

Ulm Arduino library

Generated by Doxygen 1.9.8



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Data Structure Index</b>	<b>3</b>
2.1 Data Structures	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Data Structure Documentation</b>	<b>7</b>
4.1 SensorData Struct Reference	7
4.1.1 Detailed Description	7
4.2 Ulm_Beginnable Class Reference	7
4.2.1 Detailed Description	8
4.2.2 Member Function Documentation	8
4.2.2.1 begin()	8
4.3 Ulm_LED Class Reference	8
4.3.1 Detailed Description	9
4.3.2 Constructor & Destructor Documentation	9
4.3.2.1 Ulm_LED()	9
4.3.3 Member Function Documentation	9
4.3.3.1 begin()	9
4.3.3.2 off()	10
4.3.3.3 on()	10
4.3.3.4 toggle()	10
4.4 Ulm_LED_BuiltIn Class Reference	10
4.4.1 Detailed Description	11
4.4.2 Constructor & Destructor Documentation	11
4.4.2.1 Ulm_LED_BuiltIn()	11
4.5 Ulm_LSM6DS3 Class Reference	11
4.5.1 Detailed Description	12
4.5.2 Constructor & Destructor Documentation	12
4.5.2.1 Ulm_LSM6DS3()	12
4.5.3 Member Function Documentation	12
4.5.3.1 begin()	12
4.6 Ulm_MS5607 Class Reference	13
4.6.1 Detailed Description	13
4.6.2 Constructor & Destructor Documentation	13
4.6.2.1 Ulm_MS5607()	13
4.6.3 Member Function Documentation	14
4.6.3.1 begin()	14
4.6.3.2 getAltitude()	14
4.6.3.3 getPressure()	14

4.6.3.4 <code>getTemperature()</code> . . . . .	14
4.7 <code>Ulm_OLED_Display</code> Class Reference . . . . .	15
4.7.1 Detailed Description . . . . .	15
4.7.2 Constructor & Destructor Documentation . . . . .	15
4.7.2.1 <code>Ulm_OLED_Display()</code> . . . . .	15
4.7.3 Member Function Documentation . . . . .	15
4.7.3.1 <code>begin()</code> . . . . .	15
4.8 <code>Ulm_RGB_LED</code> Class Reference . . . . .	16
4.8.1 Detailed Description . . . . .	16
4.8.2 Constructor & Destructor Documentation . . . . .	16
4.8.2.1 <code>Ulm_RGB_LED()</code> . . . . .	16
4.8.3 Member Function Documentation . . . . .	17
4.8.3.1 <code>begin()</code> . . . . .	17
4.8.3.2 <code>off()</code> . . . . .	17
4.8.3.3 <code>showError()</code> . . . . .	17
4.8.3.4 <code>showSuccess()</code> . . . . .	17
4.8.3.5 <code>showWarning()</code> . . . . .	17
4.9 <code>Ulm_SDStorage</code> Class Reference . . . . .	18
4.9.1 Detailed Description . . . . .	18
4.9.2 Member Enumeration Documentation . . . . .	19
4.9.2.1 <code>FileType</code> . . . . .	19
4.9.3 Member Function Documentation . . . . .	19
4.9.3.1 <code>begin()</code> . . . . .	19
4.9.3.2 <code>builder()</code> . . . . .	19
4.9.3.3 <code>setCsPin()</code> . . . . .	19
4.9.3.4 <code>setDataFileName()</code> . . . . .	20
4.9.3.5 <code>setDetectPin()</code> . . . . .	20
4.9.3.6 <code>setDirectory()</code> . . . . .	20
4.9.3.7 <code>setExtremeEnvironmentLogic()</code> . . . . .	20
4.9.3.8 <code>setRedundancyFileName()</code> . . . . .	21
4.9.3.9 <code>store()</code> [1/2] . . . . .	21
4.9.3.10 <code>store()</code> [2/2] . . . . .	21
4.10 <code>Ulm_SDStorage::Ulm_SDStorage_Builder</code> Class Reference . . . . .	22
4.10.1 Detailed Description . . . . .	22
4.10.2 Constructor & Destructor Documentation . . . . .	22
4.10.2.1 <code>Ulm_SDStorage_Builder()</code> . . . . .	22
4.10.3 Member Function Documentation . . . . .	22
4.10.3.1 <code>atDirectory()</code> . . . . .	22
4.10.3.2 <code>atPin()</code> . . . . .	23
4.10.3.3 <code>atRootDirectory()</code> . . . . .	23
4.10.3.4 <code>build()</code> . . . . .	23
4.10.3.5 <code>withDataFile()</code> . . . . .	24

4.10.3.6 withDetectPin()	24
4.10.3.7 withLogicForExtremeEnvironments()	24
4.10.3.8 withoutDetectPin()	25
4.10.3.9 withRedundancyFile()	25
4.11 UIm_SDStorage::UIm_SDStorage_CSPinBuilder Class Reference	25
4.11.1 Detailed Description	26
4.11.2 Member Function Documentation	26
4.11.2.1 atPin()	26
4.12 UIm_SDStorage::UIm_SDStorage_DataFileBuilder Class Reference	26
4.12.1 Detailed Description	27
4.12.2 Member Function Documentation	27
4.12.2.1 withDataFile()	27
4.13 UIm_SDStorage::UIm_SDStorage_DetectPinBuilder Class Reference	27
4.13.1 Detailed Description	28
4.13.2 Member Function Documentation	28
4.13.2.1 withDetectPin()	28
4.13.2.2 withoutDetectPin()	29
4.14 UIm_SDStorage::UIm_SDStorage_DirectoryBuilder Class Reference	29
4.14.1 Detailed Description	29
4.14.2 Member Function Documentation	29
4.14.2.1 atDirectory()	29
4.14.2.2 atRootDirectory()	30
4.15 UIm_SDStorage::UIm_Storage_OptionalArgsBuilder Class Reference	30
4.15.1 Detailed Description	31
4.15.2 Member Function Documentation	31
4.15.2.1 build()	31
4.15.2.2 withLogicForExtremeEnvironments()	31
4.15.2.3 withRedundancyFile()	31
4.16 UIm_TemperatureSensor_DS18B20 Class Reference	32
4.16.1 Detailed Description	32
4.16.2 Constructor & Destructor Documentation	32
4.16.2.1 UIm_TemperatureSensor_DS18B20()	32
4.16.3 Member Function Documentation	33
4.16.3.1 begin()	33
4.16.3.2 readTemperature()	33
<b>5 File Documentation</b>	<b>35</b>
5.1 UIm_Beginnable.h	35
5.2 UIm_LED.h	35
5.3 UIm_LED_BuiltIn.h	36
5.4 UIm_LSM6DS3.h	36
5.5 UIm_MS5607.h	36

5.6 UIm_OLED_Display.h . . . . .	37
5.7 UIm_RGB_LED.h . . . . .	37
5.8 UIm_SDStorage.h . . . . .	37
5.9 UIm_TemperatureSensor_DS18B20.h . . . . .	39
5.10 UIm_Weatherballoon.h . . . . .	40
<b>Index</b>	<b>41</b>

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Adafruit_NeoPixel	
Ulm_RGB_LED . . . . .	16
Adafruit_SSD1306	
Ulm_OLED_Display . . . . .	15
DallasTemperature	
Ulm_TemperatureSensor_DS18B20 . . . . .	32
LSM6DS3Class	
Ulm_LSM6DS3 . . . . .	11
MS5x	
Ulm_MS5607 . . . . .	13
SDClass	
Ulm_SDStorage . . . . .	18
SensorData . . . . .	7
Ulm_Beginnable . . . . .	7
Ulm_LED . . . . .	8
Ulm_LED_BuiltIn . . . . .	10
Ulm_LSM6DS3 . . . . .	11
Ulm_MS5607 . . . . .	13
Ulm_OLED_Display . . . . .	15
Ulm_RGB_LED . . . . .	16
Ulm_SDStorage . . . . .	18
Ulm_TemperatureSensor_DS18B20 . . . . .	32
Ulm_SDStorage::Ulm_SDStorage_CSPinBuilder . . . . .	25
Ulm_SDStorage::Ulm_SDStorage_Builder . . . . .	22
Ulm_SDStorage::Ulm_SDStorage_DataFileBuilder . . . . .	26
Ulm_SDStorage::Ulm_SDStorage_Builder . . . . .	22
Ulm_SDStorage::Ulm_SDStorage_DetectPinBuilder . . . . .	27
Ulm_SDStorage::Ulm_SDStorage_Builder . . . . .	22
Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder . . . . .	29
Ulm_SDStorage::Ulm_SDStorage_Builder . . . . .	22
Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder . . . . .	30
Ulm_SDStorage::Ulm_SDStorage_Builder . . . . .	22





## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">SensorData</a>	7
<a href="#">Ulm_Beginnable</a>	7
<a href="#">Ulm_LED</a>	8
<a href="#">Ulm_LED_BuiltIn</a>	10
<a href="#">Ulm_LSM6DS3</a>	11
<a href="#">Ulm_MS5607</a>	13
<a href="#">Ulm_OLED_Display</a>	15
<a href="#">Ulm_RGB_LED</a>	16
<a href="#">Ulm_SDStorage</a>	18
<a href="#">Ulm_SDStorage::Ulm_SDStorage_Builder</a>	22
<a href="#">Ulm_SDStorage::Ulm_SDStorage_CSPinBuilder</a>	25
<a href="#">Ulm_SDStorage::Ulm_SDStorage_DataFileBuilder</a>	26
<a href="#">Ulm_SDStorage::Ulm_SDStorage_DetectPinBuilder</a>	27
<a href="#">Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder</a>	29
<a href="#">Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder</a>	30
<a href="#">Ulm_TemperatureSensor_DS18B20</a>	32



## Chapter 3

# File Index

### 3.1 File List

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

include/Ulm_Beginnable.h . . . . .	35
include/Ulm_LED.h . . . . .	35
include/Ulm_LED_BuiltIn.h . . . . .	36
include/Ulm_LSM6DS3.h . . . . .	36
include/Ulm_MS5607.h . . . . .	36
include/Ulm_OLED_Display.h . . . . .	37
include/Ulm_RGB_LED.h . . . . .	37
include/Ulm_SDStorage.h . . . . .	37
include/Ulm_TemperatureSensor_DS18B20.h . . . . .	39
include/Ulm_Weatherballoon.h . . . . .	40



## Chapter 4

# Data Structure Documentation

### 4.1 SensorData Struct Reference

#### Data Fields

- unsigned long **uptimeMillis**
- float **temperature**
- float **accelerationX**
- float **accelerationY**
- float **accelerationZ**
- float **gyroscopeX**
- float **gyroscopeY**
- float **gyroscopeZ**
- float **temperatureAlt**
- float **pressure**
- float **altitude**

#### 4.1.1 Detailed Description

Structure representing our raw sensor data. All of these values will be sampled and stored to SD card.

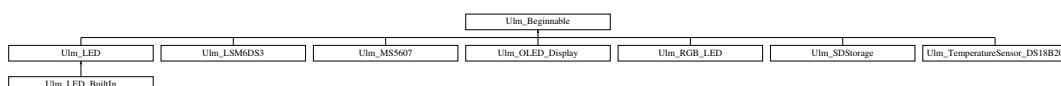
The documentation for this struct was generated from the following file:

- src/main.cpp

### 4.2 Ulm\_Beginnable Class Reference

```
#include <Ulm_Beginnable.h>
```

Inheritance diagram for Ulm\_Beginnable:



## Public Member Functions

- virtual bool `begin()`=0

### 4.2.1 Detailed Description

All electrical components need some sort of initialisation process after the microcontroller receives power. Sadly, the arduino framework does not provide a common name for an init-method and so there are multiple wild names in libraries, like `begin()`, `init()`, `start()` or others.

To at least provide a uniform name within this library, all components will inherit this class and therefore must implement the virtual method `begin(...)`.

### 4.2.2 Member Function Documentation

#### 4.2.2.1 `begin()`

```
virtual bool Ulm_Beginnable::begin ( ) [pure virtual]
```

Initializes the component.

#### Returns

`true` on, and only on, success without ANY error, `false` otherwise.

Implemented in [Ulm\\_LED](#), [Ulm\\_LSM6DS3](#), [Ulm\\_MS5607](#), [Ulm\\_OLED\\_Display](#), [Ulm\\_RGB\\_LED](#), [Ulm\\_SDStorage](#), and [Ulm\\_TemperatureSensor\\_DS18B20](#).

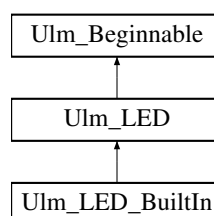
The documentation for this class was generated from the following file:

- `include/Ulm_Beginnable.h`

## 4.3 Ulm\_LED Class Reference

```
#include <Ulm_LED.h>
```

Inheritance diagram for `Ulm_LED`:



## Public Types

- enum **Mode** { **ACTIVE\_LOW** , **ACTIVE\_HIGH** }

## Public Member Functions

- [UIm\\_LED](#) (uint8\_t pin, Mode mode=ACTIVE\_HIGH)
- bool [begin](#) () override
- void [on](#) ()
- void [off](#) ()
- void [toggle](#) () const

## 4.3.1 Detailed Description

This class represents a simple LED.

### Author

Falko Schmidt

### Since

1.0

## 4.3.2 Constructor & Destructor Documentation

### 4.3.2.1 UIm\_LED()

```
UIm_LED::UIm_LED (
    uint8_t pin,
    UIm_LED::Mode mode = ACTIVE_HIGH ) [explicit]
```

Default constructor to create an object of type [UIm\\_LED](#).

### Parameters

<i>pin</i>	the physical pin, which the LED is connected to.
<i>mode</i>	optional argument to configure the activation mode of the LED. This can either be <code>ACTIVE_HIGH</code> , meaning that the LED will turn on, if pin is set to <code>HIGH</code> , or <code>ACTIVE_LOW</code> , meaning that the LED will turn on, if pin is set to <code>LOW</code> . This value defaults to <code>ACTIVE_HIGH</code> .

## 4.3.3 Member Function Documentation

### 4.3.3.1 begin()

```
bool UIm_LED::begin ( ) [override], [virtual]
```

Initializes LED and turns it off.

### Returns

always `true`.

Implements [UIm\\_Beginnable](#).

#### 4.3.3.2 off()

```
void Ulm_LED::off ( )
```

Turns the LED off.

#### 4.3.3.3 on()

```
void Ulm_LED::on ( )
```

Turns the LED on.

#### 4.3.3.4 toggle()

```
void Ulm_LED::toggle ( ) const
```

Toggles the LED. If the LED was on before, then calling this method will turn off the LED, otherwise the LED will be turned on.

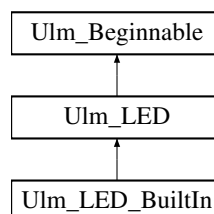
The documentation for this class was generated from the following files:

- include/Ulm\_LED.h
- src/Ulm\_LED.cpp

## 4.4 Ulm\_LED\_BuiltIn Class Reference

```
#include <Ulm_LED_BuiltIn.h>
```

Inheritance diagram for Ulm\_LED\_BuiltIn:



### Public Member Functions

- [Ulm\\_LED\\_BuiltIn](#) ()

### Public Member Functions inherited from [Ulm\\_LED](#)

- [Ulm\\_LED](#) (uint8\_t pin, Mode mode=ACTIVE\_HIGH)
- bool [begin](#) () override
- void [on](#) ()
- void [off](#) ()
- void [toggle](#) () const



## Additional Inherited Members

## Public Types inherited from [UIm\\_LED](#)

- enum **Mode** { **ACTIVE\_LOW** , **ACTIVE\_HIGH** }

### 4.4.1 Detailed Description

This class represents the built-in LED of an Arduino.

Since

1.0

Author

Falko Schmidt

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 UIm\_LED\_BuiltIn()

```
UIm_LED_BuiltIn::UIm_LED_BuiltIn ( )
```

Default constructor. This will construct an object of type `UIm_LED` at pin `LED_BUILTIN`.

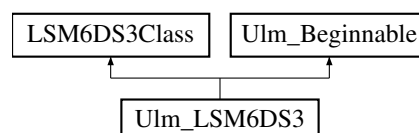
The documentation for this class was generated from the following files:

- `include/UIm_LED_BuiltIn.h`
- `src/UIm_LED_BuiltIn.cpp`

## 4.5 UIm\_LSM6DS3 Class Reference

```
#include <UIm_LSM6DS3.h>
```

Inheritance diagram for `UIm_LSM6DS3`:



## Public Member Functions

- [UIm\\_LSM6DS3](#) (uint8\_t address, TwoWire &wire=Wire)
- bool [begin](#) () override

### 4.5.1 Detailed Description

This class represents IMUs of type LSM6DS3.

