ArduinoDistanceSensorLibrary

Generated by Doxygen 1.7.6.1

Mon May 21 2012 20:56:36

Contents

1 Class Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

2 Class Index

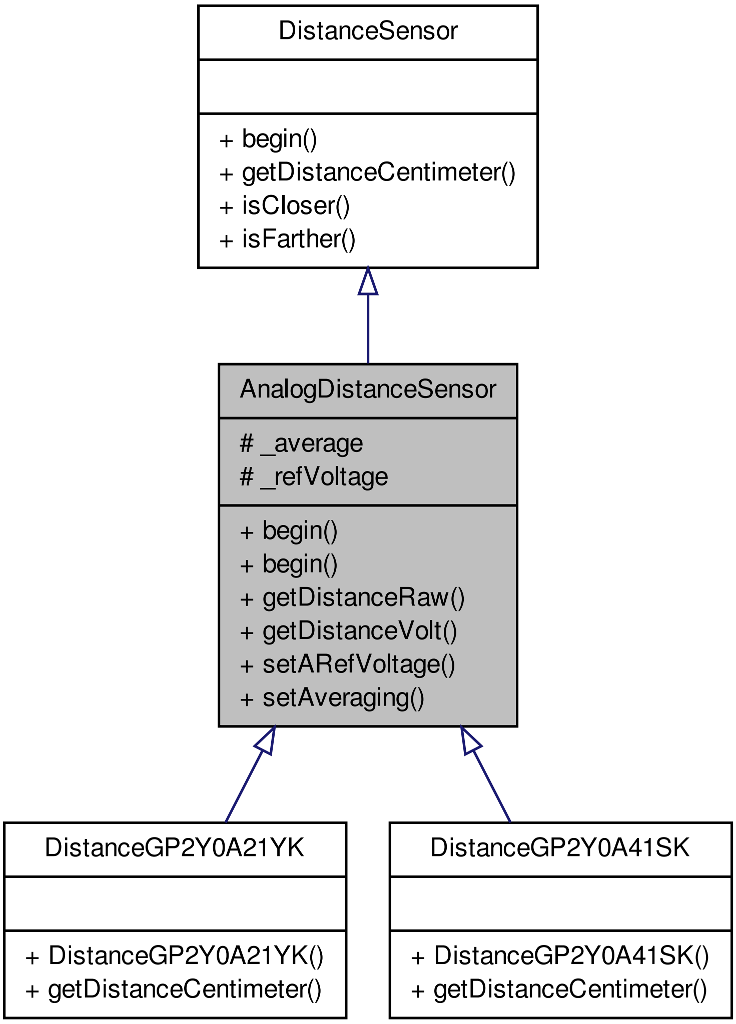
2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

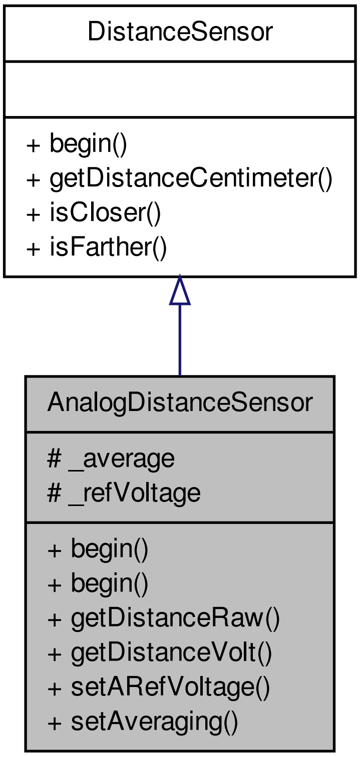
3 Class Documentation

3.1 AnalogDistanceSensor Class Reference

Inheritance diagram for AnalogDistanceSensor:



Collaboration diagram for AnalogDistanceSensor:



Public Member Functions

- void begin ()

  AnalogDistanceSensor.cpp - Library for retrieving data from Analog Distance sensors.
- void begin (int distancePin)

