Erriez DHT22 library for Arduino

1.0.1

Generated by Doxygen 1.8.11

Contents

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

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.

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	D2

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

Arduino IDE | Examples | Erriez DHT22 Temperature & Humidity:

- DHT22

- DHT22Average

- DHT22DurationTest

Documentation

- Doxygen online HTML

- Doxygen PDF

- AM2303 datasheet

- DHT22 datasheet

Usage

Initialization

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

// Connect DTH22 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) {
```

```
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

- 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:

Chapter 3

File Index

3.1 File List

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

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 maxReadRetries=2)

    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.
- uint8_t getNumRetriesLastConversion ()

    Get number of retries during last conversion.

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 $\sim$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

pin	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

true	Available, interval between sensor reads >= 2000 ms.
false	Not available, interval between sensor reads too short.

Definition at line 95 of file ErriezDHT22.cpp.

4.1.3.2 void DHT22::begin (uint8_t *maxReadRetries* = 2)

Initialize sensor.

Parameters

maxReadRetries	Maximum number of sensor read retries after a sensor read error. Set maxReadRetries to 0 to read data from sensor once. Default value: 2

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

- Connect an external 3k3..10k pull-up resistor between the DAT and VCC pins only when:\n
    - using a AM2302/AM2303 sensor without a DT22 breakout PCB\n
       AND\n
    - 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\n
  pull-up resistors.
```

Definition at line 73 of file ErriezDHT22.cpp.

4.1.3.3 uint8_t DHT22::getNumRetriesLastConversion ( )

Get number of retries during last conversion.

Returns

> Number of retries during sensor read. Value 0 is one successful read without retries.

Definition at line 158 of file ErriezDHT22.cpp.

4.1.3.4 int16_t DHT22::readHumidity ( )

Read humidity from sensor.

Return values

Humidity	Signed humidity with last digit after the point.
∼0	Invalid conversion: Sensor read error occured. Use getNumRetriesLastConversion() to get number of read retries.

Definition at line 140 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

Temperature	Signed temperature with last digit after the point.
∼0	Invalid conversion: Sensor read occurred. Use getNumRetriesLastConversion() to get number of read retries.

Definition at line 116 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.↩](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