#### Author

Falko Schmidt

#### Since

1.0

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 UIm\_LSM6DS3()

```
UIm_LSM6DS3::UIm_LSM6DS3 (
    uint8_t address,
    TwoWire & wire = Wire ) [explicit]
```

Constructor.

#### Parameters

<i>address</i>	the i2c address of the LSM6DS3 sensor.
<i>wire</i>	the i2c interface.

### 4.5.3 Member Function Documentation

#### 4.5.3.1 begin()

```
bool UIm_LSM6DS3::begin ( ) [override], [virtual]
```

Initiates the sensor.

#### Returns

`true` on success, `false` otherwise.

Implements [UIm\\_Beginnable](#).

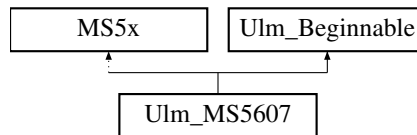
The documentation for this class was generated from the following files:

- include/UIm\_LSM6DS3.h
- src/UIm\_LSM6DS3.cpp

## 4.6 UIm\_MS5607 Class Reference

```
#include <UIm_MS5607.h>
```

Inheritance diagram for UIm\_MS5607:



### Public Member Functions

- [UIm\\_MS5607](#) (int8\_t address)
- bool [begin](#) () override
- float [getTemperature](#) ()
- float [getPressure](#) ()
- float [getAltitude](#) ()

### 4.6.1 Detailed Description

This class represents altitude sensors of type MS5607. The pressure will be measured in mBar, temperature is in Celsius.

#### Author

Falko Schmidt

#### Since

1.0

### 4.6.2 Constructor & Destructor Documentation

#### 4.6.2.1 UIm\_MS5607()

```
UIm_MS5607::UIm_MS5607 (
    int8_t address ) [explicit]
```

Constructor.

#### Parameters

<i>address</i>	the i2c address of MS5607 sensor.
<i>aWire</i>	the i2c connection, defaults to &Wire.

### 4.6.3 Member Function Documentation

#### 4.6.3.1 begin()

```
bool Ulm_MS5607::begin ( ) [override], [virtual]
```

Initializes the sensor.

##### Returns

`true` on success, `false` otherwise.

Implements [Ulm\\_Beginnable](#).

#### 4.6.3.2 getAltitude()

```
float Ulm_MS5607::getAltitude ( )
```

Calculates the altitude using temperature-compensation.

##### Returns

the current altitude in meters.

#### 4.6.3.3 getPressure()

```
float Ulm_MS5607::getPressure ( )
```

Reads the current pressure from the sensor.

##### Returns

the pressure in mBar.

#### 4.6.3.4 getTemperature()

```
float Ulm_MS5607::getTemperature ( )
```

Reads the temperature from the sensor.

##### Returns

the temperature in Celcius.

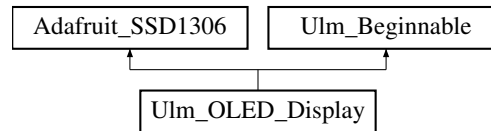
The documentation for this class was generated from the following files:

- include/Ulm\_MS5607.h
- src/Ulm\_MS5607.cpp

## 4.7 UIm\_OLED\_Display Class Reference

```
#include <UIm_OLED_Display.h>
```

Inheritance diagram for UIm\_OLED\_Display:



### Public Member Functions

- [UIm\\_OLED\\_Display\(\)](#)
- `bool` [begin\(\)](#) override

### 4.7.1 Detailed Description

This class represents an OLED with dimensions of 128 x 64 pixels which is connected via i2c at address 0x3C.

#### Author

Falko Schmidt

#### Since

1.0

### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 UIm\_OLED\_Display()

```
UIm_OLED_Display::UIm_OLED_Display ( )
```

Default constructor.

### 4.7.3 Member Function Documentation

#### 4.7.3.1 begin()

```
bool UIm_OLED_Display::begin ( ) [override], [virtual]
```

Initializes the display. After calling this method, the display can be written to.

#### Returns

`true` on success, `false` otherwise.

Implements [UIm\\_Beginnable](#).

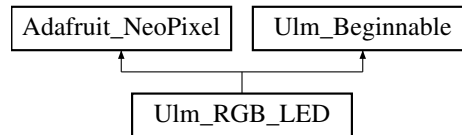
The documentation for this class was generated from the following files:

- `include/UIm_OLED_Display.h`
- `src/UIm_OLED_Display.cpp`

## 4.8 Ulm\_RGB\_LED Class Reference

```
#include <Ulm_RGB_LED.h>
```

Inheritance diagram for Ulm\_RGB\_LED:



### Public Member Functions

- [Ulm\\_RGB\\_LED](#) (int16\_t pin)
- bool [begin](#) () override
- void [off](#) ()
- void [showSuccess](#) ()
- void [showError](#) ()
- void [showWarning](#) ()

### 4.8.1 Detailed Description

This class represents an RGB LED of type WS2812, WS2813 or similar.

#### Author

Falko Schmidt

#### Since

1.0

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 Ulm\_RGB\_LED()

```
Ulm_RGB_LED::Ulm_RGB_LED (
    int16_t pin ) [explicit]
```

Default constructor.

#### Parameters

<i>pin</i>	the pin, which the RGB LED is connected to.
------------	---

## 4.8.3 Member Function Documentation

### 4.8.3.1 begin()

```
bool Ulm_RGB_LED::begin ( ) [override], [virtual]
```

Initiates the RGB LED and turns it off.

#### Returns

always `true`.

Implements [Ulm\\_Beginnable](#).

### 4.8.3.2 off()

```
void Ulm_RGB_LED::off ( )
```

Turns the RGB LED off.

### 4.8.3.3 showError()

```
void Ulm_RGB_LED::showError ( )
```

Shows a red color on the RGB LED.

### 4.8.3.4 showSuccess()

```
void Ulm_RGB_LED::showSuccess ( )
```

Shows a green color on the RGB LED.

### 4.8.3.5 showWarning()

```
void Ulm_RGB_LED::showWarning ( )
```

Shows a yellow color on the RGB LED.

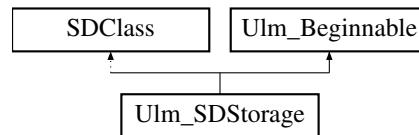
The documentation for this class was generated from the following files:

- `include/Ulm_RGB_LED.h`
- `src/Ulm_RGB_LED.cpp`

## 4.9 Ulm\_SDStorage Class Reference

```
#include <Ulm_SDStorage.h>
```

Inheritance diagram for Ulm\_SDStorage:



### Data Structures

- class [Ulm\\_SDStorage\\_Builder](#)
- class [Ulm\\_SDStorage\\_CSPinBuilder](#)
- class [Ulm\\_SDStorage\\_DataFileBuilder](#)
- class [Ulm\\_SDStorage\\_DetectPinBuilder](#)
- class [Ulm\\_SDStorage\\_DirectoryBuilder](#)
- class [Ulm\\_Storage\\_OptionalArgsBuilder](#)

### Public Types

- enum [FileType](#) { [CSV](#) , [LOG](#) , [TXT](#) }

### Public Member Functions

- bool [begin](#) () override
- bool [store](#) (const String &s)
- bool [store](#) (const char \*str)
- void [setCsPin](#) (uint8\_t csPin)
- void [setDetectPin](#) (int8\_t detectPin)
- void [setDirectory](#) (const String &directory)
- void [setDataFileName](#) (const String &dataFileName)
- void [setRedundancyFileName](#) (const String &redundancyFileName)
- void [setExtremeEnvironmentLogic](#) (bool extremeEnvironmentLogic)

### Static Public Member Functions

- static [Ulm\\_SDStorage\\_Builder](#) [builder](#) ()

#### 4.9.1 Detailed Description

This class represents and SD storage with optional detection pin. Some special additions, like data redundancy and other additional actions for extreme environments are implemented to be used in high altitude balloons.

Please note, that rather than directly instantiating this class, you want to use the according builder.

