

ArduinoDistanceSensorLibrary

Generated by Doxygen 1.7.4

Mon Jan 23 2012 14:39:38

Contents

1 Class Index	1
1.1 Class Hierarchy	1
2 Class Index	1
2.1 Class List	2
3 Class Documentation	2
3.1 AccousticDistanceSensor Class Reference	3
3.1.1 Member Function Documentation	5
3.2 AnalogDistanceSensor Class Reference	6
3.2.1 Member Function Documentation	8
3.3 DistanceGP2Y0A21YK Class Reference	10
3.3.1 Constructor & Destructor Documentation	12
3.3.2 Member Function Documentation	12
3.4 DistanceSensor Class Reference	13
3.4.1 Member Function Documentation	14
3.5 DistanceSRF04 Class Reference	15
3.5.1 Constructor & Destructor Documentation	17
3.5.2 Member Function Documentation	17

1 Class Index

1.1 Class Hierarchy

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

DistanceSensor	13
AccousticDistanceSensor	3
DistanceSRF04	15
AnalogDistanceSensor	6
DistanceGP2Y0A21YK	10

2 Class Index

2.1 Class List

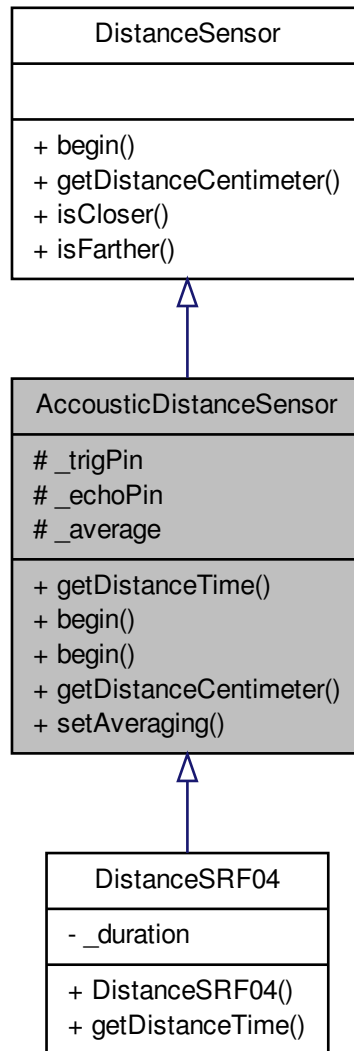
Here are the classes, structs, unions and interfaces with brief descriptions:

AccousticDistanceSensor	3
AnalogDistanceSensor	6
DistanceGP2Y0A21YK	10
DistanceSensor	13
DistanceSRF04	15

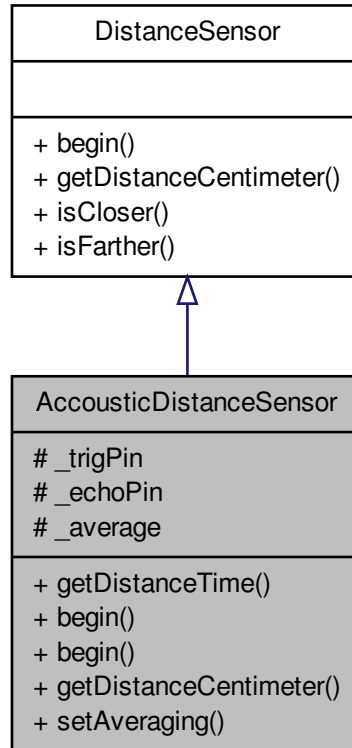
3 Class Documentation

3.1 AccousticDistanceSensor Class Reference

Inheritance diagram for AccousticDistanceSensor:



Collaboration diagram for AccousticDistanceSensor:



Public Member Functions

- virtual int **getDistanceTime** ()=0
- void **begin** ()
AccousticDistanceSensor.cpp - Library for retrieving data from the GP2Y0A21YK IR Distance sensor. For more information: variable declaration, changelog,... see [AccousticDistanceSensor.h](#).
- void **begin** (int echoPin, int trigPin)
Begin variables.
- int **getDistanceCentimeter** ()
getDistanceTime(): Returns the time between transmission and echo receive
- void **setAveraging** (int avg)
setAveraging(int avg): Sets how many samples have to be averaged in getDistance-Centimeter, default value is 100.