  Begin variables.
- int getDistanceRaw ()

  getDistanceRaw(): Returns the distance as a raw value: ADC output: 0 -> 1023
- int getDistanceVolt ()

  getDistanceVolt(): Returns the distance as a Voltage: ADC Input: 0V -> 5V (or 0V -> 3.3V)
- void setARefVoltage (int _refV)

  setARefVoltage:set the ADC reference voltage: (default value: 5V, set to 3 for external reference value, typically 3.3 on Arduino boards)

- void setAveraging (int avg)

  setAveraging(int avg): Sets how many samples have to be averaged (in getDistanceCentimeter), default value is 1. TODO: is this needed here?

Protected Attributes

- int **_average**
- int **_refVoltage**

3.1.1 Member Function Documentation

3.1.1.1 void AnalogDistanceSensor::begin ( ) [virtual]

AnalogDistanceSensor.cpp - Library for retrieving data from Analog Distance sensors.

Begin function to set input pins: distancePin = A0.

Reimplemented from DistanceSensor.

3.1.1.2 void AnalogDistanceSensor::begin (int *distancePin*)

Begin variables.

- int _distancePin: number indicating the distance to an object: ANALOG IN When you use begin() without parameters standard values are loaded: A0

3.1.1.3 int AnalogDistanceSensor::getDistanceRaw ( )

getDistanceRaw(): Returns the distance as a raw value: ADC output: 0 -> 1023

3.1.1.4 int AnalogDistanceSensor::getDistanceVolt ( )

getDistanceVolt(): Returns the distance as a Voltage: ADC Input: 0V -> 5V (or 0V -> 3.3V)

3.1.1.5 void AnalogDistanceSensor::setARefVoltage (int *refV*)

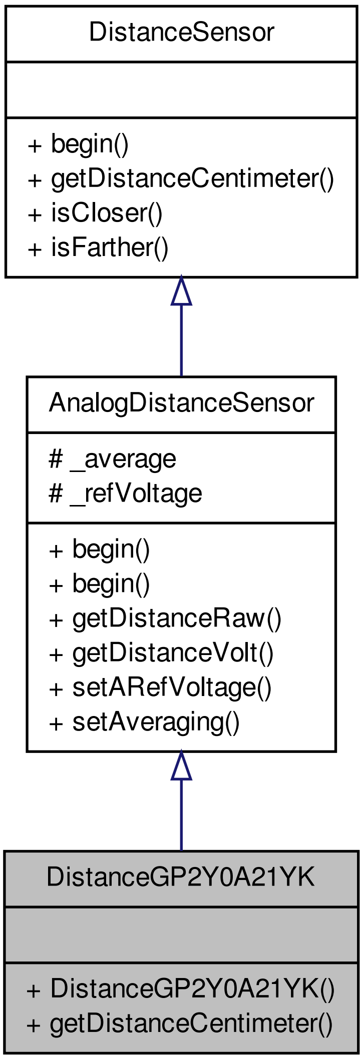
setARefVoltage:set the ADC reference voltage: (default value: 5V, set to 3 for external reference value, typically 3.3 on Arduino boards)

3.1.1.6 void AnalogDistanceSensor::setAveraging (int *avg*)

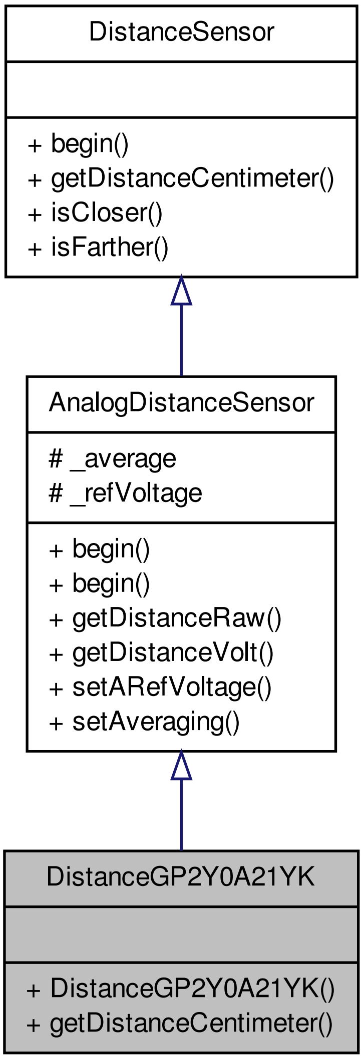
setAveraging(int avg): Sets how many samples have to be averaged (in getDistanceCentimeter), default value is 1. TODO: is this needed here?