See also

[Ulm\\_SDStorage::builder](#)

Author

Falko Schmidt

Since

1.0



## 4.9.2 Member Enumeration Documentation

### 4.9.2.1 FileType

enum [UIm\\_SDStorage::FileType](#)

This enum represents valid and supported file types.

## 4.9.3 Member Function Documentation

### 4.9.3.1 begin()

```
bool UIm_SDStorage::begin ( ) [override], [virtual]
```

Initialises the SD card, creates the directory and all needed files. After calling this method, you can directly start storing data into files.

#### Returns

`true` if, and only if, everything went as planned and without ANY error, `false` otherwise.

Implements [UIm\\_Beginnable](#).

### 4.9.3.2 builder()

```
UIm_SDStorage::UIm_SDStorage_Builder UIm_SDStorage::builder ( ) [static]
```

This static method can be used to receive a builder object for storage creation.

#### Returns

a new builder.

#### See also

[UIm\\_SDStorage\\_Builder](#)

### 4.9.3.3 setCsPin()

```
void UIm_SDStorage::setCsPin (
    uint8_t csPin )
```

Sets the chip-select pin of the SD card.

#### Parameters

<i>csPin</i>	the physical pin number, which the SD card is connected to.
--------------	---

#### 4.9.3.4 setDataFileName()

```
void Ulm_SDStorage::setDataFileName (
    const String & dataFileName )
```

Sets the file name of primary data file.

##### Parameters

<i>dataFileName</i>	the data file name.
---------------------	---------------------

#### 4.9.3.5 setDetectPin()

```
void Ulm_SDStorage::setDetectPin (
    int8_t detectPin )
```

Sets the detect pin of the SD card slot. If no detect pin is available or unused, then set the parameter to -1.

##### Parameters

<i>detectPin</i>	the physical pin number, which the SD card slot is connected to.
------------------	--

#### 4.9.3.6 setDirectory()

```
void Ulm_SDStorage::setDirectory (
    const String & directory )
```

Sets the directory, in which all log files are stored. Please note, that subdirectories are not supported. You can either use the root directory ("/") or DIRECT subdirs within the root.

##### Parameters

<i>directory</i>	
------------------	--

#### 4.9.3.7 setExtremeEnvironmentLogic()

```
void Ulm_SDStorage::setExtremeEnvironmentLogic (
    bool extremeEnvironmentLogic )
```

Sets whether additional actions for extreme environments should be taken or not. Please note, that using this feature will result in overhead in flash and execution time.

##### Parameters

<i>extremeEnvironmentLogic</i>	whether additional actions should be taken or not.
--------------------------------	--

#### 4.9.3.8 setRedundancyFileName()

```
void Ulm_SDStorage::setRedundancyFileName (
    const String & redundancyFileName )
```

Sets the file name of the optional redundancy file.

##### Parameters

<i>redundancyFileName</i>	the redundancy file name.
---------------------------	---------------------------

#### 4.9.3.9 store() [1/2]

```
bool Ulm_SDStorage::store (
    const char * str )
```

Stores a const char\* in data and optional redundancy file.

##### Parameters

<i>str</i>	the string of type const char* to be stored.
------------	--

##### Returns

true on success, false otherwise.

#### 4.9.3.10 store() [2/2]

```
bool Ulm_SDStorage::store (
    const String & s )
```

Stores a string in data and optional redundancy file.

##### Parameters

<i>s</i>	the string to be stored.
----------	--------------------------

##### Returns

true on success, false otherwise.

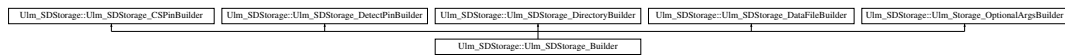
The documentation for this class was generated from the following files:

- include/Ulm\_SDStorage.h
- src/storage/Ulm\_SDStorage.cpp

## 4.10 Ulm\_SDStorage::Ulm\_SDStorage\_Builder Class Reference

```
#include <Ulm_SDStorage.h>
```

Inheritance diagram for Ulm\_SDStorage::Ulm\_SDStorage\_Builder:



### Public Member Functions

- [Ulm\\_SDStorage\\_Builder](#) ()
- [Ulm\\_SDStorage\\_DetectPinBuilder](#) & [atPin](#) (uint8\_t pin) override
- [Ulm\\_SDStorage\\_DirectoryBuilder](#) & [withDetectPin](#) (int8\_t detPin) override
- [Ulm\\_SDStorage\\_DirectoryBuilder](#) & [withoutDetectPin](#) () override
- [Ulm\\_SDStorage\\_DataFileBuilder](#) & [atDirectory](#) (const String &dir) override
- [Ulm\\_SDStorage\\_DataFileBuilder](#) & [atRootDirectory](#) () override
- [Ulm\\_Storage\\_OptionalArgsBuilder](#) & [withDataFile](#) (const String &dataFilePrefix, enum [FileType](#) fileType) override
- [Ulm\\_Storage\\_OptionalArgsBuilder](#) & [withRedundancyFile](#) (const String &redFilePrefix, enum [FileType](#) fileType) override
- [Ulm\\_Storage\\_OptionalArgsBuilder](#) & [withLogicForExtremeEnvironments](#) () override
- [Ulm\\_SDStorage](#) [build](#) () override

### 4.10.1 Detailed Description

This class is the actual builder class for [Ulm\\_SDStorage](#). It is implemented in a step-builder pattern, meaning, that a certain order of method calls is enforced. This enables better bug finding and ensures, that the SD Storage runs with valid and all required values.

#### Author

Falko Schmidt

#### Since

1.0

### 4.10.2 Constructor & Destructor Documentation

#### 4.10.2.1 Ulm\_SDStorage\_Builder()

```
Ulm_SDStorage::Ulm_SDStorage_Builder::Ulm_SDStorage_Builder ( )
```

Default constructor.

### 4.10.3 Member Function Documentation

#### 4.10.3.1 atDirectory()

```
Ulm_SDStorage::Ulm_SDStorage_DataFileBuilder & Ulm_SDStorage::Ulm_SDStorage_Builder::at←
Directory (
    const String & dir ) [override], [virtual]
```

The directory, in which all logs will be created. Please note: The maximum length of the directory name is restricted to 8 chars! Also, the directory will only consist of capital letters, no matter what your input to this method is. Also do not end this string with a '/'. This will be done automatically internally! If the given directory name is invalid, then the root directory will be used instead.

**Parameters**

<i>dir</i>	the directory name.
------------	---------------------

**Returns**

a pointer to this builder to enable method chaining.

Implements [UIm\\_SDStorage::UIm\\_SDStorage\\_DirectoryBuilder](#).

**4.10.3.2 atPin()**

```
UIm_SDStorage::UIm_SDStorage_DetectPinBuilder & UIm_SDStorage::UIm_SDStorage_Builder::atPin (
    uint8_t pin ) [override], [virtual]
```

Configures the physical pin number, which the chip-select pin of the SD card is connected to.

**Parameters**

<i>pin</i>	the physical pin of CS of SD card.
------------	------------------------------------

**Returns**

a pointer to this builder to enable method chaining.

Implements [UIm\\_SDStorage::UIm\\_SDStorage\\_CSPinBuilder](#).

**4.10.3.3 atRootDirectory()**

```
UIm_SDStorage::UIm_SDStorage_DataFileBuilder & UIm_SDStorage::UIm_SDStorage_Builder::atRoot↵
Directory ( ) [override], [virtual]
```

If you do not want to create any additional directory on your SD card, but instead just directly save all files in your top-level directory, then call this method.

**Returns**

a pointer to this builder to enable method chaining.

Implements [UIm\\_SDStorage::UIm\\_SDStorage\\_DirectoryBuilder](#).

**4.10.3.4 build()**

```
UIm_SDStorage UIm_SDStorage::UIm_SDStorage_Builder::build ( ) [override], [virtual]
```

Finally, we want to create our SD Storage! Once you are happy with your configuration, you can call this method, which will return the SD Storage configured the way to wanted to.

**Returns**

the SD Storage according to your arguments.

Implements [UIm\\_SDStorage::UIm\\_Storage\\_OptionalArgsBuilder](#).

