ArduinoDistanceSensorLibrary

Generated by Doxygen 1.7.4

Mon Jan 23 2012 03:08:09

Contents

1 Class Index

1.1 Class Hierarchy

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

2 Class Index

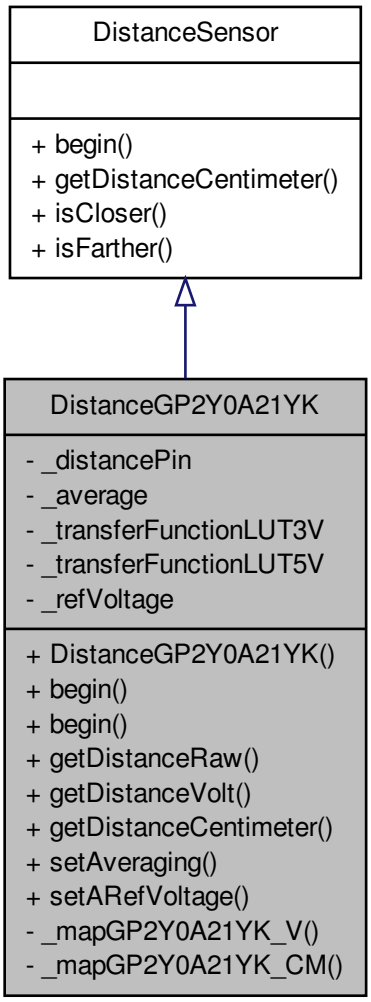
2.1 Class List

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

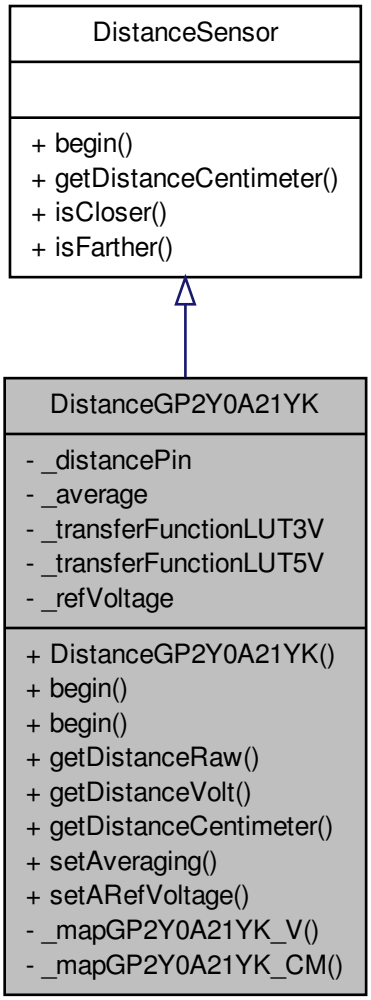
3 Class Documentation

3.1 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.
- void begin ()

Begin function to set pins: distancePin = A0.

- 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)
- 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 1.
- 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)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 DistanceGP2Y0A21YK::DistanceGP2Y0A21YK ( )

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

Constructor

3.1.2 Member Function Documentation

3.1.2.1 void DistanceGP2Y0A21YK::begin ( ) [virtual]

Begin function to set pins: distancePin = A0.

Reimplemented from DistanceSensor.

3.1.2.2 void DistanceGP2Y0A21YK::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

3.1.2.3 int DistanceGP2Y0A21YK::getDistanceCentimeter ( ) [virtual]

getDistanceCentimeter(): Returns the distance in centimeters

Implements DistanceSensor.

3.1.2.4 int DistanceGP2Y0A21YK::getDistanceRaw ( )

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

3.1.2.5 int DistanceGP2Y0A21YK::getDistanceVolt ( )

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

3.1.2.6 void DistanceGP2Y0A21YK::setARefVoltage (int *refV*)

