

DHT22 library for Arduino

1.0.1

Generated by Doxygen 1.8.14

Contents

1	DHT22 - AM2303 temperature and humidity sensor library for Arduino	1
2	Class Index	5
2.1	Class List	5
3	File Index	7
3.1	File List	7
4	Class Documentation	9
4.1	DHT22 Class Reference	9
4.1.1	Detailed Description	9
4.1.2	Constructor & Destructor Documentation	10
4.1.2.1	DHT22()	10
4.1.3	Member Function Documentation	10
4.1.3.1	available()	10
4.1.3.2	begin()	10
4.1.3.3	readHumidity()	11
4.1.3.4	readTemperature()	11
5	File Documentation	13
5.1	DHT22.cpp File Reference	13
5.1.1	Detailed Description	13
5.2	DHT22.h File Reference	13
5.2.1	Detailed Description	14
5.2.2	Macro Definition Documentation	14
5.2.2.1	DHT22_MIN_READ_INTERVAL	14
5.2.2.2	DHT22_NUM_DATA_BITS	14
	Index	15

Chapter 1

DHT22 - AM2303 temperature and humidity sensor library for Arduino

This is a calibrated AM2303 digital temperature and relative humidity sensor on a [DHT22](#) breakout PCB.

Library features

- Read 16-bit temperature (synchronous blocking)
- Read 16-bit relative humidity (synchronous blocking)

AM2303 sensor specifications

- Voltage: 3.3 .. 5V
- Ultra-low power:
 - Typical 15uA dormancy
 - Typical 500uA measuring
- Single wire digital serial interface
- Calibrated digital signal
- Outstanding long term stability
- No additional electronic components needed
- Humidity:
 - Range: 0 .. 99.9 RH (Relative Humidity)
 - Resolution: 0.1 RH
 - Accuracy: +/- 2 RH (at 25 degree Celsius)
- Temperature:
 - Range: -40 .. +125 degree Celsius
 - Resolution: 0.1 degree Celsius
 - Accuracy: +/- 0.4 degree Celsius
- Minimum read interval: 2000 ms
- ~31ms to synchronous read humidity, temperature and parity data from sensor (5 Bytes)

Safety warning

According to the datasheet, the AM2302 is a low cost consumer temperature sensor. It may not be used in safety critical applications, emergency stop devices or any other occasion that failure of AM2303 may cause personal injury.

Hardware

Pull-up resistor DAT pin

- Connect an external 3k3 . . 10k pull-up resistor between the DAT and VCC pins only when:
 - Using a AM2302 sensor without a DT22 breakout PCB **and** the MCU IO pin has no built-in or external pull-up resistor.
- The [DHT22](#) breakout PCB contains a 3k3 pull-up resistor between DAT and VCC.
- Please refer to the MCU datasheet or board schematic for more information about IO pin pull-up resistors.

External capacitor

- Tip: Connect a 100nF capacitor between the sensor pins VCC and GND when read errors occurs. This may stabilize the power supply.

Connection [DHT22](#) - Arduino

DHT22	Arduino UNO / Nano / Pro Mini / Leonardo / Mega2560
GND	GND
VCC	5V (or 3.3V)
DAT	2 (DIGITAL pin)

Connection [DHT22](#) - ESP8266

Some ESP8266 boards uses Arduino pin 2 -> GPIO4 which is D4 text on the board. Make sure you're using the right pin.

DHT22	ESP8266 / WeMos D1 R2 / ESP12E / NodeMCU
GND	GND
VCC	3.3V
DAT	D4

Connection [DHT22](#) - WeMos LOLIN32

WeMos LOLIN32 requires an additional 100nF capacitor over the GND - VCC pins to prevent parity errors.

Use pin 0 to prevent flash problems.

DHT22	WeMos Lolin32
GND	GND
VCC	3.3V
DAT	0

Other MCU's may work, but are not tested.