#### 4.10.3.5 withDataFile()

```
Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder & Ulm_SDStorage::Ulm_SDStorage_Builder::withDataFile (
    const String & dataFilePrefix,
    enum FileType fileType ) [override], [virtual]
```

Configures the data file name using given prefix and file type. If you for example pass these values: prefix: DATA\_ and fileType: CSV, then the resulting name will be DATA\_000.csv where '000' will automatically be counted up on every start (existing files are not appended to or overwritten).

##### Parameters

<i>dataFilePrefix</i>	the prefix of the data file.
<i>fileType</i>	the type of the file, that will be created.

##### Returns

a pointer to this builder to enable method chaining.

Implements [Ulm\\_SDStorage::Ulm\\_SDStorage\\_DataFileBuilder](#).

#### 4.10.3.6 withDetectPin()

```
Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder & Ulm_SDStorage::Ulm_SDStorage_Builder::withDetectPin (
    int8_t detPin ) [override], [virtual]
```

Configures the physical pin number, which the detection pin of the SD card is connected to. The detection pin will tell you, if an SD card is inserted or not.

##### Parameters

<i>detectPin</i>	the physical pin of the detection pin of your SD card slot.
------------------	---

##### Returns

a pointer to this builder to enable method chaining.

Implements [Ulm\\_SDStorage::Ulm\\_SDStorage\\_DetectPinBuilder](#).

#### 4.10.3.7 withLogicForExtremeEnvironments()

```
Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder & Ulm_SDStorage::Ulm_SDStorage_Builder::withLogicForExtremeEnvironments ( ) [override], [virtual]
```

If you use this class for some extreme environments (cold, hot, high radiation,...), then you can simply call this method to add some additional actions to hopefully safely store your data.

##### Returns

a pointer to this builder to enable method chaining.

Implements [Ulm\\_SDStorage::Ulm\\_Storage\\_OptionalArgsBuilder](#).

#### 4.10.3.8 withoutDetectPin()

```
UIm_SDStorage::UIm_SDStorage_DirectoryBuilder & UIm_SDStorage::UIm_SDStorage_Builder::withoutDetectPin ( ) [override], [virtual]
```

Call this method, if your hardware does not support any detection pin.

##### Returns

a pointer to this builder to enable method chaining.

Implements [UIm\\_SDStorage::UIm\\_SDStorage\\_DetectPinBuilder](#).

#### 4.10.3.9 withRedundancyFile()

```
UIm_SDStorage::UIm_Storage_OptionalArgsBuilder & UIm_SDStorage::UIm_SDStorage_Builder::withRedundancyFile (
    const String & redFilePrefix,
    enum FileType fileType ) [override], [virtual]
```

Configures optional data redundancy by creating a redundancy file, which will contain the same data as the prior defined data file. This feature is very useful to prevent eventual data corruption, but it comes at the cost of higher flash usage and runtime. The time needed to store the data doubles.

##### Parameters

<i>redFilePrefix</i>	the prefix of the redundancy file.
<i>fileType</i>	the type of the file, that will be created.

##### Returns

a pointer to this builder to enable method chaining.

Implements [UIm\\_SDStorage::UIm\\_Storage\\_OptionalArgsBuilder](#).

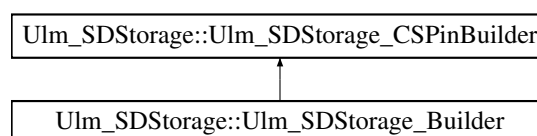
The documentation for this class was generated from the following files:

- include/UIm\_SDStorage.h
- src/storage/UIm\_SDStorage.cpp

## 4.11 UIm\_SDStorage::UIm\_SDStorage\_CSPinBuilder Class Reference

```
#include <UIm_SDStorage.h>
```

Inheritance diagram for UIm\_SDStorage::UIm\_SDStorage\_CSPinBuilder:



## Public Member Functions

- virtual [Ulm\\_SDStorage::Ulm\\_SDStorage\\_DetectPinBuilder](#) & [atPin](#) (uint8\_t pin)=0

### 4.11.1 Detailed Description

This class represents the first mandatory step of building an SD Storage, which is setting the chip-select pin of the SD card.

#### Author

Falko Schmidt

#### Since

1.0

### 4.11.2 Member Function Documentation

#### 4.11.2.1 [atPin\(\)](#)

```
virtual Ulm\_SDStorage::Ulm\_SDStorage\_DetectPinBuilder & Ulm\_SDStorage::Ulm\_SDStorage\_CSPin↔
Builder::atPin (
    uint8_t pin ) [pure virtual]
```

Configures the physical pin number, which the chip-select pin of the SD card is connected to.

#### Parameters

<i>pin</i>	the physical pin of CS of SD card.
------------	------------------------------------

#### Returns

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

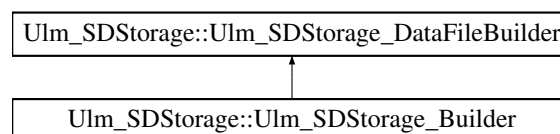
The documentation for this class was generated from the following file:

- include/Ulm\_SDStorage.h

## 4.12 [Ulm\\_SDStorage::Ulm\\_SDStorage\\_DataFileBuilder](#) Class Reference

```
#include <Ulm_SDStorage.h>
```

Inheritance diagram for [Ulm\\_SDStorage::Ulm\\_SDStorage\\_DataFileBuilder](#):





## Public Member Functions

- virtual [Ulm\\_Storage\\_OptionalArgsBuilder](#) & [withDataFile](#) (const String &dataFilePrefix, enum [FileType](#) fileType)=0

### 4.12.1 Detailed Description

This class represents the fourth mandatory step of building an SD Storage, which is configuring the data file name.

#### Author

Falko Schmidt

#### Since

1.0

### 4.12.2 Member Function Documentation

#### 4.12.2.1 withDataFile()

```
virtual Ulm\_Storage\_OptionalArgsBuilder & Ulm\_SDStorage::Ulm\_SDStorage\_DataFileBuilder::withDataFile (
    const String & dataFilePrefix,
    enum FileType fileType ) [pure virtual]
```

Configures the data file name using given prefix and file type.

#### Parameters

<i>dataFilePrefix</i>	the prefix of the data file.
<i>fileType</i>	the type of the file, that will be created.

#### Returns

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

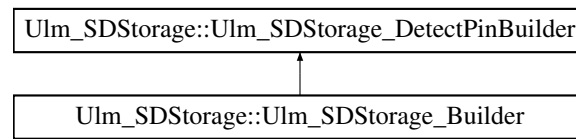
The documentation for this class was generated from the following file:

- include/Ulm\_SDStorage.h

## 4.13 Ulm\_SDStorage::Ulm\_SDStorage\_DetectPinBuilder Class Reference

```
#include <Ulm_SDStorage.h>
```

Inheritance diagram for `Ulm_SDStorage::Ulm_SDStorage_DetectPinBuilder`:



## Public Member Functions

- virtual [Ulm\\_SDStorage::Ulm\\_SDStorage\\_DirectoryBuilder](#) & `withDetectPin` (int8\_t detectPin)=0
- virtual [Ulm\\_SDStorage::Ulm\\_SDStorage\\_DirectoryBuilder](#) & `withoutDetectPin` ()=0

### 4.13.1 Detailed Description

This class represents the second mandatory step of building an SD Storage, which is setting a detection pin. To also support other hardware-structures, it is possible to not set any detection pin.

#### Author

Falko Schmidt

#### Since

1.0

### 4.13.2 Member Function Documentation

#### 4.13.2.1 `withDetectPin()`

```
virtual Ulm\_SDStorage::Ulm\_SDStorage\_DirectoryBuilder & Ulm_SDStorage::Ulm_SDStorage_DetectPinBuilder::withDetectPin (
    int8_t detectPin ) [pure virtual]
```

Configures the physical pin number, which the detection pin of the SD card is connected to. The detection pin will tell you, if an SD card is inserted or not.

#### Parameters

<i>detectPin</i>	the physical pin of the detection pin of your SD card slot.
------------------	---

#### Returns

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

#### 4.13.2.2 withoutDetectPin()

```
virtual Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder & Ulm_SDStorage::Ulm_SDStorage_Detect↔
PinBuilder::withoutDetectPin ( ) [pure virtual]
```

Call this method, if your hardware does not support any detection pin.

