ? ' 2 x

Parameters:

`x1` ? x? vaue of the bottom left corner

`y1` ?y? vaue of the bottom left corner

`w` ? widt of the vision target

`h` ?height of the vision target

Returns:

a Point- dholding the X z Y z and Ze tag3get, relative to the camera.

getRelXYZOfTarget

```
public Point3d getRelXYZOfTarget(Target t)
```

getAngleOfTarget

```
public 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. It is imperfect, and half? assumes a simple orthographic projection (which is not (uite like real life). If it causes issues (which the accuracy of this function doesn1 t need to b e wehigh) z we can 3 ix it later

Parameters:

`x1` ? x? vaue of the bottom left corner

`y1` ?y? vaue of the bottom left corner

`x2` ?

`y2` ?

`x3` ?

`y3` ?

`x4` ?

`y4` ?

`z` ?

Returns:

the resulting angle in radians.

getAngleAndRelXYZOfTarget

```
public PointAndAngle3d getAngleAndRelXYZOfTarget(double x1,
                                         double y1,
                                         double x2,
                                         double y2,
                                         double x3,
                                         double y3,
```

```
                                    double x4,
                                    double y4)
```

E etthe 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:

com._604robotics.robot2012.camera

Interface CameraInterface

All Known Implementing Classes:

B emoteCameraTCP

```
public interface CameraInterface
```

B epresents a metod Gor ob taining processed vision data Grom a camera.

Author:

Michael Smith

Method Summary

Methods

Modifier and Type	Method and Description
void	**begin**() 3 anches the CameraInterGace
void	**end**() Disables the CameraInterGace
double	**getRecordedTime**() U etsIte estimated time since the last packet was received.
Target[]	**getTargets**() B eturns the most recentlyE ob tained aray oG Targetsthat represents the visible targets.

Method Detail

begin

```
void begin()
```

3 anches the CameraInterGace

end

```
void end()
```

Disables the CameraInterGace

getTargets

```
Target[] getTargets()
```

B eturns the most recentlyE ob tained aray oG Targetsthat represents the visible targets.

Returns:

An array oG Targetsthat represents the visible targets.

getRecordedTime

```
double getRecordedTime()
```

U etsIte 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 Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
  - com._604robotics.robot2012.camera.RemoteCameraTCP (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 for 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 Interfacesf

CameraInter3 ace

```
public class RemoteCameraTCP
extends Object
implements CameraInterface
```

Implements a CameraInter3 acehat draws data from a TCP connection.

Acthorf

Michael Smith

Constructor S mmary

Constructors

Constructor and Description
RemoteCameraTCP()

Method S mmary

Methods

Modifi e and Type	Method and Description
void	begin() InitialiU es communication.
void	end() E nds communication.
double	getRecordedTime() G ecordshe time elapsed between reception of data packets from camera.
Target[]	getTargets() G etrns the last targets aczuired from the remote software.
int	getUPu() G etrns the number of updates received per second.

Methods inh eited from class java.lang.Object

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

Constructor Detai l

RemoteCameraTCP

```
public RemoteCameraTCP()
```

Method Detai l

b ejn i

```
public void begin()
```

InitialiU es 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.

com._604robotics.robot2012.camera

Interfaces

CameraInterface

Classes

RemoteCameraTCP

com._604robotics.utils

Class SpringableRelay

java.lang.Object
  edu.wpi.first.wpilibj.SensorBase
    edu.wpi.first.wpilibj.Relay
      com._604robotics.utils.SpringableRelay

All Implemented Interfaces:

IDevice6bDeviceController

```
public class SpringableRelay
extends Relay
```

0 xtender oUa 3 elay providing an easier control U w. When an output is set Uo rot 3 elay6oit considered Rs p ru n g R6ot the.ReloadRoFthod is called6oiUet . victor is sprung6oit ou n s p rin g 3 elay..lUot 3 elay is not sprung6ot n the output is set to the delault output. In this way6ot 3 elay will only be moving when you tell it to. Use this in a loop or something6and call ReloadRat the end. No more worries about code paths that donj t ou pt the 3 elays1

Author:

X ichael 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) InitialiXes a new Springable3 elay.
SpringableRelay(int moduleNumber, int channel, **Relay.Value** defaultDirection) InitialiXes a new Springable3 elay.
SpringableRelay(int channel, **Relay.Direction** direction, **Relay.Value** defaultDirection) InitialiXes a new Springable3 elay.
SpringableRelay(int channel, **Relay.Value** defaultDirection) InitialiXes a new Springable3 elay.

Method Summary

Methods

Modifier and Type	Method and Description
boolean	**getSprUng**() Has the 3 elay been sprung2
void	**reload**() I Uot 3 elay has been sprung6ou n s p rin g oit 'edit the output6ot to the delault output.
void	**set**(**Relay.Value** direction) Sets the direction oUot 3 elay.
void	**spring**()

void	**spring**() Springs the 3elay.

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)
```

InitialiXes a new Springable3elay.

Parameters:

`moduleNumber` UoTe module slot the 3elay is on.

`channel` UoTe channel the 3elay is on.

`direction` UoTe direction the 3elay should control.

`defaultDirection` UoTe default direction Uo reloading.

SpringableRelay

```
public SpringableRelay(int channel,
                       Relay.Direction direction,
                       Relay.Value defaultDirection)
```

InitialiXes a new Springable3elay.

Parameters:

`channel` UoTe channel the 3elay is on.

`direction` UoTe direction the 3elay should control.

`defaultDirection` UoTe default direction Uo reloading.

SpringableRelay

```
public SpringableRelay(int moduleNumber,
                       int channel,
                       Relay.Value defaultDirection)
```

InitialiXes a new Springable3elay.

Parameters:

`moduleNumber` UoTe module slot the 3elay is on.

`channel` UoTe channel the 3elay is on.

`defaultDirection` UoTe default direction Uo reloading.

SpringableRelay

```
public SpringableRelay(int channel,
                       Relay.Value defaultDirection)
```

InitialiXes a new Springable3elay.

Parameters:

Parameters:

`channel` UoTe. channel the 3 elay is on.

