Arduino Weather Sensor Driver

Generated by Doxygen 1.8.9.1

Wed Aug 19 2015 01:07:08

Contents

1 Hierarchical Index **1**

1.1 Class Hierarchy 1

2 Class Index **2**

2.1 Class List 2

3 File Index **2**

3.1 File List 2

4 Class Documentation **2**

4.1 WeatherSensor Class Reference 2

4.1.1 Detailed Description 3

4.1.2 Member Function Documentation 3

4.2 WeatherSensorDHT11 Class Reference 3

4.2.1 Detailed Description 4

4.2.2 Member Enumeration Documentation 5

4.2.3 Constructor & Destructor Documentation 5

4.2.4 Member Function Documentation 5

4.2.5 Member Data Documentation 6

5 File Documentation **7**

5.1 WeatherSensor.cpp File Reference 7

5.1.1 Macro Definition Documentation 7

5.2 WeatherSensor.cpp 8

5.3 WeatherSensor.h File Reference 8

5.4 WeatherSensor.h 8

5.5 WeatherSensorDHT11.cpp File Reference 9

5.5.1 Macro Definition Documentation 9

5.6 WeatherSensorDHT11.cpp 9

5.7 WeatherSensorDHT11.h File Reference 11

5.7.1 Macro Definition Documentation 12

5.8 WeatherSensorDHT11.h 12

Index **15**

1 Hierarchical Index

1.1 Class Hierarchy

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

WeatherSensor **2**

WeatherSensorDHT11 3

2 Class Index

2.1 Class List

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

WeatherSensor
Arduino - Weather sensor 2

WeatherSensorDHT11 3

3 File Index

3.1 File List

Here is a list of all files with brief descriptions:

WeatherSensor.cpp 7

WeatherSensor.h 8

WeatherSensorDHT11.cpp 9

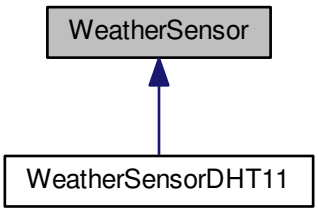
WeatherSensorDHT11.h 11

4 Class Documentation

4.1 WeatherSensor Class Reference

`#include <WeatherSensor.h>`

Inheritance diagram for WeatherSensor:



Public Member Functions

- virtual float getHumidity ()=0
- virtual float getTemperature ()=0

4.1.1 Detailed Description

Arduino - Weather sensor.

WeatherSensor.h

The abstract class for the weather sensors.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 14 of file WeatherSensor.h.

4.1.2 Member Function Documentation

4.1.2.1 virtual float WeatherSensor::getHumidity ( ) [pure virtual]

Returns the air humidity.

The air humidity.

Implemented in WeatherSensorDHT11.

4.1.2.2 virtual float WeatherSensor::getTemperature ( ) [pure virtual]

Returns the temperature.

The temperature.

Implemented in WeatherSensorDHT11.

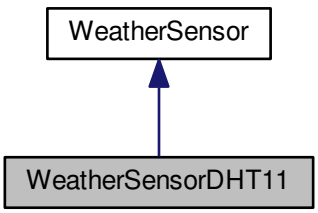
The documentation for this class was generated from the following file:

- WeatherSensor.h

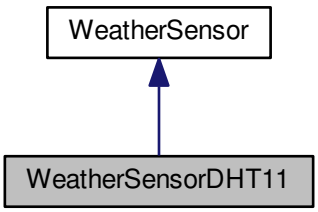
4.2 WeatherSensorDHT11 Class Reference

`#include <WeatherSensorDHT11.h>`

Inheritance diagram for WeatherSensorDHT11:



Collaboration diagram for WeatherSensorDHT11:



Public Types

- enum Code {
  SUCCESS = 0x00, ERROR_CHECKSUM = 0x01, ERROR_CONDITION1_NOT_MET = 0x02, ERROR_C↩
  ONDITION2_NOT_MET = 0x04,
  ERROR_CONDITION3_NOT_MET = 0x08 }

Public Member Functions

- WeatherSensorDHT11 (unsigned char dataPin)
- float getHumidity ()
- float getTemperature ()
- Code getLastCode ()

Private Member Functions

- Code readPackage (unsigned char ∗buf)
- float makeFloat (unsigned char ∗buf)
- bool isAvailable ()
- unsigned char read ()
- unsigned long waitFor (unsigned char pin, unsigned char state, unsigned long timeout)

Private Attributes

- unsigned char buf [5]
- unsigned char dataPin
- unsigned long lastReadTime
- float lastHumity
- float lastTemperature
- Code lastCode

4.2.1 Detailed Description

Definition at line 37 of file WeatherSensorDHT11.h.