Protected Attributes

- `int _trigPin`
- `int _echoPin`
- `int _average`

3.1.1 Member Function Documentation

3.1.1.1 `void AccousticDistanceSensor::begin ()` [virtual]

AccousticDistanceSensor.cpp - Library for retrieving data from the GP2Y0A21YK IR Distance sensor. For more information: variable declaration, changelog,... see [AccousticDistanceSensor.h](#).

Constructor Begin function to set default pins

Reimplemented from [DistanceSensor](#).

3.1.1.2 `void AccousticDistanceSensor::begin (int echoPin, int trigPin)`

Begin variables.

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

3.1.1.3 `int AccousticDistanceSensor::getDistanceCentimeter ()` [virtual]

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

`getDistanceCentimeter()`: Returns the distance in centimeters

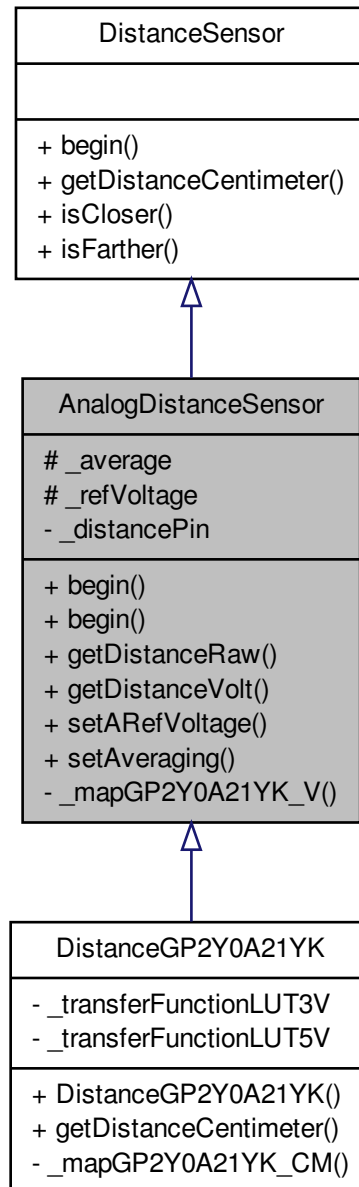
Implements [DistanceSensor](#).

3.1.1.4 `void AccousticDistanceSensor::setAveraging (int avg)`

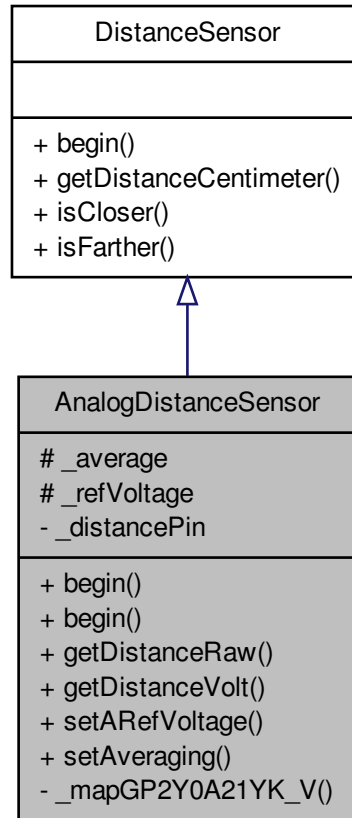
`setAveraging(int avg)`: Sets how many samples have to be averaged in `getDistanceCentimeter`, default value is 100.

3.2 AnalogDistanceSensor Class Reference

Inheritance diagram for AnalogDistanceSensor:



Collaboration diagram for AnalogDistanceSensor:



Public Member Functions

- void [begin](#) ()
AnalogDistanceSensor.cpp - Library for retrieving data from the GP2Y0A21YK IR Distance sensor. For more information: variable declaration, changelog,... see [AnalogDistanceSensor.h](#).
- 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.

Protected Attributes

- int `_average`
- int `_refVoltage`

3.2.1 Member Function Documentation

3.2.1.1 void AnalogDistanceSensor::begin () [virtual]

AnalogDistanceSensor.cpp - Library for retrieving data from the GP2Y0A21YK IR Distance sensor. For more information: variable declaration, changelog,... see [AnalogDistanceSensor.h](#).