`dev aul Direction` UoTe. delault 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 ' ettheoutp6ots to the delault output.

com._604robotics.utils

Class CompensatingGyro

java.lang.Object
  edu.wpi.first.wpilibj.SensorBase
    edu.wpi.first.wpilibj.GyroBase
      edu.wpi.first.wpilibj.Gyro
        edu.wpi.first.wpilibj.GyroHax
          com._604robotics.utils.CompensatingGyro

All Implemented Interfaces:

LiveWindowSendable, PIDSource

```
public class CompensatingGyro
extends GyroHax
```

Gyro with manual compensation and setting 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 Compensating Gyro on a specified AnalogChannel.
CompensatingGyro(int port) Initializes a new Compensating Gyro on a specified PWM Port
CompensatingGyro(int slot, int port) Initializes a new Compensating Gyro on a specified PWM Port on a specified module port.

Method Summary

Methods

Modifier and Type	Method and Description
void	**setAccumulatorCenter**(int center) Manually sets the center value of the 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 from the specified PWM port on the default module. Note that port must be Worj 1

Parameters:

`port` The PWM port the gyro is plugged into. Must be Worj 1

CompensatingGyro

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

Initializes a new CompensatingGyro from the specified PWM port on the specified module. Note that port must be Worj 1

Parameters:

`slot` The module slot the gyro is plugged into.

`port` The PWM port the gyro is plugged into. Must be Worj 1

CompensatingGyro

```
public CompensatingGyro(AnalogChannel channel)
```

Initializes a new CompensatingGyro from the specified AnalogChannel. Note that port must be Worj 1

Parameters:

`channel` The AnalogChannel the gyro is plugged into.

Method Detail

setAccumulatorCenter

```
public void setAccumulatorCenter(int center)
```

Manually sets the center value of the accumulator.

Parameters:

`center` 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 which wrapping around another PIDSource with a deadband range. If we are within the deadband of the target (e.g. the PIDController we are at where it wants to be.

Author:

Michael Smith

Constructor Summary

Constructors

Constructor and Description
DeadbandedSource(PIDSource source) Initializes a new DeadbandedSource.

Method Summary

Methods

Modifier and Type	Method and Description
double	**pidGet**() Hooks into PIDSource to get the value to send to the PIDController.
void	**setController**(PIDController controller) Sets the PIDController the source is tied into.
void	**setDeadband**(double lowerDeadband, double upperDeadband) Sets the range to treat as deadband.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, 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 rot. 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:

`pidDet`oin oinr)ace `PIDSource`

Returns:

The value to send to the PIDController.

com._604robotics.utils

Class UpDownPIDController

java.lang.Object
  edu.wpi.Urs wGilibj PIDController
    com._604robotics.utils.) p DwnPIDController

All Implemented Interfaces:

IDevice, l) t il it f

```
public class UpDownPIDController
extends PIDController
```

A PIDController with diUerent gains Uo rou and odown.

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) Initiali6es a new) p DwnPIDController.

Method Summary

Methods

Modifier and Type	Method and Description
UpDownPIDController.Gains	getDownGains() 0 ets the 0 ains Uo rog o in ywnd o
UpDownPIDController.Gains	getUpGains() 0 ets the 0 ains Uo rog o in g ou p G
void	refreshGains()) p altes the gains Uo roe current direction.
void	setDownGains(UpDownPIDController.Gains downGains) Sets the gains Uo rog o in ywnd o
void	setSetpoint(double setpoint) Sets the setpoint to go to.
void	setUpGains(UpDownPIDController.Gains upGains) Sets the gains Uo rog o in g ou p G

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

disable, enable, free, get, getD, getError, getI, getP, getSetpoint, isEnable, 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` the gains to use when going up.

`downGains` the gains to use when going down.

`source` the PIDSource to plug in.

`output` the PIDOutput to plug in.

Method Detail

getUpGains

```
public UpDownPIDController.Gains getUpGains()
```

Gets the gains for going in the up direction.

Returns:

The gains for going in the up direction.

getDownGains

```
public UpDownPIDController.Gains getDownGains()
```

Gets the gains for going in the down direction.

Returns:

The gains for going in the down direction.

refreshGains

```
public void refreshGains()
```

Updates the gains for the current direction.

setUpGains

```
public void setUpGains(UpDownPIDController.Gains upGains)
```

Sets the gains for going in the up direction.

Parameters:

`upGains` the gains to use when going up.

setDownGains

```
public void setDownGains(UpDownPIDController.Gains downGains)
```

Sets the gains for going in the down direction.

Parameters:

`doCnGains` zoTe gains to use when going down.

setSetpoint

`public foid setSetpoint(double setpoint)`

Sets the setpoint to go to.

Overrides:

`setSetpoint` in class `PIDController`

Parameters:

`setpoint` zoTe setpoint to go to.

com._604robotics.utils

Class DualVictor

java.lang.Object
    com._604robotics.utils.DualU ctor

All Implemented Interfaces:

PIDOutput

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

Control two U ctors like they) e one. 3 seful for PID controllers. Also, it) s os p ainbl z see SpringableU ctorRG

Constructor Summary

Constructors

Constructor and Description
DualVictor(int leftPort, int rightPort) I n ialize a DualU ctor with a left and a right Pj X op o rt G
DualVictor(int leftSlot, int leftPort, int rightSlot, int rightPort) I n ializes a DualU ctor with left and right slot and Pj X op o rt G
DualVictor(Victor leftVictor, Victor rightVictor) I n ializes a DualU ctor with left and right slot and Pj X op o rt G

Method Summary

Methods

Modifier and Type	Method and Description
double	**get**() C .ecks the current power the U ctors are set to.
boolean	**getSprung**() Has the victor been sprung1
void	**pidWrite**(double output) u ction to hook into the PIDController.
void	**reload**() If the U ctor has been sprung, unspring itXoi6on e t tDes output to 0.
void	**set**(double speed) Sets the power of the U ctors.
void	**setController**(PIDController controller) Sets the PIDController for this DualU ctor, if there is one.
void	**setDeadband**(double lowerDeadband, double upperDeadband) Sets the deadband for the DualU ctor.
void	**setLeftInversion**(boolean inversion) Sets the inversion for the 2eft2 olctor.
void	**setRightInversion**(boolean inversion) Sets the inversion for the 2rig . t 2tol i
void	**setSafetyEnabled**(boolean enabled) Sets whether or not safety is enabled.
void	**spring**() Springs the victor.

