

Air Data Computer Library

0.9 beta

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	AirDC Class Reference	3
2.1.1	Detailed Description	4
2.1.2	Constructor & Destructor Documentation	4
2.1.2.1	AirDC()	4
2.1.3	Member Function Documentation	4
2.1.3.1	IAS()	4
2.1.3.2	ISAAltitude()	5
2.1.3.3	OAT()	5
2.1.3.4	OutputSerial()	6
2.1.3.5	PitotCorrection()	6
2.1.3.6	Red()	6
2.1.3.7	RhoAir()	7
2.1.3.8	Viscosity()	7
2.1.4	Member Data Documentation	7
2.1.4.1	_AOA	8
2.1.4.2	_AOAdot	8
2.1.4.3	_AOS	8
2.1.4.4	_AOSdot	8
2.1.4.5	_CAS	8
2.1.4.6	_d	8

2.1.4.7	_h	8
2.1.4.8	_IAS	8
2.1.4.9	_lp	9
2.1.4.10	_lq	9
2.1.4.11	_lr	9
2.1.4.12	_M	9
2.1.4.13	_mu	9
2.1.4.14	_p	9
2.1.4.15	_pid	9
2.1.4.16	_PitotXcog	9
2.1.4.17	_PitotYcog	10
2.1.4.18	_PitotZcog	10
2.1.4.19	_pSeaLevel	10
2.1.4.20	_qc	10
2.1.4.21	_Re	10
2.1.4.22	_RH	10
2.1.4.23	_Rho	10
2.1.4.24	_T	10
2.1.4.25	_TAS	11
2.1.4.26	_TASPCorrected	11
2.1.4.27	_TAT	11
2.1.4.28	_uCAS	11
2.1.4.29	_uh	11
2.1.4.30	_uIAS	11
2.1.4.31	_up	11
2.1.4.32	_uqc	11
2.1.4.33	_uRH	12
2.1.4.34	_uRho	12
2.1.4.35	_uT	12
2.1.4.36	_uTAS	12
2.1.4.37	_uTAT	12
2.2	AirSensor Class Reference	12
2.2.1	Detailed Description	13
2.2.2	Constructor & Destructor Documentation	13
2.2.2.1	AirSensor()	13
2.2.3	Member Function Documentation	13
2.2.3.1	ReadDifferentialPressure()	13
2.2.3.2	ReadRH()	14
2.2.3.3	ReadStaticPressure()	14
2.2.3.4	ReadTAT()	14

Chapter 1

Class Index

1.1 Class List

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

AirDC	3
AirSensor	12

Chapter 2

Class Documentation

2.1 AirDC Class Reference

```
#include <AirDC.h>
```

Public Member Functions

- [AirDC](#) (int pid)
- void [RhoAir](#) (int mode)
- void [IAS](#) (int mode)
- void **CAS** (int mode)
- void **TAS** (int mode)
- void **Mach** (int mode)
- void [OAT](#) (int mode)
- void [ISAAltitude](#) (int mode)
- String [OutputSerial](#) (int mode)
- void [PitotCorrection](#) (int mode)
- void [Viscosity](#) (int mode)
- void [Red](#) (int mode)

Public Attributes

- int [_pid](#)
- double [_d](#)
- double [_PitotXcog](#)
- double [_PitotYcog](#)
- double [_PitotZcog](#)
- double [_p](#)
- double [_T](#)
- double [_RH](#)
- double [_qc](#)
- double [_AOA](#)
- double [_AOS](#)
- double [_pSeaLevel](#)
- double [_Rho](#)
- double [_IAS](#)

- double [_CAS](#)
- double [_TAS](#)
- double [_TASPCorrected](#)
- double [_M](#)
- double [_TAT](#)
- double [_h](#)
- double [_mu](#)
- double [_Re](#)
- double [_AOAdot](#)
- double [_AOSdot](#)
- double [_up](#)
- double [_uT](#)
- double [_uRH](#)
- double [_uqc](#)
- double [_uRho](#)
- double [_uIAS](#)
- double [_uCAS](#)
- double [_uTAS](#)
- double [_uTAT](#)
- double [_uh](#)
- double [_lp](#)
- double [_lq](#)
- double [_lr](#)

2.1.1 Detailed Description

[AirDC](#) - Library for Basic Air Data calculations Created by J.L.J., December 3, 2015.
Refer to <http://www.basicairdata.eu>

2.1.2 Constructor & Destructor Documentation

2.1.2.1 [AirDC\(\)](#)

```
AirDC::AirDC (
    int pid )
```

[AirDC](#) Default constructor

2.1.3 Member Function Documentation

2.1.3.1 [IAS\(\)](#)

```
void AirDC::IAS (
    int mode )
```

Calculates Indicated Airspeed IAS=ASI=EAS

Parameters

<i>Mode</i>	Indicates the calculation method. 1 is Basic Air Data default http://www.basicairdata.eu/pitot-tube.html https://en.wikipedia.org/wiki/Equivalent_airspeed
-------------	---