Begin function to set pins: `distancePin = A0`.

Reimplemented from [DistanceSensor](#).

3.2.1.2 void AnalogDistanceSensor::begin (int `distancePin`)

Begin variables.

- int `_distancePin`: number indicating the distance to an object: ANALOG IN When you use [`begin\(\)`](#) without variables standard values are loaded: A0

3.2.1.3 int AnalogDistanceSensor::getDistanceRaw ()

[`getDistanceRaw\(\)`](#): Returns the distance as a raw value: ADC output: 0 -> 1023

3.2.1.4 int AnalogDistanceSensor::getDistanceVolt ()

[`getDistanceVolt\(\)`](#): Returns the distance as a Voltage: ADC Input: 0V -> 5V (or 0V -> 3.3V)

3.2.1.5 void AnalogDistanceSensor::setARefVoltage (int `refV`)

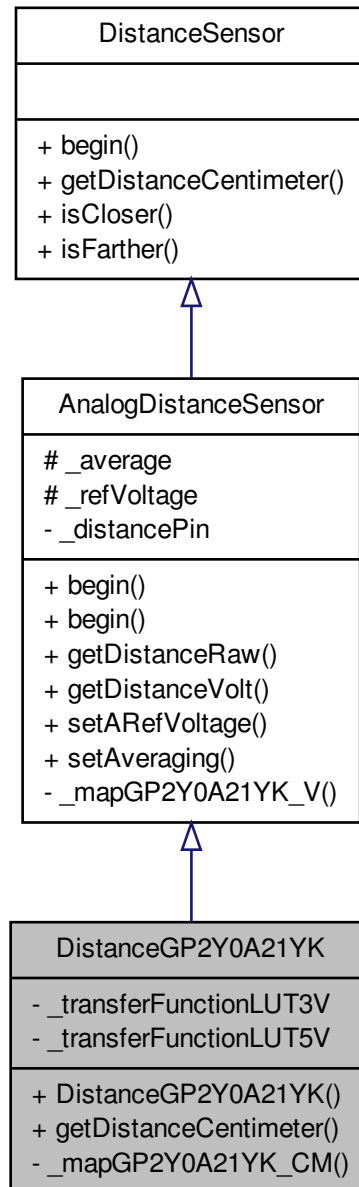
[`setARefVoltage`](#):set the ADC reference voltage: (default value: 5V, set to 3 for external reference value, typically 3.3 on Arduino boards)

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

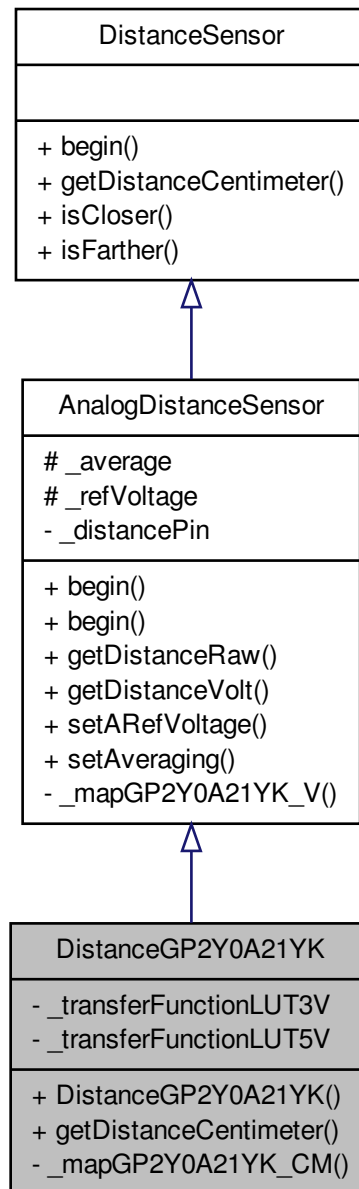
[setAveraging\(int avg\)](#): Sets how many samples have to be averaged in getDistance-Centimeter, default value is 1.

3.3 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 \(\)](#)
Begin function to set pins: distancePin = A0.

3.3.1 Constructor & Destructor Documentation

3.3.1.1 DistanceGP2Y0A21YK::DistanceGP2Y0A21YK ()

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

Constructor

3.3.2 Member Function Documentation

3.3.2.1 int DistanceGP2Y0A21YK::getDistanceCentimeter () [virtual]

Begin function to set pins: distancePin = A0.