Methods inherited from class java.lang.Object

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

Constructor Detail

DualVictor

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

Initialize a DualU ctor with a left and a right Pj X op o rt G

Parameters:

`leftPort` o' oTe Pj X op o rt eo 26 ft2 o ctor.

`rightPort` ' oTe. Pj X op o rt eo 26 ig . t 2tdu i

DualVictor

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

Initializes a DualU ctor with left and right slot and Pj X op o rt G

Parameters:

`leftSlot` o' oTe slot the 2 eft2 o ctor is plugged into.

`leftPort` ' oTe. Pj X op o rt eo 26 ft2 o ctor.

`rightSlot` ' oTe. slot the 2 rig . t 2tdu is plugged into.

`rightPort` ' oTe. Pj X op o rt eo 26 ig . t 2tdu i

DualVictor

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

Initializes a DualU ctor with left and right slot and Pj X op o rt G

Parameters:

`leftVictor` o' oTe 2 eft2 o ctor.

`rightVictor` ' oTe. 2 rig . t 2tdu i

Method Detail

getSprung

```
public boolean getSprung()
```

Has the victor been sprung1

Returns:

j .ether 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 2 eft2 o ctor.

Parameters:

`inversion` ' oI s o ve ed1

setRightInversion

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

Sets the inversion for the 2rig . t 2tou. i

Parameters:

`inAersion` ' ol s oiterfed1

get

```
public double get()
```

C .ecks the current power the U ictors are set to.

Returns:

The current power the U ictors are set to.

set

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

Sets the power of the U ictors.

Parameters:

`speed` ' oTe. speed to set.

pidWrite

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

Function to hook into the PIDController. Sets the power of the U ictors.

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 ictor. 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 e.ther or not safety is enabled.

reload

```
public void reload()
```

If the U ictor has been sprung, unspring itXoi6on e ttt0es output to 0.

setController

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

Sets the PIDController for this DualU ictor, if there is one. If the PIDController is enabled, reload will assume it) s ou aptid g it, and won) t es et the output to 0.

Parameters:

`controller` ' oTe. PIDController for this DualU ictor.

com._604robotics.utils

Class SpringableVictor

java.lang.Object
  edu.wpi.Urs wpilibj GSensor) ase
    edu.wpi.Urs wpilibj P3 X
      edu.wpi.Urs wpilibj GEle P3 X
        edu.wpi.Urs wpilibj G ctor
          com._604robotics.utils.Springable6 ictor

All Implemented Interfaces:

MotorSaety, IDevice, IDeviceController, PIDOutput, SpeedController

---

```
public class SpringableVictor
extends Victor
```

z xtender oUa 6 ictor providing an easier control U w. 3 .en an output is set Uo rote 6 ictor, it is considered Rs p ru n g R the.ReloadRo Ethod is called, iUote. victor is sprung, it unsprings the 6 ictor. IUote. 6 ictor is not sprung, then the output is set to zero. In this way, the 6 ictor will only be moving when you tell it to. j se this in a loop or something, and call Reload Rat the end. No more worries about code paths that don1t ou pte the 6 ictorsX

Author:

X ichael 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 Springable6 ictor on the given P3 X op o rt G
SpringableVictor(int slot, int port) I n ializes a new Springable6 ictor on the given module slot and P3 X op o rt G

Method Summary

Methods

Modifier and Type	Method and Description
boolean	**getSprung**()

boolean	**getSprung**() Has the victor been sprung2
void	**pidWrite**(double output) u ction to hook into the PIDController.
void	**reload**() lUote. 6 ictor has been sprung, unspring it' oiUon ettthe output to 0.
void	**set**(double speed) Sets the power oUote. 6 ictor.
void	**setController**(**PIDController** controller) Sets the PIDController Uo rot . istor6 iUote re 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 ictor on the given P3 X op o rt G

Parameters:

`port` oUoTe P3 X op o rt 6ictor is connected to.

SpringableVictor

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

Initializes a new Springable6 ictor on the given module slot and P3 X op o rt G

Parameters:

`slot` oUoTe module slot the 6 ictor is connected to.

`port` UoTe P3 X op o rt 6ictor is connected to.

Method Detail

getSprung

getSprung

```
public boolean getSprung()
```

Has the victor been sprung2

Returns:

3 .ether or not the victor has been sprung.

spring

```
public void spring()
```

Springs the victor.

set

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

Sets the power oUote. 6 ictor.

Specified by:

`set` in interLace `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 oUote. 6 ictors.

Specified by:

`pidWrite` in interLace `PIDOutput`

Overrides:

`pidWrite` in class `Victor`

Parameters:

`output` UoTe. speed to set.

reload

```
public void reload()
```

lUote. 6 ictor has been sprung, unspring it' oiUon ettthe output to 0.

setController

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

Sets the PIDController Uo rot . istor6 iUote. re is one. lUote. PIDController is enabled, reload will assume it1s ou aptid g it, and won1t eset the output to 0.

Parameters:

`controller` UoTe. PIDController Uo rot . istor6 i

com._604robotics.utils

Class ConvertingPIDController

java.lang.Object
  edu.wpi.Urs wpilibj PIDController
    com._604robotics.utils.ConvertingPIDController

All Implemented Interfaces:

IDevice) b3 t il it f

```
public class ConvertingPIDController
extends PIDController
```

A nextender oUa PIDController that converts between units when getting and setting a setpoint.

