

Erriez DHT22 library for Arduino

1.2.0

Generated by Doxygen 1.8.11

Contents

1	DHT22 - AM2302/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(uint8_t pin)	10
4.1.3	Member Function Documentation	10
4.1.3.1	available()	10
4.1.3.2	begin(uint8_t numSamples=0)	10
4.1.3.3	readHumidity()	11
4.1.3.4	readSensorData()	11
4.1.3.5	readTemperature()	11
5	File Documentation	13
5.1	ErriezDHT22.cpp File Reference	13
5.1.1	Detailed Description	13
5.2	ErriezDHT22.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 - AM2302/AM2303 temperature and humidity sensor library for Arduino

This is an Arduino library for the calibrated AM2302/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)
- Configurable number of read retries when a read error occurs (default is 1 read + 2 retries)
- Long time duration example
- Temperature and humidity average with a configurable number of samples to remove jitter

AM2302/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/AM2303 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 AM2302/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/AM2303 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.

Board - DHT22 pins	VCC	GND	DAT
Arduino UNO / Nano / Micro (ATMega328 boards)	5V (or 3V3)	GND	2 (DIGITAL pin)
Arduino Leonardo	5V (or 3V3)	GND	2 (DIGITAL pin)
Arduino Mega2560	5V (or 3V3)	GND	2 (DIGITAL pin)
Arduino DUE (ATSAM3X8E)	3V3	GND	2 (DIGITAL pin)
ESP8266 (ESP12E / WeMos D1 R2 / NodeMCU v2 or v3)	3V3	GND	GPIO4 (D2)
ESP32 (WeMos Lolin32 OLED / WeMos LOLIN D32)	3V3	GND	GPIO4

Notes:

- GPIO4 uses sketch pin number 4 and is labeled as D2 on some WeMos ESP8266 boards.
- Other MCU's may work, but are not tested.

Examples

Arduino IDE | Examples | Erriez [DHT22](#) Temperature & Humidity:

- [DHT22](#) Getting started example.
- [DHT22Average](#) Calculate average temperature and humidity.
- [DHT22DurationTest](#) Test reliability connection.
- [DHT22Logging](#) Write temperature and humidity every 10 minutes to .CSV file on SD-card with DS3231 RTC.
- [DHT22LowPower](#) LowPower AVR targets only. Arduino Pro or Pro Mini at 8MHz is recommended.

Documentation

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

Usage

Initialization

```
1 {c++}
2 #include <ErriezDHT22.h>
3
4 // Connect DHT22 DAT pin to Arduino board
5
6 // Connect DHT22 DAT pin to Arduino DIGITAL pin
7 #if defined(ARDUINO_ARCH_AVR) || defined(ARDUINO_SAM_DUE)
8 #define DHT22_PIN 2
9 #elif defined(ESP8266) || defined(ESP32)
10 #define DHT22_PIN 4 // GPIO4 (Labeled as D2 on some ESP8266 boards)
11 #else
12 #error "May work, but not tested on this target"
13 #endif
14
15 DHT22 dht22 = DHT22(DHT22_PIN);
16
17 void setup()
18 {
19     // Initialize serial port
20     Serial.begin(115200);
21     Serial.println(F("DHT22 temperature and humidity sensor example\n"));
22
23     // Initialize sensor
24     dht22.begin();
25 }
```

Read temperature and humidity

```
1 {c++}
2 void loop()
3 {
4     // Check minimum interval of 2000 ms between sensor reads
5     if (dht22.available()) {
6         // Read temperature from sensor
7         int16_t temperature = dht22.readTemperature();
8
9         // Read humidity from sensor
10        int16_t humidity = dht22.readHumidity();
11
12        if (temperature == ~0) {
13            // Print error (Check hardware connection)
14            Serial.print(F("Temperature: Error"));
15        } else {
16            // Print temperature
17            Serial.print(F("Temperature: "));
18            Serial.print(temperature / 10);
19            Serial.print(F("."));
20            Serial.print(temperature % 10);
21            Serial.println(F(" *C"));
22        }
23
24        if (humidity == ~0) {
25            // Print error (Check hardware connection)
26            Serial.print(F("Humidity: Error"));
27        } else {
28            // Print humidity
29            Serial.print(F("Humidity: "));
30            Serial.print(humidity / 10);
31            Serial.print(F("."));
32            Serial.print(humidity % 10);
33            Serial.println(F(" %"));
34        }
35
36        Serial.println();
37    }
38 }
```

Serial output

```
1 DHT22 temperature and humidity sensor example
2
3 Temperature: 17.7 *C
4 Humidity: 41.0 %
5
6 Temperature: 17.8 *C
7 Humidity: 41.1 %
8
9 ...
```

Library dependencies

- `LowPower` library for `DHT22LowPower.ino`.