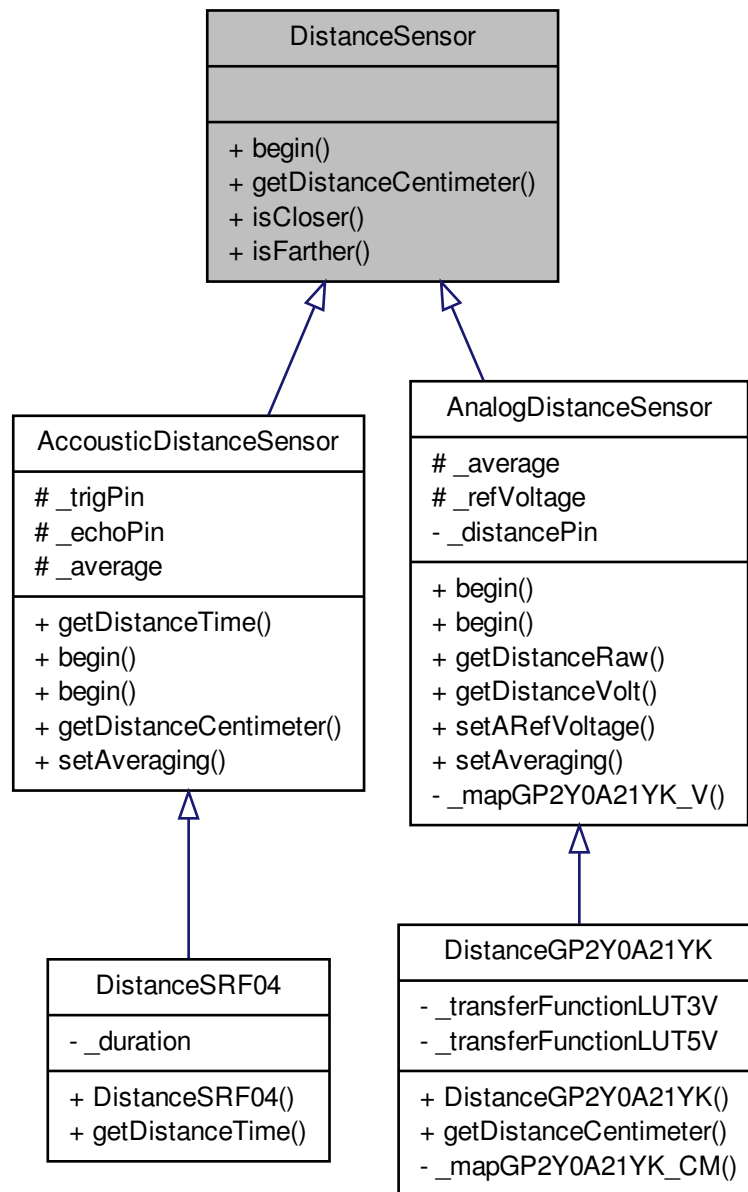
Begin variables

- `int _distancePin`: number indicating the distance to an object: ANALOG IN When you use [begin\(\)](#) without variables standard values are loaded: A0 [setAveraging\(int avg\)](#): Sets how many samples have to be averaged in [getDistanceCentimeter](#), default value is 1. [getDistanceRaw\(\)](#): Returns the distance as a raw value: ADC output: 0 -> 1023 [getDistanceVolt\(\)](#): Returns the distance as a Voltage: ADC Input: 0V -> 5V (or 0V -> 3.3V) [getDistanceCentimeter\(\)](#): Returns the distance in centimeters

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,...
- boolean **isFarther** (int threshold)
isFarther: check whether the distance to the detected object is smaller than a given threshold

3.4.1 Member Function Documentation

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

DistanceSensor.cpp - Library for,...