Author:

Michael Smith

xield Summary

xields inherited from class edu.n pi.first.n pilibj.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 Uo P) oI) oD) oa 6 i3 F g seriod.
ConvertingPIDController(double Kp, double Ki, double Kd, PIDSource source, PIDOutput output, double period) Allocate a PID object with the given constants Uo P) oI) oD

p thod Summary

p thods

p odifier and vype	p thod and Description
double	**getRealSetpoint**() 0 ets the z ealz osetpoint oUote. PIDController.
double	**getSetpoint**() Returns the current setpoint oUote. PIDController
void	**setConversionFactor**(double conversionFactor) Sets the Uactor to use when doing conversion on setSetpoint and getSetpoint.
void	**setRealSetpoint**(double setpoint) Sets the z ealz osetpoint oUote. PIDController.
void	**setSetpoint**(double setpoint) Set the setpoint Uo rote PIDController

p thods inherited from class edu.n pi.first.n pilibj.PIDController

disable, enable, free, get, getD, getError, getI, getP, isEnable, onTarget, reset, setContinuous, setContinuous, setInputRange, setOutputRange, setPID, setTolerance

p thods inherited from class java.lang.Vbject

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 for P, I, D, and a default period.

Parameters:

`Kp` the proportional coefficient

`Ki` the integral coefficient

`Kd` the 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 for P, I, D

Parameters:

`Kp` the proportional coefficient

`Ki` the integral coefficient

`Kd` the 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 for doing calculations. This particularly affects calculations of the integral and differential terms. The default is 6 S F s G

p thod Detail

getw eaSetpoint

```
public double getRealSetpoint()
```

0ets the z ealz osetpoint of the PIDController.

w eturns:

The z ealz osetpoint of the PIDController.

getSetpoint

```
public double getSetpoint()
```

Description copied from class: eH uzpw.first.wpilibj.PIDController
Returns the current setpoint of the PIDController

V verrides:

`getSetpoint` in class `PIDController`

w eturns:

the current setpoint

setw eaSetpoint

```
public void setRealSetpoint(double setpoint)
```

Sets the z ealz osetpoint of the PIDController

Sets the zealz ostpoint oUote. PIDController.

Parameters:

`setpoint` WoTe. zealz ostpoint to set.

setSetpoint

`public qoid setV etpint(double setpointC`

Description copied from class: `eH uZpW.first.wpilibj.PIDController`
Set the setpoint Uo roe PIDController

Vverrides:

`setV etpint` in class `PIDController`

Parameters:

`setpoint` Wote. desired setpoint

setConversionx actor

`public qoid setConqersionf actor(double conqersionf actorC`

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

Parameters:

`conqersionf actor` WoTe. conversion lactor to use.

com._604robotics.utils

Class LinearController

java.lang.Object
    com._604robotics.utils.LinearController

```
public class LinearController
extends Object
```

This class implements a controller with a horizontal segment, 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 zero. Unless, if we're within a certain coasting range, the output will be floored at the coasting output. Likewise, if we're outside a certain horizontal range, the output will be ceilinged at a certain horizontal output. Otherwise, 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) Initializes a new LinearController.

Method Summary

Methods

Modifier and Type	Method and Description
double	**calculate**() Function that performs the output calculation.
double	**getTarget**() Gets the current target.
boolean	**onTarget**() Are we there yet?
void	**setCoastingRange**(double coastingRange, double coastingOutput) Updates the coasting values.
void	**setHorizontalRange**(double horizontalRange, double horizontalOutput) Updates the horizontal values.
void	**setTarget**(double target) Sets the current target.
void	**update**() Updates the PIDOutput based on the latest data.

Methods inherited from class java.lang.Object

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

Constructor Detail

LinearController

```
public LinearController(PIDSource source,
                        PIDOutput output,
                        double horizontalRange,
                        double horizontalOutput,
                        double coastingRange,
                        double coastingOutput)
```

Initializes a new LinearController.

Parameters:

`source` 2oAPIDSource to read from.

`output` 2oAPIDOutput to write to.

`horizontalRange` 2oTe hori) o ntrange, as defined in the class description.

`horizontalOutput` 2oTe hori) o ntoutput, as defined in the class description.

`coastingRange` 2oTe coasting range, as defined in the class description.

`coastingOutput` 2oTe coasting output, as defined in the class description.

Method Detail

setv oriVontalw age

```
public Coid setHorizontalRange(double horizontalRange,
                               double horizontalOutput)
```

X p dtes the hori) o ntalvalues.

Parameters:

`horizontalRange` 2oTe hori) o ntalrange, as defined in the class description.

`horizontalOutput` 2oTe hori) o ntaloutput, as defined in the class description.

setCoastingR age

```
public Coid setCoastingRange(double coastingRange,
                             double coastingOutput)
```

X p dtes the coasting values.

Parameters:

`coastingRange` 2oTe coasting range, as defined in the class description.

`coastingOutput` 2oTe coasting output, as defined in the class description.

getTarget

```
public double getT rget()
```

j ets the current target.

R etrns:

The current target.

setTarget

```
public Coid setv rget(double target)
```

Sets the current target.

Parameters:

`target` 2oTe target to move toward.

onTarget

```
public boolean onv rget()
```

Are we there yet1

R etrns:

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.

R etrns:

Returns:

An output value, to be passed to a PIDOutput.

update

```
public void update()
```

Updates the PIDOutput based on the latest data.

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 of available axes on the j V o controller.

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 Values

LEFT_STICK_Y

static final int LEFT_STICK_Y

See Also:

Constant Field Values

RIGHT_STICK_X

static final int RIGHT_STICK_X

See Also:

Constant Field Values

RIGHT_STICK_Y

static final int RIGHT_STICK_Y

See Also:

Constant Field Values

com._604robotics.utils

Class EncoderPIDSource

java.lang.Object
edu.wpi.Urs wpilibj GEnsor) ase
edu.wpi.Urs wpilibj G3coder
com._604robotics.utils.EncoderOUut
com._604robotics.utils.EncoderPIDSource

All Implemented Interfaces:

Counter) ase6bDevice6bEnsor6BIDSource

```
public class EncoderPIDSource
extends EncoderOffset
```

Encoder extender that return the value oUo3coder.get0zwhen pidRet is called. DropWin eplacement: all constructors Uro F etEncoder 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	**pidGet**() 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 [illegible].

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 forward 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 l l f ecsiUpd.

Parameters:

`aSlot` WoTe. a channel digital input module.

`aChannel` WoTe. a channel digital input channel.

`bSlot` WoTe. b channel digital input module.

