DHT11_DEVICE class

Designed and written by
Stephen Harris

Macros

Macro: typeof(n) CTOFAHR(n)

Description: Converts any generic integer/float from Celsius to Fahrenheit.
Parameters:
n (input): Celsius value (can be of any native type)
Return Value:
Returns converted value in the same type of parameter “n”
Notes:
Thanks to the property of macros, this works on both floats and integers of just about any type

---

Macro: typeof(n) CTOKELV(n)

Description: Converts any generic integer/float from Celsius to Kelvin.
Parameters:
n (input): Celsius value (can be of any native type)
Return Value:
Returns converted value in the same type of parameter “n”
Notes:
Thanks to the property of macros, this works on both floats and integers of just about any type

---

Public methods

Method: DHT11_DEVICE();
DHT11_DEVICE(uint8_t dpin);

Description: Constructors for the DHT11_DEVICE class

Parameters:

uint8_t **dpin** (optional): the value for the digital data pin on the arduino board that will be sending and reading data from the DHT11 ‘out’ pin

---

Method: float readtemp();

Description: Returns the temperature (Celsius) from the DHT11 device.
Parameters: None
Return Value:

- On success, returns a floating point value
- On failure, returns NaN

Notes:

Due to device limitations, if you call this function within a second of another similar read call such as readtemp() or readhumid(), it will return the value(s) obtained from the previous call.

The method will fail if either there is a checksum failure, or a timeout failure. Both are typically issues with the connection with the device, or the device itself

---

Method: float readhumid();

Description: Returns the relative humidity (RH) from the DHT11 device
Parameters: None
Return Value:

- On success, returns a floating point value
- On failure, returns NaN

Notes:

Due to device limitations, if you call this function within a second of another similar read call such as readtemp() or readhumid(), it will return the value(s) obtained from the previous call.

The method will fail if either there is a checksum failure, or a timeout failure. Both are typically issues with the connection with the device, or the device itself

---

Method: static const char* geterr(int8_t err);

Description: Returns a const char* (string) associated with the error code passed as an argument. The returned char* represents a string of the error's name.

Parameters:

int8_t **err** (input): The DHT11_DEVICE error code

Return Value:

- Will always return a valid pointer to a string literal successfully
- If you pass an invalid error code to this function, it will return a string pointer to the literal "DHT11_UNKNOWN_ERROR"

Notes:

As of the current implementation of the DHT11_DEVICE class, there are no public methods that return the error code from the device polls. So until further notice, it is only useful within class methods.

Method: void debug();

Description: Runs code for testing and debugging the device.

Parameters: None

Return Value: None

Notes:

This function is typically used by the class developer. While it might print useful information in a public context, it generally isn't meant to be used outside of the private interface of the class.

Method: int8_t isready();

Description: Returns a nonzero value if the DHT11 device is not ready to be polled from for new values

Parameters: None

Return Value:

- If device is ready, returns 0
- If device is not ready, returns a nonzero value

Notes:

You do not need to call this method in order to use readtemp() or readhumid(), as they both default to known values if called before the device is ready to be re-read.

The HDT11 device is typically ready between 1000ms intervals.

Private methods

(For class developers)

Method: int8_t sendstart();

Description: A helper function for updateword(). Sends the start signal to the DHT11 device, after which the device will start sending data

Parameters: None

Return Value:

- On success, returns 0
- On failure, returns a nonzero value

Notes:

There is currently no way for this function to fail. So as of the current implementation it will always return zero

---

Method: int8_t updateword();

Description: Initiates handshake and data transfer from the DHT11 device, and stores it in the word (DHT11_WORD) member of the class.

Parameters: None

Return Value:

DHT11_SUCCESS (0) - Device signal start and data transfer was successful

DHT11_ERR_TIMEOUT (1) - Timeout occurred when waiting for the next bit from the device

DHT11_ERR_CHECKSUM (2) - DHT11_WORD checksum failed to match

DHT11_ERR_NOT_READY (3) - Either the device has not reached the STARTUP_DELAY time from device power-on, or was called within SAMPLE_DELAY from a previous updateword() or graphsig() call

Notes:

If for any reason an error occurs, the word (DHT11_WORD) member is not overwritten, preserving the values from the last successful updateword() call. This way methods such as readtemp() or readhumid() are able to reuse old values if called at a higher rate than allowed by the SAMPLE_DELAY

Method: void graphsig();

Description: Utilizes arduino's Serial Plotter tool in order to graph the entire signal sent by the DHT11 device (for debugging)

Parameters: None

Return Value: None

Method: void formatword();

Description: A helper function for updateword() that will set the 'temp' and 'humid' float members to the formatted values specified in the word (DHT11_WORD) member

Parameters: None

Return Value: None