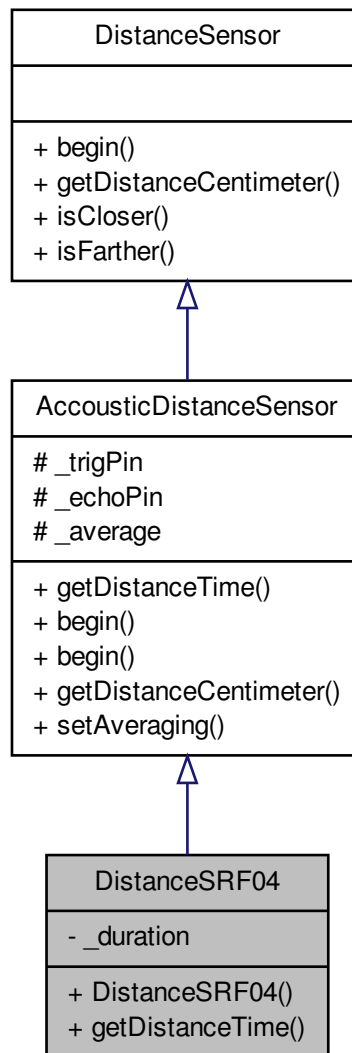
isCloser: check whether the distance to the detected object is smaller than a given threshold

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

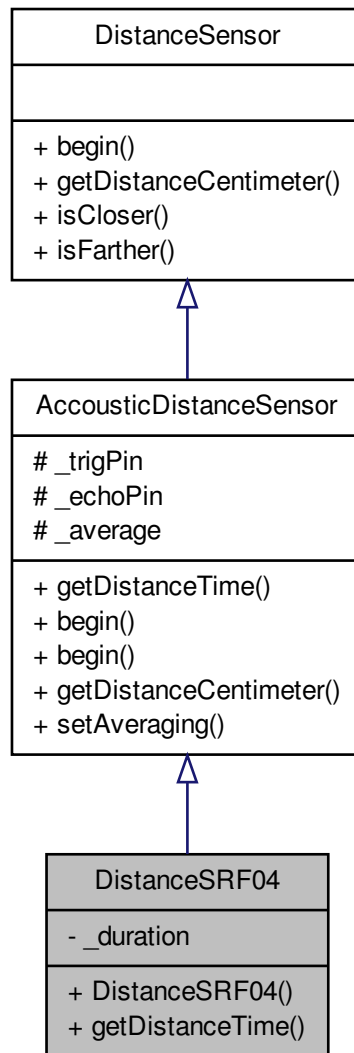
isFarther: check whether the distance to the detected object is smaller than a given threshold

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 [DistanceSRF04.h](#).

- [int getDistanceTime \(\)](#)
Begin function to set default pins.

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]

Begin function to set default pins.

Begin variables

- `int trigPin`: pin used to activate the sensor
- `int echoPin`: pin used to read the reflection [setAveraging\(int avg\)](#): Sets how many samples have to be averaged in `getDistanceCentimeter`, default value is 100.
[getDistanceTime\(\)](#): Returns the time between transmission and echo receive

Implements [AccousticDistanceSensor](#).

Index

- AccousticDistanceSensor, [2](#)
 - begin, [4](#)
 - getDistanceCentimeter, [4](#)
 - setAveraging, [4](#)
- AnalogDistanceSensor, [5](#)
 - begin, [7](#)
 - getDistanceRaw, [7](#)
 - getDistanceVolt, [7](#)
 - setARefVoltage, [7](#)
 - setAveraging, [7](#)
- begin
 - AccousticDistanceSensor, [4](#)
 - AnalogDistanceSensor, [7](#)
- DistanceGP2Y0A21YK, [9](#)
 - DistanceGP2Y0A21YK, [11](#)
 - getDistanceCentimeter, [11](#)
- DistanceSensor, [12](#)
 - isCloser, [13](#)
 - isFarther, [13](#)
- DistanceSRF04, [14](#)
 - DistanceSRF04, [16](#)
 - getDistanceTime, [16](#)
- getDistanceCentimeter
 - AccousticDistanceSensor, [4](#)
 - DistanceGP2Y0A21YK, [11](#)
- getDistanceRaw
 - AnalogDistanceSensor, [7](#)
- getDistanceTime
 - DistanceSRF04, [16](#)
- getDistanceVolt
 - AnalogDistanceSensor, [7](#)
- isCloser
 - DistanceSensor, [13](#)
- isFarther
 - DistanceSensor, [13](#)
- setARefVoltage
 - AnalogDistanceSensor, [7](#)
- setAveraging
 - AccousticDistanceSensor, [4](#)
 - AnalogDistanceSensor, [7](#)