Returns

Void

2.1.3.2 ISAAltitude()

```
void AirDC::ISAAltitude (
    int mode )
```

Calculates barometric altitude with ISA atmosphere

Parameters

<i>Mode</i>	1 Uncorrected altitude above mean sea level http://www.basicairdata.eu/altimeter.html
<i>Mode</i>	2 Corrected above mean sea level altitude, pressure at sea level should be available, https://en.wikipedia.org/wiki/QNH

Returns

Void

2.1.3.3 OAT()

```
void AirDC::OAT (
    int mode )
```

Calculates Outside Air Temperature

Parameters

<i>Mode</i>	Indicates the calculation method. 1 is Basic Air Data default https://en.wikipedia.org/wiki/Total_air_temperature
-------------	--

Returns

Void

2.1.3.4 OutputSerial()

```
String AirDC::OutputSerial (
    int mode )
```

Output formatter

Parameters

<i>Mode</i>	1 Measurements output
<i>Mode</i>	2 Air data output
<i>Mode</i>	3 Measurements uncertainty output
<i>Mode</i>	4 Air data uncertainty output
<i>Mode</i>	51 Output for Temperature Logger Example

Returns

Void

2.1.3.5 PitotCorrection()

```
void AirDC::PitotCorrection (
    int mode )
```

Correct TAS based on pitot placement

Parameters

<i>Mode</i>	1 No compensation
<i>Mode</i>	2 Steady state(no angular acceleration) assumed for this method http://basicairdata.blogspot.it/2014/07/pitot-correction-for-position-and.html

Returns

Void

2.1.3.6 Red()

```
void AirDC::Red (
    int mode )
```

Calculates Re number

Parameters

<i>Mode</i>	1 Uses <i>_d</i> as reference dimension
-------------	---

Returns

Void

2.1.3.7 RhoAir()

```
void AirDC::RhoAir (
    int mode )
```

Calculates the Air Density

Parameters

<i>Mode</i>	Indicates the calculation method. 1 is Basic Air Data default http://www.basicairdata.eu/calculation-routines.html
-------------	--

Returns

Void

2.1.3.8 Viscosity()

```
void AirDC::Viscosity (
    int mode )
```

Calculates Air Viscosity

Parameters

<i>Mode</i>	1 Calculate viscosity with Sutherland's formula, note that output is multiplied by a 10e6 factor
<i>Mode</i>	2 Calculate viscosity with Sutherland's formula