Examples

- Examples | Erriez DH22 | [Example](#)

Usage

Initialization

```
{c++}
#include <DHT22.h>

// Connect DHT22 DAT pin to Arduino board

// Arduino DIGITAL pin
#define DHT22_PIN    2
// Some ESP8266 boards uses D2 instead of 2
// #define DHT22_PIN  D2
// LOLIN32 uses another pin
// #define DHT22_PIN  0

DHT22 sensor = DHT22(DHT22_PIN);

void setup()
{
    // Initialize serial port
    Serial.begin(115200);
    Serial.println(F("DHT22 temperature and humidity sensor example\n"));

    // Initialize sensor
    sensor.begin();
}
```

Read temperature and humidity

```
{c++}
void loop()
{
    // Check minimum interval of 2000 ms between sensor reads
    if (sensor.available()) {
        // Read temperature from sensor
        int16_t temperature = sensor.readTemperature();

        // Read humidity from sensor
        int16_t humidity = sensor.readHumidity();

        if (temperature == ~0) {
            // Print error (Check hardware connection)
            Serial.print(F("Temperature: Error"));
        } else {
            // Print temperature
            Serial.print(F("Temperature: "));
            Serial.print(temperature / 10);
            Serial.print(F("."));
            Serial.print(temperature % 10);
            Serial.println(F(" *C"));
        }

        if (humidity == ~0) {
            // Print error (Check hardware connection)
            Serial.print(F("Humidity: Error"));
        } else {
            // Print humidity
            Serial.print(F("Humidity: "));
            Serial.print(humidity / 10);
            Serial.print(F("."));
            Serial.print(humidity % 10);
            Serial.println(F(" %"));
        }

        Serial.println();
    }
}
```

Serial output

DHT22 temperature and humidity sensor example

```
Temperature: 17.7 *C  
Humidity: 41.0 %
```

```
Temperature: 17.8 *C  
Humidity: 41.1 %
```

...

Documentation

- [Doxygen online HTML](#)
- [Doxygen PDF](#)
- [AM2303 datasheet](#)
- [DHT22 datasheet](#)

Library dependencies

- None

Library installation

Please refer to the [Wiki](#) page.

Other Arduino Libraries and Sketches from Erriez

- [Erriez Libraries and Sketches](#)

Chapter 2

Class Index

2.1 Class List

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

DHT22	
DHT22 sensor class	9

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

DHT22.cpp	
DHT22 (AM2303) Humidity and Temperature sensor library for Arduino	13
DHT22.h	
DHT22 (AM2303) Humidity and Temperature sensor library for Arduino	13

Chapter 4

Class Documentation

4.1 DHT22 Class Reference

DHT22 sensor class.

```
#include <DHT22.h>
```

Public Member Functions

- [DHT22](#) (uint8_t pin)
Constructor [DHT22](#) sensor.
- void [begin](#) ()
Initialize sensor.
- bool [available](#) ()
Check if new temperature or humidity read is allowed.
- int16_t [readTemperature](#) ()
Read temperature from sensor.
- int16_t [readHumidity](#) ()
Read humidity from sensor.

4.1.1 Detailed Description

DHT22 sensor class.

According to the datasheet, the AM2302 is a low cost consumer temperature sensor. It may not be used in safety critical applications, emergency stop devices or any other occasion that failure of AM2303 may cause personal injury.

The AM2303 is factory calibrated. However, deviation may occur with aging.

The temperature/humidity read interval in this library is cached for 2 seconds to prevent heating-up the internal chip with continues reading.

Global interrupts are disabled during a synchronous sensor read transfer. This is required to sample the data bit lengths at maximum speed on low-end devices without any application interrupts. The read calls are protected with a timeout.

The application is responsible for checking ~ 0 values after a read which means that the read failed or a timeout occurred. Multiple reads by the application with an average calculation is recommended.

Definition at line 79 of file DHT22.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 DHT22()

```
DHT22::DHT22 (
    uint8_t pin ) [explicit]
```

Constructor [DHT22](#) sensor.

Parameters

<i>pin</i>	Data pin sensor.
------------	------------------

Definition at line 38 of file DHT22.cpp.