4.2.2 Member Enumeration Documentation

4.2.2.1 enum WeatherSensorDHT11::Code

Enumerator

SUCCESS

ERROR_CHECKSUM

ERROR_CONDITION1_NOT_MET

ERROR_CONDITION2_NOT_MET

ERROR_CONDITION3_NOT_MET

Definition at line 66 of file WeatherSensorDHT11.h.

4.2.3 Constructor & Destructor Documentation

4.2.3.1 WeatherSensorDHT11::WeatherSensorDHT11 (unsigned char *dataPin*)

Public constructor.

Parameters

dataPin	The data pin.

Definition at line 16 of file WeatherSensorDHT11.cpp.

4.2.4 Member Function Documentation

4.2.4.1 float WeatherSensorDHT11::getHumidity ( ) [virtual]

Returns the air humidity.

The air humidity.

Implements WeatherSensor.

Definition at line 26 of file WeatherSensorDHT11.cpp.

4.2.4.2 Code WeatherSensorDHT11::getLastCode ( ) [inline]

Gets the last read code.

Returns

Definition at line 100 of file WeatherSensorDHT11.h.

4.2.4.3 float WeatherSensorDHT11::getTemperature ( ) [virtual]

Returns the temperature.

The temperature.

Implements WeatherSensor.

Definition at line 33 of file WeatherSensorDHT11.cpp.

4.2.4.4 bool WeatherSensorDHT11::isAvailable ( ) [private]

Return true if the device is available for a read operation.

Returns

Definition at line 40 of file WeatherSensorDHT11.cpp.

4.2.4.5 float WeatherSensorDHT11::makeFloat (unsigned char * *buf*) [private]

Makes a float number from a read data.

Parameters

buf	The read data.

Returns

The float conversion.

Definition at line 113 of file WeatherSensorDHT11.cpp.

4.2.4.6 unsigned char WeatherSensorDHT11::read ( ) [private]

Reads a byte from the device.

Returns

Definition at line 81 of file WeatherSensorDHT11.cpp.

4.2.4.7 WeatherSensorDHT11::Code WeatherSensorDHT11::readPackage (unsigned char * *buf*) [private]

Reads 5 bytes from the device.

1st byte: Integer part of the humidity. 2nd byte: Fractional part of the humidity. 3th byte: Integer part of the temperature. 4th byte: Fractional part of the temperature. 5th byte: Checksum.

Returns

The error code.

Definition at line 44 of file WeatherSensorDHT11.cpp.

4.2.4.8 unsigned long WeatherSensorDHT11::waitFor (unsigned char *pin,* unsigned char *state,* unsigned long *timeout*) [private]

Waits for a given state on a input pin.

Definition at line 96 of file WeatherSensorDHT11.cpp.

4.2.5 Member Data Documentation

4.2.5.1 unsigned char WeatherSensorDHT11::buf[5] [private]

A internal buffer.

Definition at line 42 of file WeatherSensorDHT11.h.

4.2.5.2 unsigned char WeatherSensorDHT11::dataPin [private]

The data pin.

Definition at line 47 of file WeatherSensorDHT11.h.

4.2.5.3 Code WeatherSensorDHT11::lastCode [private]

Definition at line 106 of file WeatherSensorDHT11.h.

4.2.5.4 float WeatherSensorDHT11::lastHumity [private]

The last humity.

Definition at line 57 of file WeatherSensorDHT11.h.

4.2.5.5 unsigned long WeatherSensorDHT11::lastReadTime [private]

The last read time.

Definition at line 52 of file WeatherSensorDHT11.h.

4.2.5.6 float WeatherSensorDHT11::lastTemperature [private]

The last temperature.

Definition at line 62 of file WeatherSensorDHT11.h.