3.2 DistanceGP2Y0A21YK Class Reference

Inheritance diagram for DistanceGP2Y0A21YK:



Collaboration diagram for DistanceGP2Y0A21YK:



Public Member Functions

- DistanceGP2Y0A21YK ()

  DistanceGP2Y0A21YK.cpp - Library for retrieving data from the GP2Y0A21YK IR - Distance sensor. For more information: variable declaration, changelog,... see -

DistanceGP2Y0A21YK.h.

- int getDistanceCentimeter ()

  getDistanceCentimeter(): Returns the distance in centimeters

3.2.1 Constructor & Destructor Documentation

3.2.1.1 DistanceGP2Y0A21YK::DistanceGP2Y0A21YK ( )

DistanceGP2Y0A21YK.cpp - Library for retrieving data from the GP2Y0A21YK IR - Distance sensor. For more information: variable declaration, changelog,... see - DistanceGP2Y0A21YK.h.

Constructor

3.2.2 Member Function Documentation

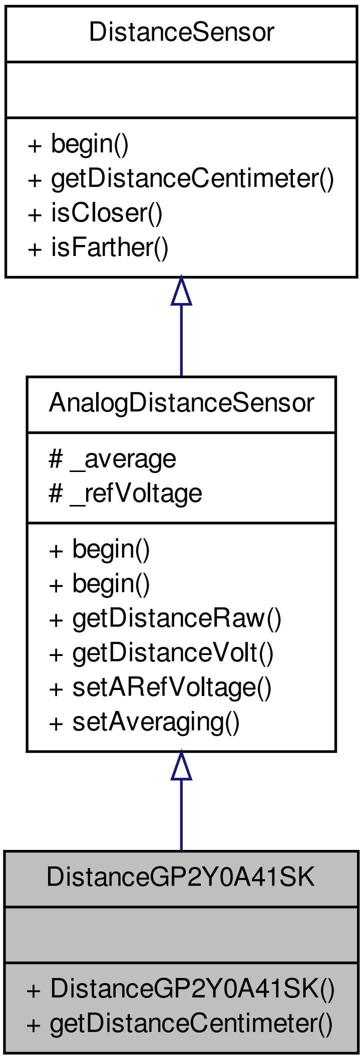
3.2.2.1 int DistanceGP2Y0A21YK::getDistanceCentimeter ( ) [virtual]

getDistanceCentimeter(): Returns the distance in centimeters

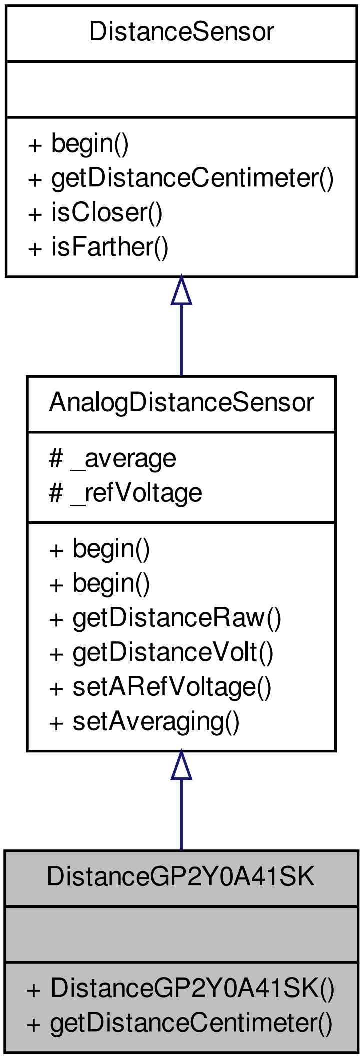
Implements DistanceSensor.