4.1.3 Member Function Documentation

4.1.3.1 available()

```
bool DHT22::available ( )
```

Check if new temperature or humidity read is allowed.

The application should call this function and check if a new temperature and humidity can be read to prevent too fast sensor reads.

Return values

<i>true</i>	Available, interval between sensor reads \geq 2000 ms.
<i>false</i>	Not available, interval between sensor reads too short.

Definition at line 87 of file DHT22.cpp.

4.1.3.2 begin()

```
void DHT22::begin ( )
```

Initialize sensor.

Call this function from `setup()`.

- Connect an external 3k3..10k pull-up resistor between the DAT and VCC pins only when:
 - using a AM2302 sensor without a DT22 breakout PCB
AND
 - the MCU IO pin has no built-in or external pull-up resistor.
- The [DHT22](#) breakout PCB contains a 3k3 pull-up resistor between DAT and VCC.
- Please refer to the MCU datasheet or board schematic for more information about IO pin pull-up resistors.

Definition at line 68 of file DHT22.cpp.

4.1.3.3 readHumidity()

```
int16_t DHT22::readHumidity ( )
```

Read humidity from sensor.

Return values

<i>Humidity</i>	Signed humidity with last digit after the point.
<i>~0</i>	An error occurred.

Definition at line 130 of file DHT22.cpp.

4.1.3.4 readTemperature()

```
int16_t DHT22::readTemperature ( )
```

Read temperature from sensor.

Returns the actual temperature, or a cached temperature when read interval is too short.

Return values

<i>Temperature</i>	Signed temperature with last digit after the point.
<i>~0</i>	An error occurred.

Definition at line 107 of file DHT22.cpp.

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

- [DHT22.h](#)
- [DHT22.cpp](#)

Chapter 5

File Documentation

5.1 DHT22.cpp File Reference

DHT22 (AM2303) Humidity and Temperature sensor library for Arduino.

```
#include "DHT22.h"
```

5.1.1 Detailed Description

DHT22 (AM2303) Humidity and Temperature sensor library for Arduino.

Source: <https://github.com/Erriez/ErriezDHT22>

5.2 DHT22.h File Reference

DHT22 (AM2303) Humidity and Temperature sensor library for Arduino.

```
#include <Arduino.h>
```

Classes

- class **DHT22**
DHT22 sensor class.

Macros

- #define **DHT22_MIN_READ_INTERVAL** 2000
Enable debug prints to Serial.
- #define **DHT22_NUM_DATA_BITS** (5 * 8)
- #define **DEBUG_PRINTLN**(...) {}
Debug print configuration.

5.2.1 Detailed Description

[DHT22](#) (AM2303) Humidity and Temperature sensor library for Arduino.

Source: <https://github.com/Erriez/ErriezDHT22>

5.2.2 Macro Definition Documentation

5.2.2.1 DHT22_MIN_READ_INTERVAL

```
#define DHT22_MIN_READ_INTERVAL 2000
```

Enable debug prints to Serial.

Minimum interval between sensor reads in milli seconds

Definition at line 41 of file DHT22.h.

5.2.2.2 DHT22_NUM_DATA_BITS

```
#define DHT22_NUM_DATA_BITS (5 * 8)
```

Number of data bits is 5 Bytes * 8 bits: 1 Byte: Humidity high 1 Byte: Humidity low 1 Byte: Temperature high 1 Byte: Temperature low 1 Byte: Parity

Definition at line 49 of file DHT22.h.

Index

available

DHT22, [10](#)

begin

DHT22, [10](#)

DHT22, [9](#)

available, [10](#)

begin, [10](#)

DHT22, [10](#)

readHumidity, [11](#)

readTemperature, [11](#)

DHT22.cpp, [13](#)

DHT22.h, [13](#)

DHT22_MIN_READ_INTERVAL, [14](#)

DHT22_NUM_DATA_BITS, [14](#)

DHT22_MIN_READ_INTERVAL

DHT22.h, [14](#)

DHT22_NUM_DATA_BITS

DHT22.h, [14](#)

readHumidity

DHT22, [11](#)

readTemperature

DHT22, [11](#)