Returns

Void

2.1.4 Member Data Documentation

2.1.4.1 `_AOA`

```
double AirDC::_AOA
```

Angle of Attack, rads

2.1.4.2 `_AOAdot`

```
double AirDC::_AOAdot
```

Time derivate of AOA rad/s

2.1.4.3 `_AOS`

```
double AirDC::_AOS
```

Angle of Sideslip, rads

2.1.4.4 `_AOSdot`

```
double AirDC::_AOSdot
```

Time derivate of AOS rad/s

2.1.4.5 `_CAS`

```
double AirDC::_CAS
```

Calibrated Air Speed m/s

2.1.4.6 `_d`

```
double AirDC::_d
```

Reference length for Re number calculation

2.1.4.7 `_h`

```
double AirDC::_h
```

Altitude m

2.1.4.8 `_IAS`

```
double AirDC::_IAS
```

Indicated Air speed m/s

2.1.4.9 _lp

```
double AirDC::_lp
```

Pitch rate

2.1.4.10 _lq

```
double AirDC::_lq
```

Roll rate

2.1.4.11 _lr

```
double AirDC::_lr
```

yaw rate

2.1.4.12 _M

```
double AirDC::_M
```

Mach number

2.1.4.13 _mu

```
double AirDC::_mu
```

Dynamic Viscosity Pas

2.1.4.14 _p

```
double AirDC::_p
```

Static Pressure Pa

2.1.4.15 _pid

```
int AirDC::_pid
```

Class ID

2.1.4.16 _PitotXcog

```
double AirDC::_PitotXcog
```

Distance along x body axes of the Pitot tip from center of gravity

2.1.4.17 `_PitotYcog`

```
double AirDC::_PitotYcog
```

Distance along y body axes of the Pitot tip from center of gravity

2.1.4.18 `_PitotZcog`

```
double AirDC::_PitotZcog
```

Distance along z body axes of the Pitot tip from center of gravity

2.1.4.19 `_pSeaLevel`

```
double AirDC::_pSeaLevel
```

Value of pressure at sea level Pa

2.1.4.20 `_qc`

```
double AirDC::_qc
```

Differential pressure at Pitot, Impact pressure minus static pressure Pa

2.1.4.21 `_Re`

```
double AirDC::_Re
```

Reynolds Number

2.1.4.22 `_RH`

```
double AirDC::_RH
```

Relative Humidity

2.1.4.23 `_Rho`

```
double AirDC::_Rho
```

Air Density kg/m^3

2.1.4.24 `_T`

```
double AirDC::_T
```

Temperature K

2.1.4.25 _TAS

```
double AirDC::_TAS
```

True Air Speed m/s

2.1.4.26 _TASPCorrected

```
double AirDC::_TASPCorrected
```

True Air Speed, corrected m/s

2.1.4.27 _TAT

```
double AirDC::_TAT
```

Total Air Temperature K

2.1.4.28 _uCAS

```
double AirDC::_uCAS
```

CAS uncertainty

2.1.4.29 _uh

```
double AirDC::_uh
```

Altitude uncertainty

2.1.4.30 _uIAS

```
double AirDC::_uIAS
```

IAS uncertainty

2.1.4.31 _up

```
double AirDC::_up
```

Pressure uncertainty Pa

2.1.4.32 _uqc

```
double AirDC::_uqc
```

Differential pressure uncertainty Pa

2.1.4.33 `_uRH`

```
double AirDC::_uRH
```

Relative Humidity uncertainty

2.1.4.34 `_uRho`

```
double AirDC::_uRho
```

Air density uncertainty kg/m^3

2.1.4.35 `_uT`

```
double AirDC::_uT
```

Temperature uncertainty Pa

2.1.4.36 `_uTAS`

```
double AirDC::_uTAS
```

TAS uncertainty

2.1.4.37 `_uTAT`

```
double AirDC::_uTAT
```

TAT uncertainty

The documentation for this class was generated from the following files:

- `AirDC.h`
- `AirDC.cpp`

2.2 AirSensor Class Reference

```
#include <AirSensor.h>
```

Public Member Functions

- [AirSensor](#) (int pid)
- void [ReadDifferentialPressure](#) ([AirDC](#) *out, int sensor)
- void [ReadStaticPressure](#) ([AirDC](#) *out, int sensor)
- void [ReadTAT](#) ([AirDC](#) *out, int sensor)
- void [ReadRH](#) ([AirDC](#) *out, int sensor)

Public Attributes

- `int _pid`

2.2.1 Detailed Description

[AirSensor.h](#) - Library for sensor interfacing Created by J.L.J., December 3, 2015.

Refer to <http://www.basicairdata.eu> [AirSensor](#) class handles hardware specific stuff for some sensors sends the output to an [AirDC](#) object

2.2.2 Constructor & Destructor Documentation

2.2.2.1 AirSensor()

```
AirSensor::AirSensor (  
    int pid )
```

[AirDC](#) Default constructor

2.2.3 Member Function Documentation

2.2.3.1 ReadDifferentialPressure()

```
void AirSensor::ReadDifferentialPressure (  
    AirDC * out,  
    int sensor )
```

Acquire differeintial pressure

Read the current differential pressure sensor value and copy the value to [AirDC](#)

Parameters

<i>*out,assign</i>	an AirDC class for data output
<i>sensor</i>	1, sensor HLCA12X5
<i>sensor</i>	2, sensor LDES205U
<i>sensor</i>	3, sensor MPXV7002

Returns

Void

2.2.3.2 ReadRH()

```
void AirSensor::ReadRH (
    AirDC * out,
    int sensor )
```

Acquire relative humidity

Acquire relative humidity and copy the value to [AirDC](#)

Parameters

<i>*out,assign</i>	an AirDC class for data output
<i>sensor</i>	99, debug mode RH=0.5 _uRH=0.05

Returns

Void

2.2.3.3 ReadStaticPressure()

```
void AirSensor::ReadStaticPressure (
    AirDC * out,
    int sensor )
```

Acquire static pressure

Read the current static pressure value and copy the value to [AirDC](#) implementation is not completed

Parameters

<i>*out,assign</i>	an AirDC class for data output
<i>sensor</i>	99, debug mode _p=90000 Pa and _up=100 Pa

Returns

Void

2.2.3.4 ReadTAT()

```
void AirSensor::ReadTAT (
    AirDC * out,
    int sensor )
```

Acquire total air temperature

Read the current total air temperature and copy the value to [AirDC](#)

Parameters

<i>*out,assign</i>	an AirDC class for data output
<i>sensor</i>	1, sensor DS18X20

Returns

Void

The documentation for this class was generated from the following files:

- AirSensor.h
- AirSensor.cpp

Index

[_M](#)
 [AirDC, 9](#)
[_T](#)
 [AirDC, 10](#)
[_AOA](#)
 [AirDC, 7](#)
[_AOS](#)
 [AirDC, 8](#)
[_AOAdot](#)
 [AirDC, 8](#)
[_AOSdot](#)
 [AirDC, 8](#)
[_CAS](#)
 [AirDC, 8](#)
[_IAS](#)
 [AirDC, 8](#)
[_lp](#)
 [AirDC, 8](#)
[_lq](#)
 [AirDC, 9](#)
[_lr](#)
 [AirDC, 9](#)
[_PitotXcog](#)
 [AirDC, 9](#)
[_PitotYcog](#)
 [AirDC, 9](#)
[_PitotZcog](#)
 [AirDC, 10](#)
[_RH](#)
 [AirDC, 10](#)
[_Re](#)
 [AirDC, 10](#)
[_Rho](#)
 [AirDC, 10](#)
[_TAS](#)
 [AirDC, 10](#)
[_TAT](#)
 [AirDC, 11](#)
[_TASPCorrected](#)
 [AirDC, 11](#)
[_d](#)
 [AirDC, 8](#)
[_h](#)
 [AirDC, 8](#)
[_mu](#)
 [AirDC, 9](#)
[_p](#)
 [AirDC, 9](#)
[_pSeaLevel](#)

[AirDC, 10](#)
[_pid](#)
 [AirDC, 9](#)
[_qc](#)
 [AirDC, 10](#)
[_uT](#)
 [AirDC, 12](#)
[_uCAS](#)
 [AirDC, 11](#)
[_uIAS](#)
 [AirDC, 11](#)
[_uRH](#)
 [AirDC, 11](#)
[_uRho](#)
 [AirDC, 12](#)
[_uTAS](#)
 [AirDC, 12](#)
[_uTAT](#)
 [AirDC, 12](#)
[_uh](#)
 [AirDC, 11](#)
[_up](#)
 [AirDC, 11](#)
[_uqc](#)
 [AirDC, 11](#)

[AirDC, 3](#)
 [_M, 9](#)
 [_T, 10](#)
 [_AOA, 7](#)
 [_AOS, 8](#)
 [_AOAdot, 8](#)
 [_AOSdot, 8](#)
 [_CAS, 8](#)
 [_IAS, 8](#)
 [_lp, 8](#)
 [_lq, 9](#)
 [_lr, 9](#)
 [_PitotXcog, 9](#)
 [_PitotYcog, 9](#)
 [_PitotZcog, 10](#)
 [_RH, 10](#)
 [_Re, 10](#)
 [_Rho, 10](#)
 [_TAS, 10](#)
 [_TAT, 11](#)
 [_TASPCorrected, 11](#)
 [_d, 8](#)
 [_h, 8](#)
 [_mu, 9](#)

- [_p](#), [9](#)
 - [_pSeaLevel](#), [10](#)
 - [_pid](#), [9](#)
 - [_qc](#), [10](#)
 - [_uT](#), [12](#)
 - [_uCAS](#), [11](#)
 - [_uIAS](#), [11](#)
 - [_uRH](#), [11](#)
 - [_uRho](#), [12](#)
 - [_uTAS](#), [12](#)
 - [_uTAT](#), [12](#)
 - [_uh](#), [11](#)
 - [_up](#), [11](#)
 - [_uqc](#), [11](#)
- [AirDC](#), [4](#)
- [IAS](#), [4](#)
- [ISAAltitude](#), [5](#)
- [OAT](#), [5](#)
- [OutputSerial](#), [5](#)
- [PitotCorrection](#), [6](#)
- [Red](#), [6](#)
- [RhoAir](#), [7](#)
- [Viscosity](#), [7](#)
- [AirSensor](#), [12](#)
 - [AirSensor](#), [13](#)
 - [ReadDifferentialPressure](#), [13](#)
 - [ReadRH](#), [13](#)
 - [ReadStaticPressure](#), [14](#)
 - [ReadTAT](#), [14](#)
- [IAS](#)
 - [AirDC](#), [4](#)
- [ISAAltitude](#)
 - [AirDC](#), [5](#)
- [OAT](#)
 - [AirDC](#), [5](#)
- [OutputSerial](#)
 - [AirDC](#), [5](#)
- [PitotCorrection](#)
 - [AirDC](#), [6](#)
- [ReadDifferentialPressure](#)
 - [AirSensor](#), [13](#)
- [ReadRH](#)
 - [AirSensor](#), [13](#)
- [ReadStaticPressure](#)
 - [AirSensor](#), [14](#)
- [ReadTAT](#)
 - [AirSensor](#), [14](#)
- [Red](#)
 - [AirDC](#), [6](#)
- [RhoAir](#)
 - [AirDC](#), [7](#)
- [Viscosity](#)
 - [AirDC](#), [7](#)