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

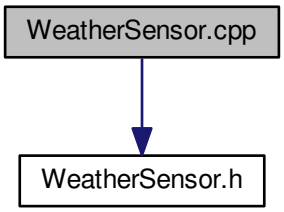
- WeatherSensorDHT11.h
- WeatherSensorDHT11.cpp

5 File Documentation

5.1 WeatherSensor.cpp File Reference

```
#include "WeatherSensor.h"
```

Include dependency graph for WeatherSensor.cpp:



Macros

- #define __ARDUINO_DRIVER_WEATHER_SENSOR_CPP__ 1

5.1.1 Macro Definition Documentation

5.1.1.1 #define __ARDUINO_DRIVER_WEATHER_SENSOR_CPP__ 1

Arduino - Weather sensor.

WeatherSensor.h

The abstract class for the color sensors.

Author

Dalmir da Silva dalmirdasilva@gmail.com

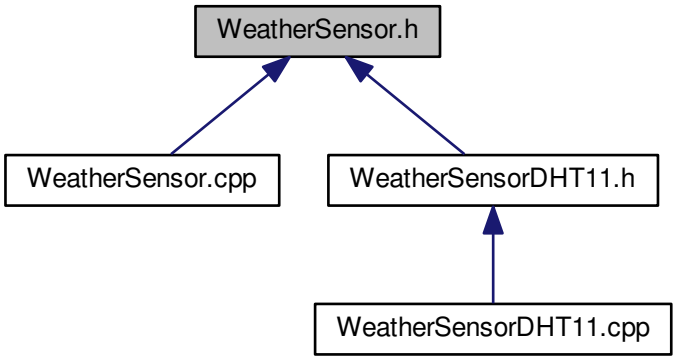
Definition at line 12 of file WeatherSensor.cpp.

5.2 WeatherSensor.cpp

```
00001
00011 #ifndef __ARDUINO_DRIVER_WEATHER_SENSOR_CPP__
00012 #define __ARDUINO_DRIVER_WEATHER_SENSOR_CPP__ 1
00013
00014 #include "WeatherSensor.h"
00015
00016 #endif /* __ARDUINO_DRIVER_WEATHER_SENSOR_CPP__ */
```

5.3 WeatherSensor.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

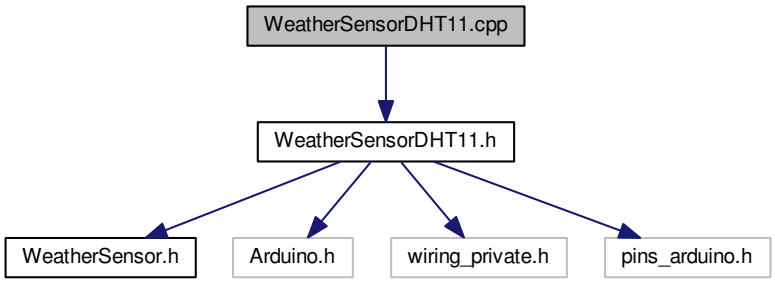
- class WeatherSensor

5.4 WeatherSensor.h

```
00001
00011 #ifndef __ARDUINO_DRIVER_WEATHER_SENSOR_H__
00012 #define __ARDUINO_DRIVER_WEATHER_SENSOR_H__ 1
00013
00014 class WeatherSensor {
00015 public:
00016
00022     virtual float getHumidity() = 0;
00023
00029     virtual float getTemperature() = 0;
00030 };
00031
00032 #endif /* __ARDUINO_DRIVER_WEATHER_SENSOR_H__ */
```

5.5 WeatherSensorDHT11.cpp File Reference

`#include "WeatherSensorDHT11.h"`
Include dependency graph for WeatherSensorDHT11.cpp:



Macros

- #define __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_CPP__ 1

5.5.1 Macro Definition Documentation

5.5.1.1 #define __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_CPP__ 1

Arduino - Weather sensor.

WeatherSensorDHT11.h

The class for the DHT11 weather sensor.

Author

Dalmir da Silva dalmirdasilva@gmail.com

Definition at line 12 of file WeatherSensorDHT11.cpp.