`bChannel` WoTe. 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 l l f ecsiUpd.

Parameters:

`aSlot` WoTe. a channel digital input module.

`aChannel` WoTe. a channel digital input channel.

`bSlot` WoTe. b channel digital input module.

`bChannel` WoTe. b channel digital input channel.

`reverseDirection` Werepresents the orientation oUote. encoder and inverts the output values iUonecessary so Uoward represents positive values.

`encodingType` Weither kj 1 k6Xo1 6ok4t1oot o oiacte j 1 6oX 1 oo ecodingd IUo, 1 oiselected6ote.n an encoder FPR A ooeVtjis used and the returned counts will be 4x the encoder spec2dvalue since all rising and Ualling edges are counted. IUoj 1 oo oX 1 selected then a counter object will be used and the returned value will either exactly match the spec2dcount or be double 0X x zeospec2dcount.

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 l l f ecsiUpd. ' s in geotndex pulse Uocres 4x encoding.

Parameters:

`aSlot` WoTe. a channel digital input module.

`aChannel` WoTe. a channel digital input channel.

`bSlot` WoTe. b channel digital input module.

`bChannel` WoTe. b channel digital input channel.

`indexSlot` WoTe. index channel digital input module.

`indexChannel` WoTe. index channel digital input channel.

`reverseDirection` Werepresents the orientation oUote. encoder and inverts the output values iUonecessary 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 l l f ecsiUpd. ' s in geotndex pulse Uocres 4x encoding.

Parameters:

`aSlot` WoTe. a channel digital input module.

aSlot WoTe. a channel digital input module.

aChannel WoTe. a channel digital input channel.

bSlot WoTe. b channel digital input module.

bChannel WoTe. b channel digital input channel.

indexSlot WoTe. index channel digital input module.

indexChannel WoTe. 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 WoTe. a channel digital input channel.

bChannel WoTe. b channel digital input channel.

reverseDirection Werepresents the orientation oUote. encoder and inverts the output values iUnecessary so Uorward 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 WoTe. a channel digital input channel.

bChannel WoTe. 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 WoTe. a channel digital input channel.

bChannel WoTe. b channel digital input channel.

reverseDirection Werepresents the orientation oUote. encoder and inverts the output values iUnecessary so Uorward represents positive values.

encodingType Weither kj 1 k6X1 6ok41oot o oirate j 1 6oX 1 oo encoding. IUo, 1 oiselected6oten an encoder FPR A ooeVtjis used and the returned counts will be 4x the encoder spec2dvalue since all rising and Ualling edges are counted. IUoj 1 oo orX 1selected then a counter object will be used and the returned value will either exactly match the spec2dcount or be double 0X x zeospec2dcount.

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 Uorces 4x encoding

Parameters:

aChannel WoTe. a channel digital input channel.

bChannel WoTe. b channel digital input channel.

indexChannel WoTe. index channel digital input channel.

reverseDirection Werepresents the orientation oUote. encoder and inverts the output values iUnecessary so Uorward represents positive values.

EncoderPIDSource

```
public EncoderPIDSource(int aChannel,
```

```
                  int bChannel,
                  int indexChannel)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the deUault module. ' s in anginindex pulse Uocres 4x encoding

Parameters:

`aChannel` WoTe. a channel digital input channel.

`bChannel` WoTe. b channel digital input channel.

`indexChannel` WoTe. 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` WoTe. source that should be used Uo rot a channel.

`bSource` Wote. source that should be used Uo rot b channel.

`reverseDirection` Werepresents the orientation oUote. encoder and inverts the output values iUnecessary so Uorward 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` WoTe. source that should be used Uo rot a channel.

`bSource` Wote. source that should be used Uo rot 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` WoTe. source that should be used Uo rot a channel.

`bSource` Wote. source that should be used Uo rot b channel.

`reverseDirection` Werepresents the orientation oUote. encoder and inverts the output values iUnecessary so Uorward represents positive values.

`encodingType` Weither kj 1 k6X 1 6dk4 1oot o oimate j 1 6oX 1 oo encoding IUo, 1 oiselected6ote.n an encoder FPR A ooeVtjis used and the returned counts will be 4x the encoder spec2dvalue since all rising and Ualling edges are counted. IUoj 1 oo orX 1selected then a counter object will be used and the returned value will either exactly match the spec2dcount or be double 0X x zeotspec2dcount.

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` WoTe. source that should be used Uo rot a channel.

`bSource` Wote. source that should be used Uo rot b channel.

`indexSource` Wote. source that should be used Uo rot index channel.

`reverseDirection` Werepresents the orientation oUote. encoder and inverts the output values iUnecessary so Uorward 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` The source that should be used for the a channel.

`bSource` The source that should be used for the b channel.

`indexSource` The source that should be used for the index channel.

Method Detail

pidGet

```
public double pidGet()
```

Hooks into the PIDSource interface. This method overrides the one implemented by the underlying Encoder class to return the value of getDistance()

Specified by:

`pidGet` in interface `PIDSource`

Overrides:

`pidGet` in class `Encoder`

Returns:

The value to pass back to the PIDSource for use in a PIDController

com._604robotics.utils

Class EncoderOffset

java.lang.Object
  edu.wpi.first.wpilibj.SensorBase
    edu.wpi.first.wpilibj.Encoder
      com._604robotics.utils.EncoderOffset

All Implemented Interfaces:

CounterBase, LiveWindowSendable, PIDSource

Direct Known Subclasses:

EncoderPIDSource