##### Returns

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

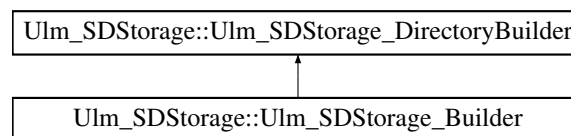
The documentation for this class was generated from the following file:

- include/Ulm\_SDStorage.h

## 4.14 Ulm\_SDStorage::Ulm\_SDStorage\_DirectoryBuilder Class Reference

```
#include <Ulm_SDStorage.h>
```

Inheritance diagram for Ulm\_SDStorage::Ulm\_SDStorage\_DirectoryBuilder:



### Public Member Functions

- virtual [Ulm\\_SDStorage\\_DataFileBuilder](#) & [atDirectory](#) (const String &dir)=0
- virtual [Ulm\\_SDStorage\\_DataFileBuilder](#) & [atRootDirectory](#) ()=0

#### 4.14.1 Detailed Description

This class represents the third mandatory step of building an SD Storage, which is configuring a directory, in which all files will be created.

##### Author

Falko Schmidt

##### Since

1.0

#### 4.14.2 Member Function Documentation

##### 4.14.2.1 atDirectory()

```
virtual Ulm_SDStorage_DataFileBuilder & Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder::at↔
Directory (
    const String & dir ) [pure virtual]
```

Configures the directory, in which the data and redundancy files will be stored. This allows you to have a somehow ordered structure on your SD card.

**Parameters**

<i>dir</i>	the directory name.
------------	---------------------

**Returns**

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

**4.14.2.2 atRootDirectory()**

```
virtual Ulm\_SDStorage\_DataFileBuilder & Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder::at↔
RootDirectory ( ) [pure virtual]
```

If you do not want to create any additional directory on your SD card, but instead just directly save all files in your top-level directory, then call this method.

**Returns**

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

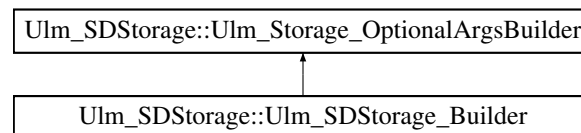
The documentation for this class was generated from the following file:

- include/Ulm\_SDStorage.h

## 4.15 Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder Class Reference

```
#include <Ulm_SDStorage.h>
```

Inheritance diagram for Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder:

**Public Member Functions**

- virtual [Ulm\\_SDStorage::Ulm\\_Storage\\_OptionalArgsBuilder](#) & withRedundancyFile (const String &redFile↔ Prefix, enum [FileType](#) fileType)=0
- virtual [Ulm\\_SDStorage::Ulm\\_Storage\\_OptionalArgsBuilder](#) & withLogicForExtremeEnvironments ()=0
- virtual [Ulm\\_SDStorage](#) build ()=0

### 4.15.1 Detailed Description

This class represents the last step(s) of building an SD storage. All of these features can be used optionally - there is no need to!

#### Author

Falko Schmidt

#### Since

1.0

### 4.15.2 Member Function Documentation

#### 4.15.2.1 build()

```
virtual Ulm\_SDStorage Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder::build ( ) [pure virtual]
```

Finally, we want to create our SD Storage! Once you are happy with your configuration, you can call this method, which will return the SD Storage configured the way to wanted to.

#### Returns

the SD Storage according to your arguments.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

#### 4.15.2.2 withLogicForExtremeEnvironments()

```
virtual Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder & Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder::withLogicForExtremeEnvironments ( ) [pure virtual]
```

If you use this class for some extreme environments (cold, hot, high radiation,...), then you can simply call this method to add some additional actions to hopefully safely store your data.

#### Returns

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

#### 4.15.2.3 withRedundancyFile()

```
virtual Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder & Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder::withRedundancyFile (
    const String & redFilePrefix,
    enum FileType fileType ) [pure virtual]
```

Configures optional data redundancy by creating a redundancy file, which will contain the same data as the prior defined data file. This feature is very useful to prevent eventual data corruption, but it comes at the cost of higher flash usage and runtime. The time needed to store the data doubles.

## Parameters

<i>redFilePrefix</i>	the prefix of the redundancy file.
<i>fileType</i>	the type of the file, that will be created.

## Returns

a pointer to this builder to enable method chaining.

Implemented in [Ulm\\_SDStorage::Ulm\\_SDStorage\\_Builder](#).

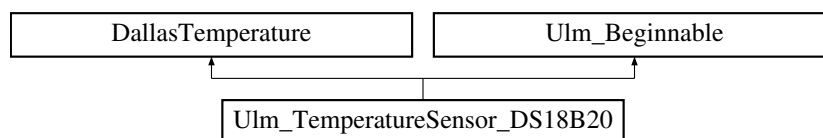
The documentation for this class was generated from the following file:

- include/Ulm\_SDStorage.h

## 4.16 Ulm\_TemperatureSensor\_DS18B20 Class Reference

```
#include <Ulm_TemperatureSensor_DS18B20.h>
```

Inheritance diagram for Ulm\_TemperatureSensor\_DS18B20:



## Public Member Functions

- [Ulm\\_TemperatureSensor\\_DS18B20](#) (uint8\_t pin)
- bool [begin](#) () override
- float [readTemperature](#) ()

### 4.16.1 Detailed Description

This class represents temperature sensors of type DS18B20.

## Author

Falko Schmidt

## Since

1.0

### 4.16.2 Constructor & Destructor Documentation

#### 4.16.2.1 Ulm\_TemperatureSensor\_DS18B20()

```
Ulm_TemperatureSensor_DS18B20::Ulm_TemperatureSensor_DS18B20 (
    uint8_t pin ) [explicit]
```

Constructor.

## Parameters

<i>pin</i>	the pin, which the sensor is connected to.
------------	--

### 4.16.3 Member Function Documentation

#### 4.16.3.1 begin()

```
bool UIm_TemperatureSensor_DS18B20::begin ( ) [override], [virtual]
```

Initializes the sensor and configures the readout configuration.

## Returns

always `true`.

Implements [UIm\\_Beginnable](#).

#### 4.16.3.2 readTemperature()

```
float UIm_TemperatureSensor_DS18B20::readTemperature ( )
```

Reads out the sensor and returns the current temperature.

## Returns

the current temperature in Celsius.

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

- `include/UIm_TemperatureSensor_DS18B20.h`
- `src/UIm_TemperatureSensor_DS18B20.cpp`





# Chapter 5

## File Documentation

### 5.1 Ulm\_Beginnable.h

```
00001 //
00002 // Created by Falko Alrik Schmidt on 11.10.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_BEGINNABLE_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_BEGINNABLE_H
00007
00018 class Ulm_Beginnable {
00019
00020 public:
00025     virtual bool begin() = 0;
00026
00027 };
00028
00029
00030 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_BEGINNABLE_H
```

### 5.2 Ulm\_LED.h

```
00001 //
00002 // Created by Falko Alrik Schmidt on 11.10.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LED_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LED_H
00007
00008
00009 #include "Ulm_Beginnable.h"
00010
00017 class Ulm_LED : public Ulm_Beginnable {
00018
00019 public:
00020     enum Mode {
00021         ACTIVE_LOW,
00022         ACTIVE_HIGH
00023     };
00024
00025 private:
00029     const uint8_t pin;
00030
00034     const enum Mode mode;
00035
00036 public:
00045     explicit Ulm_LED(uint8_t pin, Mode mode = ACTIVE_HIGH);
00046
00051     bool begin() override;
00052
00056     void on();
00057
00061     void off();
00062
00067     void toggle() const;
00068 };
00069
00070
00071 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LED_H
```

## 5.3 Ulm\_LED\_BuiltIn.h

```
00001 //
00002 // Created by Falko Alrik Schmidt on 11.10.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LED_BUILTIN_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LED_BUILTIN_H
00007
00008 #include "Ulm_LED.h"
00009
00016 class Ulm_LED_BuiltIn : public Ulm_LED {
00017
00018 public:
00023     Ulm_LED_BuiltIn();
00024
00025 };
00026
00027
00028 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LED_BUILTIN_H
```

## 5.4 Ulm\_LSM6DS3.h

```
00001 //
00002 // Created by Falko Alrik Schmidt on 26.09.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LSM6DS3_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LSM6DS3_H
00007
00008 #include <Wire.h>
00009 #include "Arduino_LSM6DS3.h"
00010 #include "Ulm_Beginnable.h"
00011
00018 class Ulm_LSM6DS3 : public LSM6DS3Class, public Ulm_Beginnable {
00019
00020 public:
00026     explicit Ulm_LSM6DS3(uint8_t address, TwoWire &wire = Wire);
00027
00032     bool begin() override;
00033 };
00034
00035
00036 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_LSM6DS3_H
```

