

AM2302-Sensor

Generated by Doxygen 1.9.1

1 AM2302-Sensor Library	1
1.1 Contents	1
1.2 Sensor Documentation	1
1.3 Library Usage	2
1.3.1 Controllers	2
1.3.2 Usage the AM2302-Sensor library in the Code	2
1.3.3 Status Codes of AM2302-Sensor	2
1.4 License	2
1.5 Helpful Links	2
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 AM2302::AM2302_Sensor Class Reference	7
4.1.1 Constructor & Destructor Documentation	7
4.1.1.1 AM2302_Sensor()	7
4.1.2 Member Function Documentation	8
4.1.2.1 begin()	8
4.1.2.2 get_sensorState()	8
4.1.2.3 read()	8
5 File Documentation	9
5.1 src/AM2302-Sensor.h File Reference	9
5.1.1 Detailed Description	10
5.1.2 Function Documentation	10
5.1.2.1 print_byte_as_bit()	10
Index	11

Chapter 1

AM2302-Sensor Library

Sensor Library for the AM2302 Sensor (aka DHT22) from ASAIR.

This Library is a controller independent library for reading the AM2302 Sensor also known as DHT22.

1.1 Contents

- [Sensor Documentation](#)
- [AM2302 Library Usage](#)
- [License](#)
- [Helpful Links](#)

1.2 Sensor Documentation

One small Docu you will find in the docs folder of this repo.

The actual manufacturer page is linked here: [ASAIR AM2302](#).

The most detailed datasheet you will find here: [AM2302 Product Manual](#).

The Sensor-Library documentation you will find in the provided [refman.pdf](#), or under github-pages [github.io](#).

REMARK:

Against the most documentations and datasheets the following Pin description is correct (from left to right):

- Pin1: VDD (3,3...5 V)
- Pin2: SDA (Serial Data, two way)
- Pin3: GND
- PIN4: GND

1.3 Library Usage

1.3.1 Controllers

The library is intended to be used on each microcontroller for Example:

- Arduino Nano
- Arduino Nano 33 IOT
- ESP8266
- ESP32
- etc ...

1.3.2 Usage the AM2302-Sensor library in the Code

Include the library

```
#include <AM2302-Sensor.h>
```

The library use namespaces, so the object can be instantiated and used by:

```
AM2302::AM2302_Sensor am2302{PIN};  
void setup() {  
    am2302.begin();  
    auto status = am2302.read();  
    Serial.print("\n\nstatus of sensor read(): ");  
    Serial.println(AM2302::AM2302_Sensor::get_sensorState(status));  
    Serial.print("Temperature: ");  
    Serial.println(am2302.get_Temperature());  
    Serial.print("Humidity: ");  
    Serial.println(am2302.get_Humidity());  
}
```

1.3.3 Status Codes of AM2302-Sensor

The following status codes exists:

- AM2302_READ_OK {0};
- AM2302_ERROR_CHECKSUM {-1};
- AM2302_ERROR_TIMEOUT {-2};
- AM2302_ERROR_READ_FREQ {-3};

1.4 License

This library is licensed under MIT Licence.

[AM2302-Sensor License](#)

1.5 Helpful Links

- [ESP8266-01-Adapter](#)

Chapter 2

Class Index

2.1 Class List

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

AM2302::AM2302_Sensor	7
---	---

Chapter 3

File Index

3.1 File List

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

src/ AM2302-Sensor.h	
Measure Temperature and Humidity of AM2302-Sensor	9

Chapter 4

Class Documentation

4.1 AM2302::AM2302_Sensor Class Reference

Public Member Functions

- [AM2302_Sensor](#) (uint8_t pin)
Construct a new am2302::am2302 sensor::am2302 sensor object.
- bool [begin](#) ()
begin function, setup pin and run sensor check.
- int8_t [read](#) ()
read function, call of read_sensor()
- float [get_Temperature](#) () const
- float [get_Humidity](#) () const

Static Public Member Functions

- static const char * [get_sensorState](#) (int8_t state)
get Sensor State in human readable manner

4.1.1 Constructor & Destructor Documentation

4.1.1.1 AM2302_Sensor()

```
AM2302::AM2302_Sensor::AM2302_Sensor (
    uint8_t pin ) [explicit]
```

Construct a new am2302::am2302 sensor::am2302 sensor object.

Parameters

<i>pin</i>	Pin for AM2302 sensor
------------	-----------------------

4.1.2 Member Function Documentation

4.1.2.1 begin()

```
bool AM2302::AM2302_Sensor::begin ( )
```

begin function, setup pin and run sensor check.

Returns

true if sensor check is successful.

false if sensor check failed.

4.1.2.2 get_sensorState()

```
const char * AM2302::AM2302_Sensor::get_sensorState (
    int8_t state ) [static]
```

get Sensor State in human readable manner

Returns

sensor state : OK, Checksum Error or Timeout Error

4.1.2.3 read()

```
int8_t AM2302::AM2302_Sensor::read ( )
```

read function, call of read_sensor()

Returns

sensor status

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

- [src/AM2302-Sensor.h](#)
- [src/AM2302-Sensor.cpp](#)

Chapter 5

File Documentation

5.1 src/AM2302-Sensor.h File Reference

Measure Temperature and Humidity of AM2302-Sensor.

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

Classes

- class [AM2302::AM2302_Sensor](#)

Functions

- void [AM2302_Tools::print_byte_as_bit](#) (char value)
helper function to print byte as bits

Variables

- constexpr const char * **AM2302::AM2302_STATE_OK** {"OK"}
- constexpr const char * **AM2302::AM2302_STATE_ERR_CKSUM** {"Error: Checksum"}
- constexpr const char * **AM2302::AM2302_STATE_ERR_TIMEOUT** {"Error: Timeout"}
- constexpr const char * **AM2302::AM2302_STATE_ERR_READ_FREQ** {"Error: Read Frequency"}
- constexpr int8_t **AM2302::AM2302_READ_OK** {0}
- constexpr int8_t **AM2302::AM2302_ERROR_CHECKSUM** {-1}
- constexpr int8_t **AM2302::AM2302_ERROR_TIMEOUT** {-2}
- constexpr int8_t **AM2302::AM2302_ERROR_READ_FREQ** {-3}
- constexpr uint8_t **AM2302::READ_TIMEOUT** {100U}
- constexpr uint16_t **AM2302::READ_FREQUENCY** {2000U}

5.1.1 Detailed Description

Measure Temperature and Humidity of AM2302-Sensor.

Author

Frank Häfele

Date

21.11.2023

Version

1.4.0

5.1.2 Function Documentation

5.1.2.1 `print_byte_as_bit()`

```
void AM2302_Tools::print_byte_as_bit (
    char value )
```

helper function to print byte as bits

Parameters

<i>value</i>	byte with 8 bits
--------------	------------------

Index

- AM2302-Sensor.h
 - print_byte_as_bit, [10](#)
- AM2302::AM2302_Sensor, [7](#)
 - AM2302_Sensor, [7](#)
 - begin, [8](#)
 - get_sensorState, [8](#)
 - read, [8](#)
- AM2302_Sensor
 - AM2302::AM2302_Sensor, [7](#)
- begin
 - AM2302::AM2302_Sensor, [8](#)
- get_sensorState
 - AM2302::AM2302_Sensor, [8](#)
- print_byte_as_bit
 - AM2302-Sensor.h, [10](#)
- read
 - AM2302::AM2302_Sensor, [8](#)
- src/AM2302-Sensor.h, [9](#)