3.3 DistanceGP2Y0A41SK Class Reference

Inheritance diagram for DistanceGP2Y0A41SK:



Collaboration diagram for DistanceGP2Y0A41SK:



Public Member Functions

- DistanceGP2Y0A41SK ()

  DistanceGP2Y0A41SK.cpp - Library for retrieving data from the GP2Y IR Distance sensor. For more information: variable declaration, changelog,... see DistanceGP2-

Y0A41SK.h.

- int getDistanceCentimeter ()

  getDistanceCentimeter(): Returns the distance in centimeters: between 4-36cm (3 & 37 are boundary values)

3.3.1 Constructor & Destructor Documentation

3.3.1.1 DistanceGP2Y0A41SK::DistanceGP2Y0A41SK ( )

DistanceGP2Y0A41SK.cpp - Library for retrieving data from the GP2Y IR Distance sensor. For more information: variable declaration, changelog,... see DistanceGP2Y0A41-SK.h.

Constructor

3.3.2 Member Function Documentation

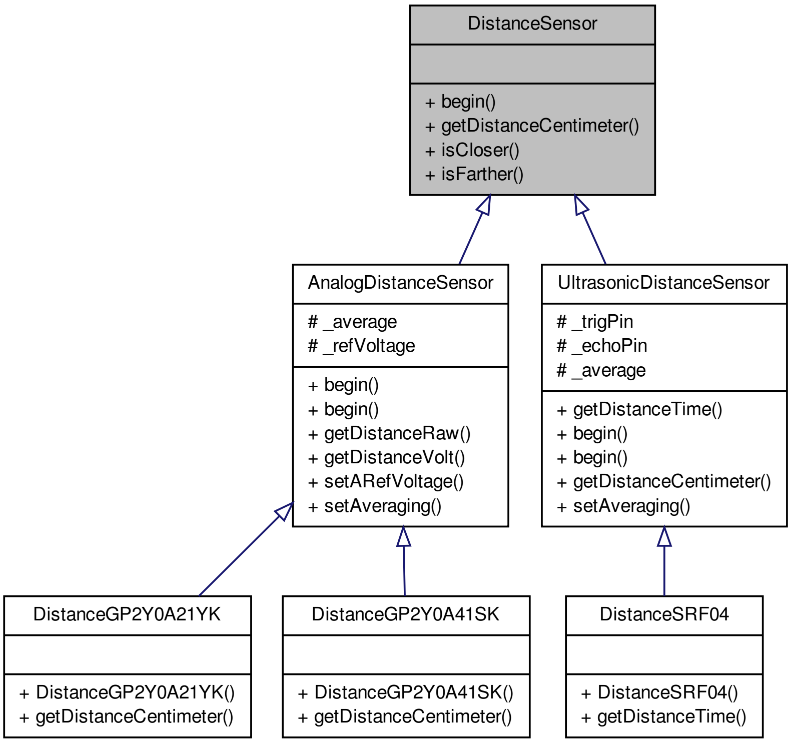
3.3.2.1 int DistanceGP2Y0A41SK::getDistanceCentimeter ( ) [virtual]

getDistanceCentimeter(): Returns the distance in centimeters: between 4-36cm (3 & 37 are boundary values)

Implements DistanceSensor.

3.4 DistanceSensor Class Reference

Inheritance diagram for DistanceSensor:



Public Member Functions

- virtual void **begin** ()
- virtual int **getDistanceCentimeter** ()=0
- boolean isCloser (int threshold)

  DistanceSensor.cpp - Library for retrieving data from Distance sensors.
- boolean isFarther (int threshold)

  isFarther: check whether the distance to the detected object is bigger than a given threshold

3.4.1 Member Function Documentation

3.4.1.1 boolean DistanceSensor::isCloser (int *threshold*)

DistanceSensor.cpp - Library for retrieving data from Distance sensors.