5.6 WeatherSensorDHT11.cpp

```
00001
00011 #ifndef __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_CPP__
00012 #define __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_CPP__ 1
00013
00014 #include "WeatherSensorDHT11.h"
00015
00016 WeatherSensorDHT11::WeatherSensorDHT11(unsigned char dataPin) :
00017         dataPin(dataPin) {
00018     lastReadTime = millis() - WEATHER_MILLIS_BETWEEN_READS;
00019     lastHumity = 0.0;
00020     lastTemperature = 0.0;
00021     lastCode = SUCCESS;
00022     pinMode(dataPin, OUTPUT);
00023     digitalWrite(dataPin, HIGH);
00024 }
00025
00026 float WeatherSensorDHT11::getHumidity() {
00027     if (isAvailable()) {
00028         lastCode = readPackage(this->buf);
00029     }
00030     return lastHumity;
00031 }
```

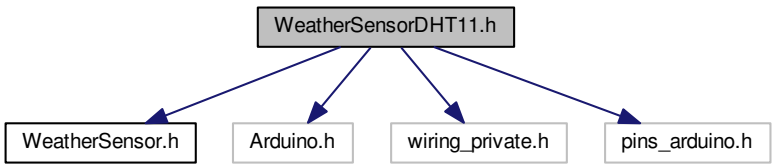
```
00032
00033 float WeatherSensorDHT11::getTemperature() {
00034     if (isAvailable()) {
00035         lastCode = readPackage(this->buf);
00036     }
00037     return lastTemperature;
00038 }
00039
00040 bool WeatherSensorDHT11::isAvailable() {
00041     return (lastReadTime + WEATHER_MILLIS_BETWEEN_READS) <= millis(
      );
00042 }
00043
00044 WeatherSensorDHT11::Code WeatherSensorDHT11::readPackage
      (unsigned char *buf) {
00045     unsigned char in, checkSum;
00046     unsigned long m = 0;
00047     while (!isAvailable())
00048         ;
00049     lastReadTime = millis();
00050     pinMode(dataPin, OUTPUT);
00051     digitalWrite(dataPin, LOW);
00052     delay(WEATHER_REQUEST_LOW_LENGTH);
00053     digitalWrite(dataPin, HIGH);
00054     pinMode(dataPin, INPUT);
00055     m = waitFor(dataPin, LOW, WEATHER_REQUEST_HIGH_LENGTH);
00056     if (m == 0) {
00057         return ERROR_CONDITION1_NOT_MET;
00058     }
00059     m = waitFor(dataPin, HIGH, WEATHER_RESPONSE_LOW_LENGTH);
00060     if (m == 0) {
00061         return ERROR_CONDITION2_NOT_MET;
00062     }
00063     m = waitFor(dataPin, LOW, WEATHER_RESPONSE_HIGH_LENGTH);
00064     if (m == 0) {
00065         return ERROR_CONDITION3_NOT_MET;
00066     }
00067     for (unsigned char i = 0; i < 5; i++) {
00068         buf[i] = read();
00069     }
00070     pinMode(dataPin, OUTPUT);
00071     digitalWrite(dataPin, HIGH);
00072     checkSum = buf[0] + buf[1] + buf[2] + buf[3];
00073     if (buf[4] != checkSum) {
00074         return ERROR_CHECKSUM;
00075     }
00076     lastHumity = makeFloat(this->buf);
00077     lastTemperature = makeFloat(&(this->buf[2]));
00078     return SUCCESS;
00079 }
00080
00081 unsigned char WeatherSensorDHT11::read() {
00082     unsigned char d = 0;
00083     unsigned long m = 0;
00084     for (unsigned char i = 0; i < 8; i++) {
00085         m = waitFor(dataPin, HIGH, WEATHER_START_TRANSMIT_LENGTH
      );
00086         if (m > 0) {
00087             m = waitFor(dataPin, LOW, WEATHER_ONE_LENGTH);
00088             if (m > WEATHER_ZERO_LENGTH) {
00089                 d |= (1 << (7 - i));
00090             }
00091         }
00092     }
00093     return d;
00094 }
00095
00096 unsigned long WeatherSensorDHT11::waitFor(unsigned char pin,
00097         unsigned char state, unsigned long timeout) {
00098     unsigned char bit = digitalPinToBitMask(pin);
00099     unsigned char port = digitalPinToPort(pin);
00100     unsigned char stateMask = (state ? bit : 0);
00101     unsigned long width = 0;
00102     unsigned long numloops = 0;
00103     unsigned long maxloops = microsecondsToClockCycles(timeout) / 16;
00104     while ((*portInputRegister(port) & bit) != stateMask) {
00105         if (numloops++ == maxloops) {
00106             return 0;
00107         }
00108         width++;
00109     }
00110     return clockCyclesToMicroseconds(width * 21 + 8);
00111 }
00112
00113 float WeatherSensorDHT11::makeFloat(unsigned char *buf) {
00114     float f = 0.0;
00115     f += *buf;
```

```
00116     f += *(buf + 1) / 256.0;
00117     return f;
00118 }
00119
00120 #endif /* __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_CPP__ */
```