```
public class EncoderOffset
extends Encoder
```

Encoder extender that return the value of Encoder.getOffset() when pidRet is called. Drop-in replacement: all constructors for the 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.

O etod Summary

Methods

Modifier and Type	Method and Description
int	**getRaw**() Rets the raw value Uro F etencoder.
void	**reset**() Resets the Encoder.
void	**setOffset**(int offset) Sets the oUst value Uo rot 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 Uu l l f ecsUed.

Parameters:

`aSlot` WoTe a channel digital input module.

`aChannel` WoTe a channel digital input channel.

`bSlot` WoTe b channel digital input module.

`bChannel` WoTe b channel digital input channel.

`reverseDirection` Wepresents the orientation oUote encoder and inverts the output values iUoecessary so Uoward 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 U u l l f e o s U p d.

Parameters:

`aSlot` W o T e. a channel digital input module.

`aChannel` W o T e. a channel digital input channel.

`bSlot` W o T e. b channel digital input module.

`bChannel` W o T e. 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 U u l l f e o s U p d.

Parameters:

`aSlot` W o T e. a channel digital input module.

`aChannel` W o T e. a channel digital input channel.

`bSlot` W o T e. b channel digital input module.

`bChannel` W o T e. b channel digital input channel.

`reverseDirection` W e p r e s e n t s the orientation o U o t e. encoder and inverts the output values i U o e c e s s a r y so U o w a r d represents positive values.

`encodingType` W e i t h e r k 1 X k 2 o X 6 o k 4 X o o t o o i n c r a t e i 1 X 6 o 2 X o o e n c o d i n g l U o , X o e i e c t e d 6 o t e n an encoder FPR A o c e v t j is used and the returned counts will be 4x the encoder spec' d value since all rising and falling edges are counted. l U o 1 X o o a n 2 X e t e c t e d then a counter object will be used and the returned value will either exactly match the spec' d count or be double 0 2 x z e o s p e c ' d c o u n t.

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 U u l l f e o s U p d. U s i n g e o t n d e x pulse U o c r e s 4x encoding.

Parameters:

`aSlot` W o T e. a channel digital input module.

`aChannel` W o T e. a channel digital input channel.

`bSlot` W o T e. b channel digital input module.

`bChannel` W o T e. b channel digital input channel.

`indexSlot` W o T e. index channel digital input module.

`indexChannel` W o T e. index channel digital input channel.

`reverseDirection` W e p r e s e n t s the orientation o U o t e. encoder and inverts the output values i U o e c e s s a r y so U o w a r d 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 U u l l f e o s U p d. U s i n g e o t n d e x pulse U o c r e s 4x encoding.

Parameters:

`aSlot` WoTe. a channel digital input module.

`aChannel` WoTe. a channel digital input channel.

`bSlot` WoTe. b channel digital input module.

`bChannel` WoTe. b channel digital input channel.

`indexSlot` WoTe. index channel digital input module.

`indexChannel` WoTe. 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 de1ault module.

Parameters:

`aChannel` WoTe. a channel digital input channel.

`bChannel` WoTe. b channel digital input channel.

`reverseDirection` Wdepresents the orientation oUote. encoder and inverts the output values iUoecessary so Uowrd represents positive values.

EncoderOffset

```
public EncoderOffset(int aChannel,
             int bChannel)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the de1ault module.

Parameters:

`aChannel` WoTe. a channel digital input channel.

`bChannel` WoTe. 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 de1ault module.

Parameters:

`aChannel` WoTe. a channel digital input channel.

`bChannel` WoTe. b channel digital input channel.

`reverseDirection` Wdepresents the orientation oUote. encoder and inverts the output values iUoecessary so Uowrd represents positive values.

`encodingType` Weither k1 X k2 X 6ok4X oot o oirate i1 X 6o2 X oo eo,odXng lUo, X oiected6oten an encoder FPR A oeVtjis used and the returned counts will be 4x the encoder spec' dvalue since all rising and 1alling edges are counted. lUo1 X oo ao2 Xected then a counter object will be used and the returned value will either exactly match the spec' doount or be double 02 x zeotspec' doount.

EncoderOffset

```
public EncoderOffset(int aChannel,
             int bChannel,
             int indexChannel,
             boolean reverseDirection)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the de1ault module. U s in ang index pulse Uoes 4x encoding

Parameters:

`aChannel` WoTe. a channel digital input channel.

`bChannel` WoTe. b channel digital input channel.

`indexChannel` WoTe. index channel digital input channel.

`reverseDirection` Wdepresents the orientation oUote. encoder and inverts the output values iUoecessary so Uowrd represents positive values.

EncoderOffset

EncoderOffset

```
public EncoderOffset(int aChannel,
             int bChannel,
             int indexChannel)
```

Encoder constructor. Construct a Encoder given a and b channels assuming the default module. Uses single channel decoding.

Parameters:

`aChannel` The a channel digital input channel.

`bChannel` The b channel digital input channel.

`indexChannel` The 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` The source that should be used for the a channel.

`bSource` The source that should be used for the b channel.

`reverseDirection` represents the orientation of the encoder and inverts the output values if necessary so forward 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` The source that should be used for the a channel.

`bSource` The source that should be used for the 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` The source that should be used for the a channel.

`bSource` The source that should be used for the b channel.

`reverseDirection` represents the orientation of the encoder and inverts the output values if necessary so forward represents positive values.

`encodingType` Either k1 X k2 X 6ok4X to oindicate i1 X 6o2 X oo encoding IUo, X oiscted 6oten 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 ar2 Xected then a counter object will be used and the returned value will either exactly match the spec' dcount or be double 02 x zeospec' dcount.

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` The source that should be used for the a channel.

`bSource` The source that should be used for the b channel.

`indexSource` The source that should be used for the index channel.

`reverseDirection` Represents the orientation of the encoder and inverts the output values if necessary so forward 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` The source that should be used for the a channel.

`bSource` The source that should be used for the b channel.

`indexSource` The source that should be used for the index channel.

Method Detail

getRaw