## 5.5 Ulm\_MS5607.h

```
00001 //
00002 // Created by Falko Alrik Schmidt on 26.09.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_MS5607_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_MS5607_H
00007
00008 #include <Wire.h>
00009 #include "Ulm_Beginnable.h"
00010 #include "MS5x.h"
00011
00019 class Ulm_MS5607 : MS5x, public Ulm_Beginnable {
00020
00021 public:
00027     explicit Ulm_MS5607(int8_t address);
00028
00033     bool begin() override;
00034
00039     float getTemperature();
00040
00045     float getPressure();
00046
00051     float getAltitude();
00052 };
00053
00054
00055 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_MS5607_H
```

## 5.6 Ulm\_OLED\_Display.h

```

00001 //
00002 // Created by Falko Alrik Schmidt on 19.09.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_OLED_DISPLAY_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_OLED_DISPLAY_H
00007
00008 #include "Adafruit_SSD1306.h"
00009 #include "Ulm_Beginnable.h"
00010
00011 class Ulm_OLED_Display : public Adafruit_SSD1306, public Ulm_Beginnable {
00012 public:
00013     Ulm_OLED_Display();
00014
00015     bool begin() override;
00016 };
00017
00018 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_OLED_DISPLAY_H

```

## 5.7 Ulm\_RGB\_LED.h

```

00001 //
00002 // Created by Falko Alrik Schmidt on 19.09.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_RGB_LED_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_RGB_LED_H
00007
00008 #include "Adafruit_NeoPixel.h"
00009 #include "Ulm_Beginnable.h"
00010
00011 class Ulm_RGB_LED : public Adafruit_NeoPixel, public Ulm_Beginnable {
00012 public:
00013     explicit Ulm_RGB_LED(int16_t pin);
00014
00015     bool begin() override;
00016
00017     void off();
00018
00019     void showSuccess();
00020
00021     void showError();
00022
00023     void showWarning();
00024 private:
00025     inline void showColor(uint32_t color);
00026 };
00027
00028 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_RGB_LED_H

```

## 5.8 Ulm\_SDStorage.h

```

00001 //
00002 // Created by Falko Alrik Schmidt on 09.09.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_SDSTORAGE_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_SDSTORAGE_H
00007
00008 #include <Arduino.h>
00009 #include <SD.h>
00010 #include "Ulm_Beginnable.h"
00011
00012 class Ulm_SDStorage : SDClass, public Ulm_Beginnable {
00013 public:
00014     enum FileType {
00015         CSV,
00016         LOG,
00017         TXT
00018     };

```

```

00035     };
00036
00037 public:
00038     class Ulm_SDStorage_Builder;
00039     class Ulm_SDStorage_CSPinBuilder;
00040     class Ulm_SDStorage_DetectPinBuilder;
00041     class Ulm_SDStorage_DirectoryBuilder;
00042     class Ulm_SDStorage_DataFileBuilder;
00043     class Ulm_Storage_OptionalArgsBuilder;
00044
00045 private:
00050     uint8_t csPin;
00051
00056     int8_t detectPin;
00057
00061     String directory;
00062
00066     String dataFileName;
00067
00071     String redundancyFileName;
00072
00077     bool extremeEnvironmentLogic;
00078
00079
00080 private:
00084     Ulm_SDStorage();
00085
00086
00087 public:
00093     static Ulm_SDStorage_Builder builder();
00094
00101     bool begin() override;
00102
00108     bool store(const String& s);
00109
00115     bool store(const char* str);
00116
00121     void setCSPin(uint8_t csPin);
00122
00128     void setDetectPin(int8_t detectPin);
00129
00136     void setDirectory(const String &directory);
00137
00142     void setDataFileName(const String &dataFileName);
00143
00148     void setRedundancyFileName(const String &redundancyFileName);
00149
00157     void setExtremeEnvironmentLogic(bool extremeEnvironmentLogic);
00158
00159
00160 private:
00170     static String determineGenericLogFileName(const String& prefix, Ulm_SDStorage::FileType fileType);
00171
00186     bool determineCurrentLogFileName(const String& dir, String& fileName);
00187
00195     size_t storeToFile(const char *str, const String& dir, const String& fileName);
00196
00201     bool hasRedundancyFile();
00202 };
00203
00204
00205 //
00206 // -----
00207 // FIRST STEP - MANDATORY - SET CS PIN OF SD CARD
00208 // -----
00215 class Ulm_SDStorage::Ulm_SDStorage_CSPinBuilder {
00216 public:
00222     virtual Ulm_SDStorage::Ulm_SDStorage_DetectPinBuilder& atPin(uint8_t pin) = 0;
00223 };
00224
00225
00234 class Ulm_SDStorage::Ulm_SDStorage_DetectPinBuilder {
00235 public:
00242     virtual Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder& withDetectPin(int8_t detectPin) = 0;
00243
00248     virtual Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder& withoutDetectPin() = 0;
00249 };
00250
00251
00259 class Ulm_SDStorage::Ulm_SDStorage_DirectoryBuilder {
00260 public:
00267     virtual Ulm_SDStorage_DataFileBuilder& atDirectory(const String& dir) = 0;
00268
00274     virtual Ulm_SDStorage_DataFileBuilder& atRootDirectory() = 0;
00275 };
00276

```

```

00277
00285 class Ulm_SDStorage::Ulm_SDStorage_DataFileBuilder {
00286 public:
00293     virtual Ulm_Storage_OptionalArgsBuilder& withDataFile(const String& dataFilePrefix, enum FileType
        fileType) = 0;
00294 };
00295
00296
00304 class Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder {
00305 public:
00315     virtual Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder&
00316     withRedundancyFile(const String& redFilePrefix, enum FileType fileType) = 0;
00317
00324     virtual Ulm_SDStorage::Ulm_Storage_OptionalArgsBuilder&
00325     withLogicForExtremeEnvironments() = 0;
00326
00332     virtual Ulm_SDStorage build() = 0;
00333 };
00334
00335
00344 class Ulm_SDStorage::Ulm_SDStorage_Builder : public Ulm_SDStorage_CSPinBuilder,
00345                                             public Ulm_SDStorage_DetectPinBuilder,
00346                                             public Ulm_SDStorage_DirectoryBuilder,
00347                                             public Ulm_SDStorage_DataFileBuilder,
00348                                             public Ulm_Storage_OptionalArgsBuilder {
00349 private:
00358     Ulm_SDStorage sdStorage;
00359 public:
00360     Ulm_SDStorage_Builder();
00365 public:
00372     Ulm_SDStorage_DetectPinBuilder &atPin(uint8_t pin) override;
00373
00380     Ulm_SDStorage_DirectoryBuilder &withDetectPin(int8_t detPin) override;
00381
00386     Ulm_SDStorage_DirectoryBuilder &withoutDetectPin() override;
00387
00399     Ulm_SDStorage_DataFileBuilder &atDirectory(const String& dir) override;
00400
00406     Ulm_SDStorage_DataFileBuilder &atRootDirectory() override;
00407
00417     Ulm_Storage_OptionalArgsBuilder &withDataFile(const String& dataFilePrefix, enum FileType
        fileType) override;
00418
00428     Ulm_Storage_OptionalArgsBuilder &withRedundancyFile(const String& redFilePrefix, enum FileType
        fileType) override;
00429
00436     Ulm_Storage_OptionalArgsBuilder &withLogicForExtremeEnvironments() override;
00437
00443     Ulm_SDStorage build() override;
00444 };
00445
00446 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_SDSTORAGE_H

```

## 5.9 Ulm\_TemperatureSensor\_DS18B20.h

```

00001 //
00002 // Created by Falko Alrik Schmidt on 19.09.23.
00003 //
00004
00005 #ifndef UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_TEMPERATURESENSOR_DS18B20_H
00006 #define UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_TEMPERATURESENSOR_DS18B20_H
00007
00008
00009 #include "DallasTemperature.h"
00010 #include "Ulm_Beginnable.h"
00011
00018 class Ulm_TemperatureSensor_DS18B20 : public DallasTemperature, public Ulm_Beginnable {
00019 private:
00020     OneWire oneWire;
00025
00026 public:
00032     explicit Ulm_TemperatureSensor_DS18B20(uint8_t pin);
00033
00038     bool begin() override;
00039
00044     float readTemperature();
00045 };
00046
00047
00048 #endif //UULM_WEATHERBALOON_ARDUINO_FRAMEWORK_TEST_ULM_TEMPERATURESENSOR_DS18B20_H