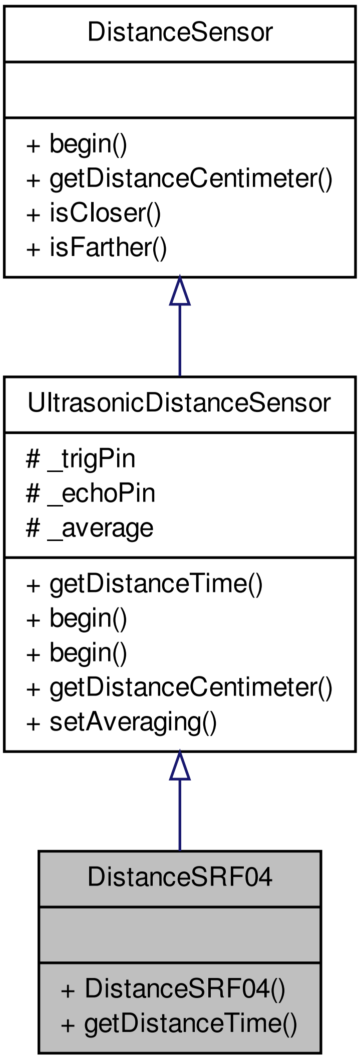
isCloser: check whether the distance to the detected object is smaller than a given threshold

3.4.1.2 boolean DistanceSensor::isFarther (int *threshold*)

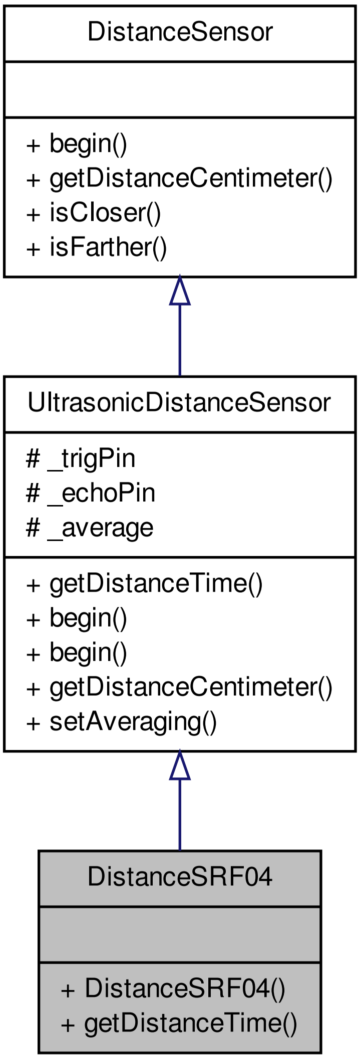
isFarther: check whether the distance to the detected object is bigger than a given threshold

3.5 DistanceSRF04 Class Reference

Inheritance diagram for DistanceSRF04:



Collaboration diagram for DistanceSRF04:



Public Member Functions

- DistanceSRF04 ()

  DistanceSRF04.cpp - Library for retrieving data from the GP2Y0A21YK IR Distance sensor. For more information: variable declaration, changelog,... see DistanceSR-

F04.h.

- int getDistanceTime ()

    getDistanceTime(): Returns the time between transmission and echo receive

3.5.1 Constructor & Destructor Documentation

3.5.1.1 DistanceSRF04::DistanceSRF04 ( )

DistanceSRF04.cpp - Library for retrieving data from the GP2Y0A21YK IR Distance sensor. For more information: variable declaration, changelog,... see DistanceSRF04.h.