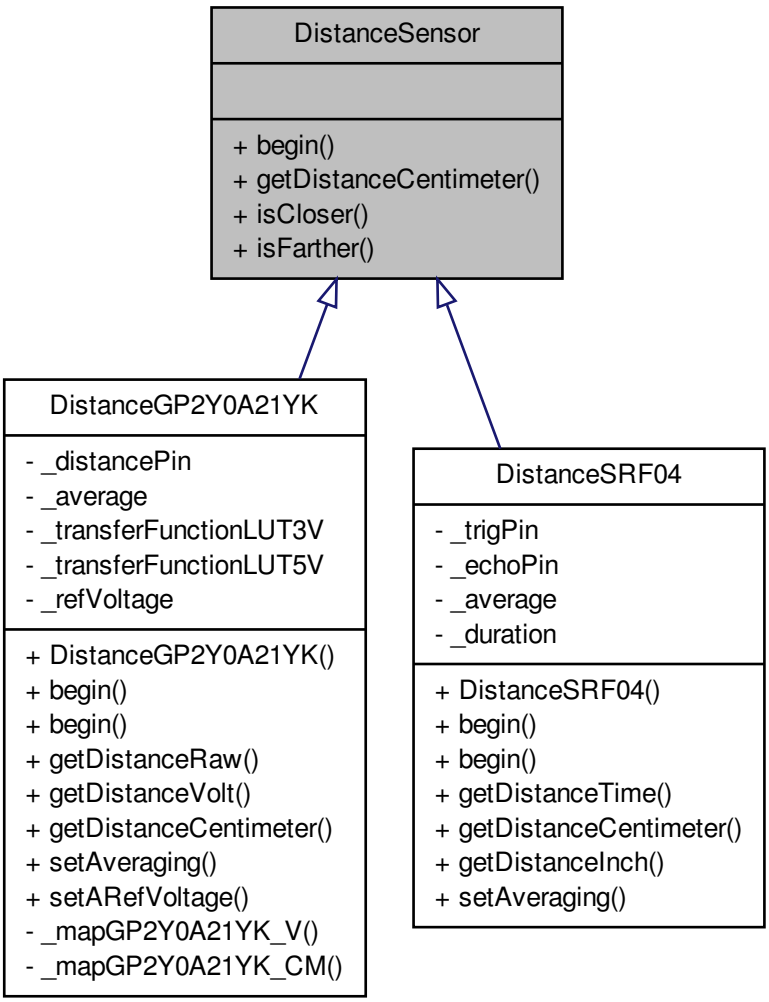
setARefVoltage:set the ADC reference voltage: (default value: 5V, set to 3 for external reference value, typically 3.3 on Arduino boards)

3.1.2.7 void DistanceGP2Y0A21YK::setAveraging (int *avg*)

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

3.2 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

3.2.1 Member Function Documentation

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

DistanceSensor.cpp - Library for,...

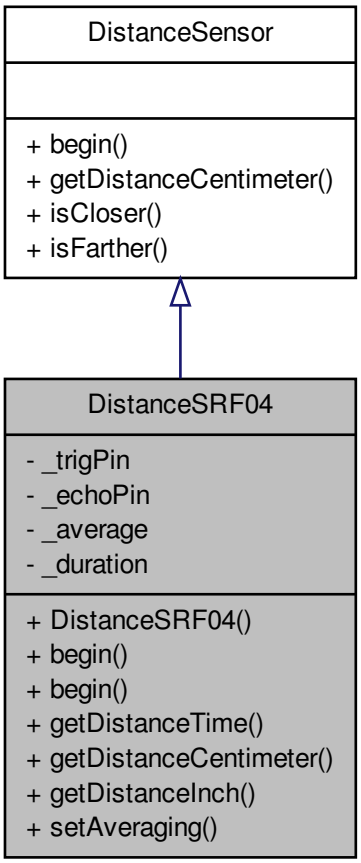
isCloser: check whether the distance to the detected object is smaller than a given threshold

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

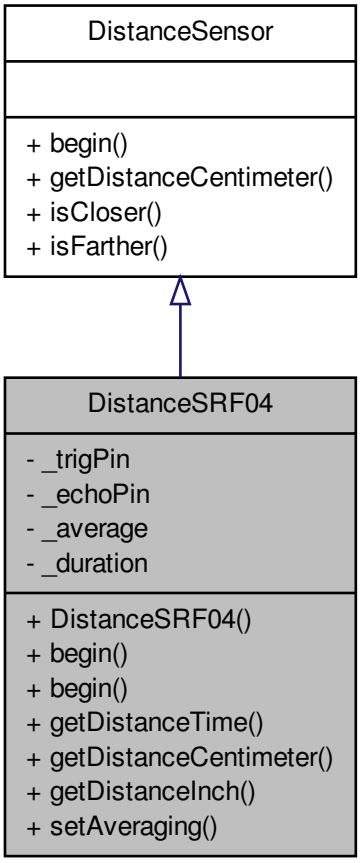
isFarther: check whether the distance to the detected object is smaller than a given threshold

3.3 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.
- void begin ()

  Begin function to set default pins.
- void begin (int echoPin, int trigPin)

  Begin variables.
- int getDistanceTime ()

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

- int getDistanceCentimeter ()

  getDistanceCentimeter(): Returns the distance in centimeters
- int getDistanceInch ()

  getDistanceInch(): Returns the distance in inches
- void setAveraging (int avg)

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

3.3.1 Constructor & Destructor Documentation

3.3.1.1 DistanceSRF04::DistanceSRF04 ( )

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

Constructor

3.3.2 Member Function Documentation

3.3.2.1 void DistanceSRF04::begin ( ) [virtual]

Begin function to set default pins.

Reimplemented from DistanceSensor.

3.3.2.2 void DistanceSRF04::begin (int *echoPin*, int *trigPin*)

Begin variables.

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

3.3.2.3 int DistanceSRF04::getDistanceCentimeter ( ) [virtual]

getDistanceCentimeter(): Returns the distance in centimeters

Implements DistanceSensor.

3.3.2.4 int DistanceSRF04::getDistanceInch ( )

getDistanceInch(): Returns the distance in inches

3.3.2.5 int DistanceSRF04::getDistanceTime ( )

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

3.3.2.6 void DistanceSRF04::setAveraging (int *avg*)

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

Index