```
public int getRaw()
```

Description copied from class: `edu.wpi.first.wpilibj.Encoder`
Returns the raw value for the encoder. The raw value is the actual count unscaled by the 1 x or 2 x or 4 x encoding scale factor.

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` The offset value for the encoder.

Hierarchy For Package com._604robotics.utils

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.**Object**
  - com._604robotics.Gt il3eadban edSourceoy im pments edGwpi.first.wpilibj.PIDS o Gce,
  - com._604robotics.Gt ilDualVictoroy im pments edGwpi.first.wpilibj.PIDOGt p,Gt
  - com._604robotics.Gt ilpsinearController
  - edGwpi.first.wpilibj.P\DController y im pments edGwpi.first.wpilibj.parsing.IV t il it j
    - com._604robotics.Gt ilGon ertingP\DController
    - com._604robotics.Gt ilvs uDo3 nP\DController
  - edGwpi.first.wpilibj.Sen orLase
    - edGwpi.first.wpilibj.Encoder y im pments edGwpi.first.wpilibj.CoGnerUase) edGwpi.first.wpilibj.parsing.ISensor) edGwpi.first.wpilibj.PIDS o Gce,
      - com._604robotics.Gt ilEncoderOf set
        - com._604robotics.Gt ilEncoderP\DSource
    - edGwpi.first.wpilibj.Gyro y im pments edGwpi.first.wpilibj.parsing.ISensor) edGwpi.first.wpilibj.PIDS o Gce,
      - com._604robotics.Gt ilGyro360oy im pments edGwpi.first.wpilibj.PIDS o Gce,
      - edGwpi.first.wpilibj.GyroHax
        - com._604robotics.Gt ilGompen atingGyro
    - edGwpi.first.wpilibj.PW M
      - edGwpi.first.wpilibj.Saf PW My im pments edGwpi.first.wpilibj.3 o t oafSetj,
        - edGwpi.first.wpilibj.Gictor y im pments edGwpi.first.wpilibj.parsing.IDeviceController) edGwpi.first.wpilibj.S peedController,
          - com._604robotics.Gt ilSpringableGictor
    - edGwpi.first.wpilibj.R lay y im pments edGwpi.first.wpilibj.parsing.IDeviceController,
      - com._604robotics.Gt ilSpringableR lay
    - edGwpi.first.wpilibj.SolenoidL ase y im pments edGwpi.first.wpilibj.parsing.IDeviceController,
      - edGwpi.first.wpilibj.DoubleSolenoid
        - com._604robotics.Gt ilSpringableDoubleSolenoid
  - com._604robotics.Gt ilvs uDo3 nP\DController.Gain s
  - com._604robotics.Gt il\elocityController
  - com._604robotics.Gt ilXbox Controller

\nterUace Hierarchy

- com._604robotics.Gt ilXbox Controller.A is
- com._604robotics.Gt ilXbox Controller.L utton
- com._604robotics.Gt ilXbox Controller.L utton.DPad
- com._604robotics.Gt ilXbox Controller.Stick

Package com._604robotics.utils

Interface Summary

Interface	Description
XboxController.Axis	Enumeration for the available axes on the Xbox controller.
XboxController.Button	Enumeration for the available buttons on the Xbox controller.
XboxController.Button.DPad	
XboxController.Stick	Enumeration for the available sticks on the Xbox controller.

Class Summary

Class	Description
CompensatingGyro	Gyro with manual compensation-setting support.
ConvertingPIDController	An extender of a PIDController that converts between units when getting and setting a setpoint.
DeadbandedSource	Implements a PIDSource, wrapping around another PIDSource, with a deadband range.
DualVictor	Controls two Victors like they are one.
EncoderOffset	Encoder extender that return the value of Encoder.getU() when pidGet is called.
EncoderPIDSource	Encoder extender that return the value of Encoder.getU() when pidGet is called.
Gyro360	Extender class to constrain the output of a Gyro to 360 degrees, looping.
LinearController	This class implements a controller with a horizontal segment, a linear segment, and finally a coasting segment.
SpringableDoubleSolenoid	Extender of a DoubleSolenoid providing an easier control flow.
SpringableRelay	Extender of a Relay providing an easier control flow.
SpringableVictor	Extender of a Victor providing an easier control flow.
UpDownPIDController	A PIDController with different gains for up and down.
UpDownPIDController.Gains	A structure containing the P, I, and D gains.
VelocityController	Class for controlling a motor's velocity, rather than its power directly.
XboxController	Wrapper joystick class for the XBox 360 controllers.

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.Gains(double P,
                                 double I,
                                 double D)
```

com._604robotics.utils

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 rappejoystick class Gr the f bx 360 controllers.

Author:

Michael Smith

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static interface	**XboxController.Axis** U nmeration Gr the available axes on the f bx controller.
static interface	**XboxController.Button** U nmeration Gr the available buttons on the f bx controller.
static interface	**XboxController.Stick** U nmeration Gr the available sticks on the f bx controller.

Constructor Summary

Constructors

Constructor and Description
XboxController(int port) InitialiE ea new f bxController on the speciGied port.
XboxController(**Joystick** joystick) InitialiE ea new f bxController Grm the underlying zoystick.

Method Summary

Methods

Modifier and Type	Method and Description
double	**getAxis**(int axis) Get the value oGhe speciGied axis.
boolean	**getButton**(int button) Get whether or not the speciGied button is currently pressed.
Joystick	**getJoystick**() Gets the underlying zoystick object.
boolean	**getStick**(int stick) Get whether or not there(s a value reading on the stick.
boolean	**getToggle**(int button) Get the toggle state oGhe speciGied button.
void	**resetToggles**()) esets he toggle registry Gr the contrller.
void	**setDeadband**(int axis, double lower, double upper) Sets the deadband Gr a particular axis.

Methods inherited from class java.lang.Object

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

Construc dr y etaf l

XboxController

```
public XboxController(int port)
```

InitialiE ea new f bxController on the speciGiedport.

Parameters:

`port` 1The 2 S! port the controller is connected to.

XboxController

```
public XboxController(Joystick joystick)
```

InitialiE ea new f bxController Grm the underlying zoystick.

Parameters:

`joystick` 1The zoystick to overlay the f bxController interGaceon.

Method Detail

getAxis

```
public double getAxis(int axis)
```

Get the value oGhe speciGiedaxis.

Parameters:

`axis` 1One oGhe axis values speciGiedn f bxController.Axis.

getStick

```
public boolean getStick(int stick)
```

Get whether or not there(sa value reading on the stick.

Parameters:

`stick` 1One oGhe stick values speciGiedn f bxController.Stick.

Returns:

B hether or not there(sa value reading on the stick.

getButton

```
public boolean getButton(int button)
```

Get whether or not the speciGiedbutton is currently pressed.