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:

ErriezDHT22.cpp	
DHT22 (AM2302/AM2303) Humidity and Temperature sensor library for Arduino	13
ErriezDHT22.h	
DHT22 (AM2302/AM2303) Humidity and Temperature sensor library for Arduino	13

Chapter 4

Class Documentation

4.1 DHT22 Class Reference

DHT22 sensor class.

```
#include <ErriezDHT22.h>
```

Public Member Functions

- [DHT22](#) (uint8_t pin)
Constructor [DHT22](#) sensor.
- void [begin](#) (uint8_t numSamples=0)
Initialize sensor.
- bool [available](#) ()
Check if new temperature or humidity read is allowed.
- bool [readSensorData](#) ()
Read data from sensor.
- 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/AM2303 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 AM2302/AM2303 may cause personal injury.

The AM2302/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 80 of file ErriezDHT22.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 DHT22::DHT22 (uint8_t *pin*) [explicit]

Constructor [DHT22](#) sensor.

Parameters

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

Definition at line 39 of file ErriezDHT22.cpp.

4.1.3 Member Function Documentation

4.1.3.1 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 and sensor read was successful.
<i>false</i>	Not available, interval between sensor reads too short, or read failed.

Definition at line 101 of file ErriezDHT22.cpp.

4.1.3.2 void DHT22::begin (uint8_t *numSamples* = 0)

Initialize sensor.

Parameters

<i>numSamples</i>	Number of samples to calculate temperature and humidity average. This allocates <code>sizeof(int16_t) * number of samples</code> . Value 0 (default) will disable average calculation.
-------------------	--

Call this function from `setup()`.
 \n

- Connect an external 3k3..10k pull-up resistor between the DAT and VCC pins only when:
 - using a AM2302/AM2303 sensor without a DT22 breakout PCB
 AND
 - the MCU IO pin has no built-in or external pull-up resistor.
 \n
- The DHT22 breakout PCB contains a 3k3 pull-up resistor between DAT and VCC.
 \n
- Please refer to the MCU datasheet or board schematic for more information about IO pin
 pull-up resistors.

Definition at line 75 of file ErriezDHT22.cpp.

4.1.3.3 `int16_t DHT22::readHumidity ()`

Read humidity from sensor.

Return values

<i>Humidity</i>	Signed humidity with last digit after the point.
~ 0	Invalid conversion: Sensor read error occurred.

Definition at line 164 of file ErriezDHT22.cpp.

4.1.3.4 `bool DHT22::readSensorData ()`

Read data from sensor.

5 Bytes data will be read when interval between previous read \geq 2000 ms.

The sensor data is read until a valid conversion has been performed. A valid conversion consists of:

- A valid start condition
- A successful sensor read (5 Bytes data)
- A correct checksum

Return values

<i>true</i>	Last conversion was successful.
<i>false</i>	Last conversion was unsuccessful.

Definition at line 212 of file ErriezDHT22.cpp.

4.1.3.5 `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.
~ 0	Invalid conversion: Sensor read occurred.

Definition at line 121 of file ErriezDHT22.cpp.

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

- [ErriezDHT22.h](#)
- [ErriezDHT22.cpp](#)

Chapter 5

File Documentation

5.1 ErriezDHT22.cpp File Reference

[DHT22](#) (AM2302/AM2303) Humidity and Temperature sensor library for Arduino.

```
#include "ErriezDHT22.h"
```

5.1.1 Detailed Description

[DHT22](#) (AM2302/AM2303) Humidity and Temperature sensor library for Arduino.

Source: <https://github.com/Erriez/ErriezDHT22> Documentation: <https://erriez.github.io/ErriezDHT22>

5.2 ErriezDHT22.h File Reference

[DHT22](#) (AM2302/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 (AM2302/AM2303) Humidity and Temperature sensor library for Arduino.

Source: <https://github.com/Erriez/ErriezDHT22> Documentation: <https://erriez.github.io/ErriezDHT22>

5.2.2 Macro Definition Documentation

5.2.2.1 `#define DHT22_MIN_READ_INTERVAL 2000`

Enable debug prints to Serial.

Minimum interval between sensor reads in milli seconds

Definition at line 42 of file ErriezDHT22.h.

5.2.2.2 `#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 50 of file ErriezDHT22.h.

Index

available

DHT22, [10](#)

begin

DHT22, [10](#)

DHT22, [9](#)

available, [10](#)

begin, [10](#)

DHT22, [10](#)

readHumidity, [10](#)

readSensorData, [11](#)

readTemperature, [11](#)

DHT22_MIN_READ_INTERVAL

ErriezDHT22.h, [14](#)

DHT22_NUM_DATA_BITS

ErriezDHT22.h, [14](#)

ErriezDHT22.cpp, [13](#)

ErriezDHT22.h, [13](#)

DHT22_MIN_READ_INTERVAL, [14](#)

DHT22_NUM_DATA_BITS, [14](#)

readHumidity

DHT22, [10](#)

readSensorData

DHT22, [11](#)

readTemperature

DHT22, [11](#)