Constructor

3.5.2 Member Function Documentation

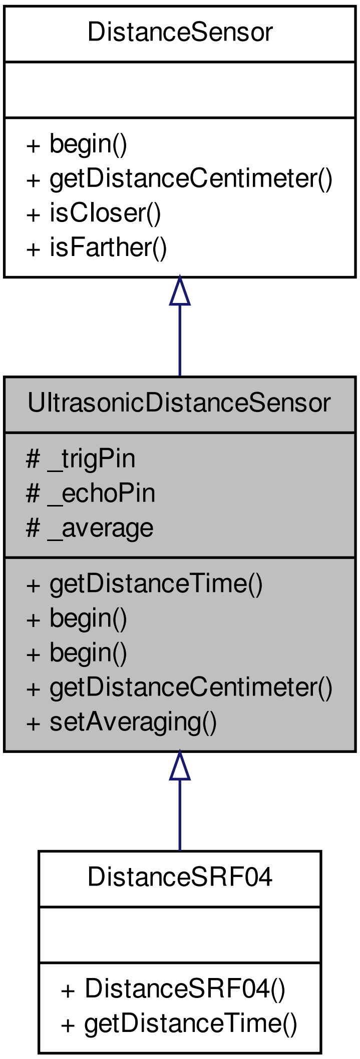
3.5.2.1 int DistanceSRF04::getDistanceTime ( ) [virtual]

getDistanceTime(): Returns the time between transmission and echo receive

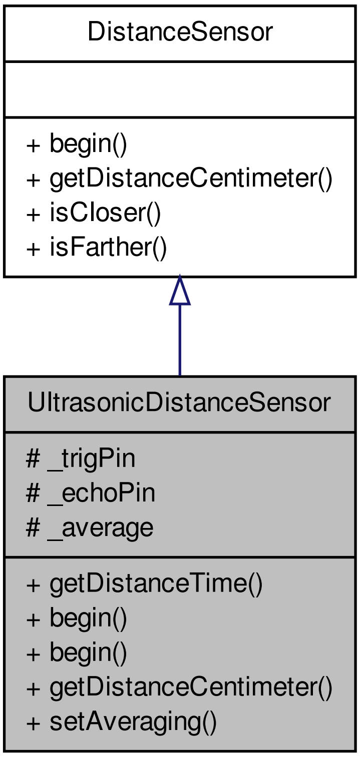
Implements UltrasonicDistanceSensor.

3.6 UltrasonicDistanceSensor Class Reference

Inheritance diagram for UltrasonicDistanceSensor:



Collaboration diagram for UltrasonicDistanceSensor:



Public Member Functions

- virtual int **getDistanceTime** ()=0
- void begin ()

  UltrasonicDistanceSensor.cpp - Library for retrieving data from the GP2Y0A21YK - IR Distance sensor. For more information: variable declaration, changelog,... see UltrasonicDistanceSensor.h.
- void begin (int echoPin, int trigPin)

  Begin variables.
- int getDistanceCentimeter ()

  getDistanceCentimeter(): Returns the distance in centimeters
- void setAveraging (int avg)

  setAveraging(int avg): Sets how many samples have to be averaged in getDistanceCentimeter, default value is 100.

Protected Attributes

- int **_trigPin**
- int **_echoPin**
- int **_average**

3.6.1 Member Function Documentation

3.6.1.1 void **UltrasonicDistanceSensor::begin** ( ) `[virtual]`

UltrasonicDistanceSensor.cpp - Library for retrieving data from the GP2Y0A21YK I-R Distance sensor. For more information: variable declaration, changelog,... see UltrasonicDistanceSensor.h.

Begin function to set default pins

Reimplemented from DistanceSensor.

3.6.1.2 void **UltrasonicDistanceSensor::begin** (int *echoPin,* int *trigPin*)

Begin variables.

- int trigPin: pin used to activate the sensor
- int echoPin: pin used to read the reflection

3.6.1.3 int **UltrasonicDistanceSensor::getDistanceCentimeter** ( ) `[virtual]`

getDistanceCentimeter(): Returns the distance in centimeters

Implements DistanceSensor.

3.6.1.4 void **UltrasonicDistanceSensor::setAveraging** (int *avg*)

setAveraging(int avg): Sets how many samples have to be averaged in getDistanceCentimeter, default value is 100.