Air Data Computer Library

0.9 beta

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

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 )
```

Calcualtes Indicated Airspeed IAS=ASI=EAS

Parameters

Mode	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

Mode	1 Uncorrected altitude above mean sea level http://www.basicairdata.eu/altimeter.html
Mode	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

Mode	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

Mode	1 Measurements output
Mode	2 Air data output
Mode	3 Measurements uncertainty output
Mode	4 Air data uncertainty output
Mode	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

Mode	1 No compensation
Mode	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

Mode	1 Uses _d as reference dimension

Returns

Void

2.1.3.7 RhoAir()

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

Calculates the Air Density

Parameters

Mode	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

Mode	1 Calculate viscosity with Sutherland's formula, note that output is multiplied by a 10e6 factor
Mode	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 _Ip

```
double AirDC::_Ip
```

Pitch rate

2.1.4.10 _Iq

```
double AirDC::_Iq
```

Roll rate

2.1.4.11 _Ir

```
double AirDC::_Ir
```

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/^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

∗out,assign	an AirDC class for data output
sensor	1, sensor HLCA12X5
sensor	2, sensor LDES205U
sensor	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

∗out,assign	an AirDC class for data output
sensor	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

∗out,assign	an AirDC class for data output
sensor	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

∗out,assign	an AirDC class for data output
sensor	1, sensor DS18X20

Returns

Void

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

- AirSensor.h
- AirSensor.cpp

Index