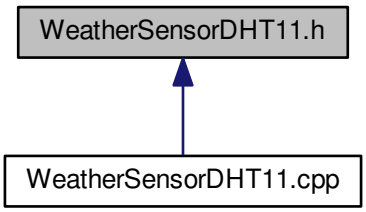
5.7 WeatherSensorDHT11.h File Reference

```
#include <WeatherSensor.h>
#include <Arduino.h>
#include <wiring_private.h>
#include <pins_arduino.h>
```

Include dependency graph for WeatherSensorDHT11.h:



This graph shows which files directly or indirectly include this file:



Classes

- class WeatherSensorDHT11

Macros

- #define WEATHER_MILLIS_BETWEEN_READS 800
- #define WEATHER_REQUEST_LOW_LENGTH 20
- #define WEATHER_REQUEST_HIGH_LENGTH 40
- #define WEATHER_RESPONSE_LOW_LENGTH 80
- #define WEATHER_RESPONSE_HIGH_LENGTH 80
- #define WEATHER_START_TRANSMIT_LENGTH 50
- #define WEATHER_ZERO_LENGTH 40
- #define WEATHER_ONE_LENGTH 70

5.7.1 Macro Definition Documentation

5.7.1.1 #define WEATHER_MILLIS_BETWEEN_READS 800

Arduino - Weather sensor.

WeatherSensorDHT11.h

The class for the DHT11 weather sensor.

Author

Dalmir da Silva dalmirdasilva@gmail.com The time between consecutive reads.

Parameters

dataPin	

Definition at line 24 of file WeatherSensorDHT11.h.

5.7.1.2 #define WEATHER_ONE_LENGTH 70

Definition at line 35 of file WeatherSensorDHT11.h.

5.7.1.3 #define WEATHER_REQUEST_HIGH_LENGTH 40

Definition at line 27 of file WeatherSensorDHT11.h.

5.7.1.4 #define WEATHER_REQUEST_LOW_LENGTH 20

Definition at line 26 of file WeatherSensorDHT11.h.

5.7.1.5 #define WEATHER_RESPONSE_HIGH_LENGTH 80

Definition at line 30 of file WeatherSensorDHT11.h.

5.7.1.6 #define WEATHER_RESPONSE_LOW_LENGTH 80

Definition at line 29 of file WeatherSensorDHT11.h.

5.7.1.7 #define WEATHER_START_TRANSMIT_LENGTH 50

Definition at line 32 of file WeatherSensorDHT11.h.

5.7.1.8 #define WEATHER_ZERO_LENGTH 40

Definition at line 34 of file WeatherSensorDHT11.h.

5.8 WeatherSensorDHT11.h

```
00001
00011 #ifndef __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_H__
00012 #define __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_H__ 1
00013
00014 #include <WeatherSensor.h>
00015 #include <Arduino.h>
00016 #include <wiring_private.h>
00017 #include <pins_arduino.h>
00018
00024 #define WEATHER_MILLIS_BETWEEN_READS            800
00025
00026 #define WEATHER_REQUEST_LOW_LENGTH              20
00027 #define WEATHER_REQUEST_HIGH_LENGTH             40
00028
00029 #define WEATHER_RESPONSE_LOW_LENGTH             80
00030 #define WEATHER_RESPONSE_HIGH_LENGTH            80
00031
00032 #define WEATHER_START_TRANSMIT_LENGTH           50
```

```
00033
00034 #define WEATHER_ZERO_LENGTH                     40
00035 #define WEATHER_ONE_LENGTH                      70
00036
00037 class WeatherSensorDHT11 : public WeatherSensor {
00038
00042     unsigned char buf[5];
00043
00047     unsigned char dataPin;
00048
00052     unsigned long lastReadTime;
00053
00057     float lastHumity;
00058
00062     float lastTemperature;
00063
00064 public:
00065
00066     enum Code {
00067         SUCCESS = 0x00,
00068         ERROR_CHECKSUM = 0x01,
00069         ERROR_CONDITION1_NOT_MET = 0x02,
00070         ERROR_CONDITION2_NOT_MET = 0x04,
00071         ERROR_CONDITION3_NOT_MET = 0x08
00072     };
00073
00079     WeatherSensorDHT11(unsigned char dataPin);
00080
00086     float getHumidity();
00087
00093     float getTemperature();
00094
00100     Code getLastCode() {
00101         return lastCode;
00102     }
00103
00104 private:
00105
00106     Code lastCode;
00107
00119     Code readPackage(unsigned char *buf);
00120
00127     float makeFloat(unsigned char *buf);
00128
00134     bool isAvailable();
00135
00141     unsigned char read();
00142
00146     unsigned long waitFor(unsigned char pin, unsigned char state, unsigned long timeout);
00147 };
00148
00149 #endif /* __ARDUINO_DRIVER_WEATHER_SENSOR_DHT11_H__ */
```