```

## 5.10 Ulm\_Weatherballoon.h

```
00001 // =====
00002 // This file is created due to the Arduino library guidelines.
00003 // Including this file will include all relevant header files from this
00004 // library.
00005 //
00006 // Falko Schmidt, 2023
00007 // =====
00008 #ifndef ULM_WEATHERBALLOON_ULM_WEATHERBALLOON_H
00009 #define ULM_WEATHERBALLOON_ULM_WEATHERBALLOON_H
00010
00011 // -----
00012 // LED
00013 // -----
00014 #include "Ulm_LED.h"
00015 #include "Ulm_LED_BuiltIn.h"
00016 #include "Ulm_RGB_LED.h"
00017
00018 // -----
00019 // SENSORS
00020 // -----
00021 #include "Ulm_LSM6DS3.h"
00022 #include "Ulm_MS5607.h"
00023 #include "Ulm_TemperatureSensor_DS18B20.h"
00024
00025 // -----
00026 // DISPLAYS
00027 // -----
00028 #include "Ulm_OLED_Display.h"
00029
00030 // -----
00031 // STORAGE
00032 // -----
00033 #include "Ulm_SDStorage.h"
00034
00035 #endif //ULM_WEATHERBALLOON_ULM_WEATHERBALLOON_H
```

# Index

atDirectory  
    Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [22](#)  
    Ulm\_SDStorage::Ulm\_SDStorage\_DirectoryBuilder, on  
        [29](#)  
atPin  
    Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [23](#)  
    Ulm\_SDStorage::Ulm\_SDStorage\_CSPinBuilder,  
        [26](#)  
atRootDirectory  
    Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [23](#)  
    Ulm\_SDStorage::Ulm\_SDStorage\_DirectoryBuilder,  
        [30](#)  
begin  
    Ulm\_Beginnable, [8](#)  
    Ulm\_LED, [9](#)  
    Ulm\_LSM6DS3, [12](#)  
    Ulm\_MS5607, [14](#)  
    Ulm\_OLED\_Display, [15](#)  
    Ulm\_RGB\_LED, [17](#)  
    Ulm\_SDStorage, [19](#)  
    Ulm\_TemperatureSensor\_DS18B20, [33](#)  
build  
    Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [23](#)  
    Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder,  
        [31](#)  
builder  
    Ulm\_SDStorage, [19](#)  
FileType  
    Ulm\_SDStorage, [19](#)  
getAltitude  
    Ulm\_MS5607, [14](#)  
getPressure  
    Ulm\_MS5607, [14](#)  
getTemperature  
    Ulm\_MS5607, [14](#)  
include/Ulm\_Beginnable.h, [35](#)  
include/Ulm\_LED.h, [35](#)  
include/Ulm\_LED\_BuiltIn.h, [36](#)  
include/Ulm\_LSM6DS3.h, [36](#)  
include/Ulm\_MS5607.h, [36](#)  
include/Ulm\_OLED\_Display.h, [37](#)  
include/Ulm\_RGB\_LED.h, [37](#)  
include/Ulm\_SDStorage.h, [37](#)  
include/Ulm\_TemperatureSensor\_DS18B20.h, [39](#)  
include/Ulm\_Weatherballoon.h, [40](#)  
off  
    Ulm\_LED, [9](#)  
    Ulm\_RGB\_LED, [17](#)  
    Ulm\_LED, [10](#)  
readTemperature  
    Ulm\_TemperatureSensor\_DS18B20, [33](#)  
SensorData, [7](#)  
setCsPin  
    Ulm\_SDStorage, [19](#)  
setDataFileName  
    Ulm\_SDStorage, [20](#)  
setDetectPin  
    Ulm\_SDStorage, [20](#)  
setDirectory  
    Ulm\_SDStorage, [20](#)  
setExtremeEnvironmentLogic  
    Ulm\_SDStorage, [20](#)  
setRedundancyFileName  
    Ulm\_SDStorage, [20](#)  
showError  
    Ulm\_RGB\_LED, [17](#)  
showSuccess  
    Ulm\_RGB\_LED, [17](#)  
showWarning  
    Ulm\_RGB\_LED, [17](#)  
store  
    Ulm\_SDStorage, [21](#)  
toggle  
    Ulm\_LED, [10](#)  
Ulm\_Beginnable, [7](#)  
    begin, [8](#)  
Ulm\_LED, [8](#)  
    begin, [9](#)  
    off, [9](#)  
    on, [10](#)  
    toggle, [10](#)  
    Ulm\_LED, [9](#)  
Ulm\_LED\_BuiltIn, [10](#)  
    Ulm\_LED\_BuiltIn, [11](#)  
Ulm\_LSM6DS3, [11](#)  
    begin, [12](#)  
    Ulm\_LSM6DS3, [12](#)  
Ulm\_MS5607, [13](#)  
    begin, [14](#)  
    getAltitude, [14](#)  
    getPressure, [14](#)

- getTemperature, [14](#)
  - Ulm\_MS5607, [13](#)
- Ulm\_OLED\_Display, [15](#)
  - begin, [15](#)
  - Ulm\_OLED\_Display, [15](#)
- Ulm\_RGB\_LED, [16](#)
  - begin, [17](#)
  - off, [17](#)
  - showError, [17](#)
  - showSuccess, [17](#)
  - showWarning, [17](#)
  - Ulm\_RGB\_LED, [16](#)
- Ulm\_SDStorage, [18](#)
  - begin, [19](#)
  - builder, [19](#)
  - FileType, [19](#)
  - setCsPin, [19](#)
  - setDataFileName, [20](#)
  - setDetectPin, [20](#)
  - setDirectory, [20](#)
  - setExtremeEnvironmentLogic, [20](#)
  - setRedundancyFileName, [20](#)
  - store, [21](#)
- Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [22](#)
  - atDirectory, [22](#)
  - atPin, [23](#)
  - atRootDirectory, [23](#)
  - build, [23](#)
  - Ulm\_SDStorage\_Builder, [22](#)
  - withDataFile, [23](#)
  - withDetectPin, [24](#)
  - withLogicForExtremeEnvironments, [24](#)
  - withoutDetectPin, [24](#)
  - withRedundancyFile, [25](#)
- Ulm\_SDStorage::Ulm\_SDStorage\_CSPinBuilder, [25](#)
  - atPin, [26](#)
- Ulm\_SDStorage::Ulm\_SDStorage\_DataFileBuilder, [26](#)
  - withDataFile, [27](#)
- Ulm\_SDStorage::Ulm\_SDStorage\_DetectPinBuilder, [27](#)
  - withDetectPin, [28](#)
  - withoutDetectPin, [28](#)
- Ulm\_SDStorage::Ulm\_SDStorage\_DirectoryBuilder, [29](#)
  - atDirectory, [29](#)
  - atRootDirectory, [30](#)
- Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder, [30](#)
  - build, [31](#)
  - withLogicForExtremeEnvironments, [31](#)
  - withRedundancyFile, [31](#)
- Ulm\_SDStorage\_Builder
  - Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [22](#)
- Ulm\_TemperatureSensor\_DS18B20, [32](#)
  - begin, [33](#)
  - readTemperature, [33](#)
  - Ulm\_TemperatureSensor\_DS18B20, [32](#)
- withDataFile
  - Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [23](#)
  - Ulm\_SDStorage::Ulm\_SDStorage\_DataFileBuilder, [27](#)
- withDetectPin
  - Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [24](#)
  - Ulm\_SDStorage::Ulm\_SDStorage\_DetectPinBuilder, [28](#)
- withLogicForExtremeEnvironments
  - Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [24](#)
  - Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder, [31](#)
- withoutDetectPin
  - Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [24](#)
  - Ulm\_SDStorage::Ulm\_SDStorage\_DetectPinBuilder, [28](#)
- withRedundancyFile
  - Ulm\_SDStorage::Ulm\_SDStorage\_Builder, [25](#)
  - Ulm\_SDStorage::Ulm\_Storage\_OptionalArgsBuilder, [31](#)