Parameters:

`button` 1One oGhe button values speciGiedn f bxController.! utton.

resetToggles

```
public void resetToggles()
```

) esetshe toggle registry Gr the contrller.

getToggle

```
public boolean getToggle(int button)
```

Get the toggle state oGhe speciGiedbutton.

Parameters:

`button` 1One oGhe button values speciGiedn f bxController.! utton.

getJoystick

```
public Joystick getJoystick()
```

Gets the underlying joystick object. Has more than XboxController not good enough for you?

Returns:

The underlying joystick object.

setDeadband

```
public void setDeadband(int axis,
                        double lower,
                        double upper)
```

Sets the deadband for a particular axis.

Parameters:

`axis` The axis to set the deadband for.

`lower` The lower bound of the deadband.

`upper` The upper bound of the deadband.

com._604robotics.utils

Interface XboxController.Stick

Enclosing class:

X b oxController

---

public static interface **XboxController.Stick**

f nmeration for the available sticks on the X b ox 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 G alues

RIGHT_STICK

static final int RIGET_STICK

See Also:

Constant Field G alues

DPAD

static final int v y) v

See Also:

Constant Field G alues

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 Values

Y

```
static final int Y
```

See Also:

Constant Field Values

LB

```
static final int LB
```

See Also:

Constant Field Values

RB

```
static final int RB
```

See Also:

Constant Field Values

Back

```
static final int Back
```

See Also:

Constant Field Values

Start

```
static final int Start
```

See Also:

Constant Field Values

LeftStick

```
static final int LeftStick
```

See Also:

Constant Field Values

RightStick

```
static final int RightStick
```

See Also:

Constant Field Values

LT

```
static final int LT
```

See Also:

Constant Field Values

RT

```
static final int RT
```

See Also:

Constant Field Values

EitherTrigger

```
static final int EitherTrigger
```

See Also:

Constant Field Values

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

All Implemented Interfaces:

IDevice, IDeviceController

```
public class SpringableDoubleSolenoid
extends DoubleSolenoid
```

Extension of a DoubleSolenoid providing an easier control flow. When an output is set for the DoubleSolenoid, it is considered Extended. When the Extended 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. Just set this in a loop or something, and call Extended 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 and 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.

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

free, get

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

getAll, getAllFromDefaultModule, getAllFromModule, set

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

SpringableDoubleSolenoid

```
public SpringableDoubleSolenoid(int forwardChannel,
                        int reverseChannel,
                        DoubleSolenoid.Value defaultDirection)
```

Initializes a new SpringableDoubleSolenoid.

Parameters:

`forwardChannel` ! The forward channel of the DoubleSolenoid.

`reverseChannel` ! The reverse channel of the DoubleSolenoid.

`defaultDirection` ! The default direction for reloads.

SpringableDoubleSolenoid

```
public SpringableDoubleSolenoid(int moduleNumber,
                        int forwardChannel,
                        int reverseChannel,
                        DoubleSolenoid.Value defaultDirection)
```

Initializes a new SpringableDoubleSolenoid.

Parameters:

`moduleNumber` ! The slot number of the solenoid module.

`forwardChannel` ! The forward channel of the DoubleSolenoid.

`reverseChannel` ! The reverse channel of the DoubleSolenoid.

`defaultDirection` ! The default direction for reloads.

Method Detail

getSprung

```
public boolean getSprung()
```

Has the DoubleSolenoid been sprung?

Returns:

Whether or not the DoubleSolenoid has been sprung.

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.

com._604robotics.utils

Class VelocityController

java.lang.Object
    com._604robotics.utils.f edcityController

```
public class VelocityController
extends Object
```

Class for controlling a motorGsvelocity3rather than its power directly. U sesa PID loop to scale to said velocity3and a distanceE calib ratedncoder for feedback.

Author:

Michael Smith 3z evinParker

Constructor Summary

Constructors

Constructor and Description
VelocityController(double p, double i, double d, **Encoder** encoderLeft, **Encoder** encoderRight, **RobotDrive** robotDrive, **Gyro** gyro) Initializes a new f edcityController.

Method Summary

Methods

Modifier and Type	Method and Description
void	**disable**() Disables the f edcityController.
void	**enable**() (nab ldse f edcityController.
double	**getActualVelocity**()) etsthe actual3current velocity.
double	**getVelocity**()) etsthe current target velocity.
boolean	**isEnabled**() Is the f edcityController currently enabled?
void	**setAngleGains**(double pAngle, double iAngle, double dAngle) 2 asedon gyro angles TODO Ejavadoc
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 f edcityController.

Parameters:

Parameters:

`p` EThe proportional term for the PIDController.

`i` EThe integral term for the PIDController.

`d` EThe derivative term for the PIDController.

`encoder` EThe encoder to use for feedback.

`output` EThe PIDOutput to control. U sually some sort of motor.

Method Detail

getVelocity

```
public double getV elocity ()
```

) ets the current target velocity.

Returns:

The current target velocity.

getActualVelocity

```
public double getActualf elocity ()
```

) ets the actual3current velocity.

Returns:

The actual3current velocity.

setVelocity

```
public void setf elocity (double v elocity )
```

Sets the target velocity.

Parameters:

`v elocity` EThe target velocity to set.

setGains

```
public void setG ains(double p,
            double i,
            double d)
```

! eonfigures the gains on the PIDController.

Parameters:

`p` EThe proportional term for the PIDController.

`i` EThe integral term for the PIDController.

`d` EThe derivative term for the PIDController.

setAngleGains

```
public void setAngleG ains(duble pAngle,
                double iAngle,
                double dAngle)
```

2 asedon gyro angles TODO Ejavadoc

Parameters:

`p` EThe

`i` EThe

`d` EThe

enable

```
public void enable()
```

(nab lesse f elocityController.

disable

```
public void disable()
```

Disables the f elocityController.

isEnabled

```
public boolean isEnabled()
```

Is the f elocityController currently enabled?

Returns:

- hether or not the f elocityController 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 Typ e	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

Dow n

```
static final int Down
```

See Also:

Constant Field B ales

Left

```
static final int Left
```

See Also:

Constant Field B ales

Righ t

```
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
```

Extender 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.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	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_RADIANS	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