Index

__ARDUINO_DRIVER_WEATHER_SENSOR_CPP_↩
_
WeatherSensor.cpp, 7
__ARDUINO_DRIVER_WEATHER_SENSOR_DH↩
T11_CPP__
WeatherSensorDHT11.cpp, 9

buf
WeatherSensorDHT11, 6

Code
WeatherSensorDHT11, 5

dataPin
WeatherSensorDHT11, 6

ERROR_CHECKSUM
WeatherSensorDHT11, 5
ERROR_CONDITION1_NOT_MET
WeatherSensorDHT11, 5
ERROR_CONDITION2_NOT_MET
WeatherSensorDHT11, 5
ERROR_CONDITION3_NOT_MET
WeatherSensorDHT11, 5

getHumidity
WeatherSensor, 3
WeatherSensorDHT11, 5
getLastCode
WeatherSensorDHT11, 5
getTemperature
WeatherSensor, 3
WeatherSensorDHT11, 5

isAvailable
WeatherSensorDHT11, 5

lastCode
WeatherSensorDHT11, 6
lastHumity
WeatherSensorDHT11, 7
lastReadTime
WeatherSensorDHT11, 7
lastTemperature
WeatherSensorDHT11, 7

makeFloat
WeatherSensorDHT11, 6

read
WeatherSensorDHT11, 6
readPackage
WeatherSensorDHT11, 6

SUCCESS
WeatherSensorDHT11, 5

WEATHER_MILLIS_BETWEEN_READS
WeatherSensorDHT11.h, 12
WEATHER_ONE_LENGTH
WeatherSensorDHT11.h, 12
WEATHER_REQUEST_HIGH_LENGTH
WeatherSensorDHT11.h, 12
WEATHER_REQUEST_LOW_LENGTH
WeatherSensorDHT11.h, 12
WEATHER_RESPONSE_HIGH_LENGTH
WeatherSensorDHT11.h, 12
WEATHER_RESPONSE_LOW_LENGTH
WeatherSensorDHT11.h, 12
WEATHER_START_TRANSMIT_LENGTH
WeatherSensorDHT11.h, 12
WEATHER_ZERO_LENGTH
WeatherSensorDHT11.h, 12
waitFor
WeatherSensorDHT11, 6
WeatherSensor, 2
getHumidity, 3
getTemperature, 3
WeatherSensor.cpp, 7, 8
__ARDUINO_DRIVER_WEATHER_SENSOR_↩
CPP__, 7
WeatherSensor.h, 8
WeatherSensorDHT11, 3
buf, 6
Code, 5
dataPin, 6
ERROR_CHECKSUM, 5
ERROR_CONDITION1_NOT_MET, 5
ERROR_CONDITION2_NOT_MET, 5
ERROR_CONDITION3_NOT_MET, 5
getHumidity, 5
getLastCode, 5
getTemperature, 5
isAvailable, 5
lastCode, 6
lastHumity, 7
lastReadTime, 7
lastTemperature, 7
makeFloat, 6
read, 6
readPackage, 6
SUCCESS, 5
waitFor, 6
WeatherSensorDHT11, 5
WeatherSensorDHT11.cpp, 9
__ARDUINO_DRIVER_WEATHER_SENSOR_↩
DHT11_CPP__, 9
WeatherSensorDHT11.h, 11, 12
WEATHER_MILLIS_BETWEEN_READS, 12
WEATHER_ONE_LENGTH, 12
WEATHER_REQUEST_HIGH_LENGTH, 12
WEATHER_REQUEST_LOW_LENGTH, 12
WEATHER_RESPONSE_HIGH_LENGTH, 12

WEATHER_RESPONSE_LOW_LENGTH, 12
WEATHER_START_TRANSMIT_LENGTH, 12
WEATHER_ZERO_LENGTH, 12