Microfire LLC SHT3x Arduino Library

Release Information

Release History

Release	Date	Description
1.0.0	7/25/2021	Initial

Legal Disclaimer

TECHNICAL AND RELIABILITY DATA FOR MICROFIRE LLC PRODUCTS (INCLUDING DATASHEETS) AS MODIFIED FROM TIME TO TIME (“RESOURCES”) ARE PROVIDED BY MICROFIRE LLC "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN NO EVENT SHALL MICROFIRE LLC BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE RESOURCES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MICROFIRE LLC reserves the right to make any enhancements, improvements, corrections or any other modifications to the RESOURCES or any products described in them at any time and without further notice.

The RESOURCES are intended for skilled users with suitable levels of design knowledge. Users are solely responsible for their selection and use of the RESOURCES and any application of the products described in them. User agrees to indemnify and hold MICROFIRE LLC harmless against all liabilities, costs, damages or other losses arising out of their use of the RESOURCES.

HIGH RISK ACTIVITIES. MICROFIRE LLC products are not designed, manufactured or intended for use in hazardous environments requiring fail safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, weapons systems or safety-critical applications (including life support systems and other medical devices), in which the failure of the products could lead directly to death, personal injury or severe physical or environmental damage, or business loss ("High Risk Activities"). MICROFIRE LLC specifically disclaims any express or implied warranty of fitness for High Risk Activities and accepts no liability for use or inclusions of MICROFIRE LLC products in High Risk Activities.

Library Documentation

Installation

Installation of this library can be done from within the Arduino IDE's library manager. Search for `Microfire` and choose the library labeled `Microfire SHT3x`.

The library is also installable through PlatformIO's library manager.

Member Variables

const float tempC

Temperature measurement, in Celsius.

const float tempF

Temperature measurement, in Fahrenheit.

const float vpd_kPa

Vapor pressure deficit in kilopascals.

const float dew_pointC

The dew point, in Celsius

const float dew_pointF

The dew point, in Fahrenheit

const float heat_indexC

The heat index, in Celsius

const float heat_indexF

The heat index, in Fahrenheit

const float wet_bulbC

The wet bulb temperature, in Celsius

const float wet_bulbF

The wet bulb temperature, in Fahrenheit

const int RH

Relative humidity

const int status

Status code of the last measurement or calibration.

0: STATUS_NO_ERROR
1: STATUS_NOT_CONNECTED
2: STATUS_CRC_ERROR

Member Methods

begin

Initializes the library and determines if the module is connected. Wire.begin() must be called prior.

Definition

```
bool begin(TwoWire &wirePort = Wire, uint8_t address = 0x0E);
```

Parameters

Parameter	Description
&wirePort	TwoWire I2C interface
address	I2C address of the module

Return

Type	Description

bool	**True** if the module is connected. **False** if the module is disconnected.

Example 1

```
#include <Microfire_SHT3x.h>
Microfire::SHT3x sht30;

void setup()
{
  Wire.begin();
  if (sht30.begin() != true)
  {
    // Error: the sensor isn't connected
  }
}

void loop () {}
```

connected

Determines if the sensor is connected.

Definition

```
bool connected();
```

Parameters

Parameter	Description
None	

Return

Type	Description
bool	**True** if the module is connected. **False** if the module is disconnected.

Example

```
#include <Microfire_SHT3x.h>
Microfire::SHT3x sht30;

void setup()
{
  Wire.begin();
  sht30.begin();
  if (sht30.connected() != true)
  {
    // Error: the sensor isn't connected
  }
}

void loop () {}
```

measure

Starts a measurement. A measurement takes 20 ms to complete.

Member variables tempC, tempF, vpd_kPa, RH, dew_pointC, dew_pointF, heat_indexC, heat_indexF, wet_bulbC, wet_bulbF, and status are updated.

Definition

```
float measure();
```

Parameters

Parameter	Description
None	

Return

Type	Description
int	An error code for the measurement. It can be one of the following: **0**: no error **1**: no sensor connected **2**: CRC error

Example 1

```
#include <Microfire_SHT3x.h>
Microfire::SHT3x sht30;

void setup()
{
  Wire.begin();
  sht30.begin();
  sht30.measure();
  // The following variables are updated:
  //   sht30.tempC, sht30.tempF, sht30.vpd_kPa, sht30.RH
  //   sht30.dew_pointC, sht30.dew_pointF, sht30.heat_indexC,
sht30.heat_indexF, sht30.wet_bulbC, sht30.wet_bulbF, sht30.status
}

void loop () {}
```