UnoLib documentation

dht.pas

version 11/10/2025

The dht.pas module contains implementation of TDHT class, providing the interface for the popular, low-cost DHT series of temperature and humidity sensors (including DHT11, DHT21/AM2301, DHT22/AM2302, DHT33, and DHT44). It provides fixed-point values for both temperature (in Celsius) and humidity (in %RH).

Since FPC for AVR does not allow dynamic object creation, an object of type TDHT declared in the var section is ready to use and does not require the use of a constructor.

TDHT methods
`procedure` **`Init`**`(pin, model: UInt8);` Initializes the DHT object and specifies the digital data pin the sensor is connected to and the model of the sensor. *Parameters* *pin* – digital data pin of the sensor. *model* – constant specifying the model of the sensor. It can be one of the following values: DHT11, DHT12, DHT21, DHT22, DHT33, DHT44. Note: Originally constructor of the class.
`function` **`Read`**`: Int8;` Initiates the full communication cycle with the sensor, reads the 40 bits of data, performs the checksum verification, and updates the internal temperature and humidity variables. Returns an error code (e.g., DHTLIB_OK, DHTLIB_ERROR_CHECKSUM, etc.).
`function` **`BitsPtr`**`: UInt8P0;` Returns the pointer to the internal bits data of 5 bytes.
`function` **`HexBits`**`: shortstring;` Returns hexadecimal representation of internal bits data as a shortstring (256 bytes).
TDHT fields

`Humidity: TFix16;`
`Temperature: TFix16;`
 	
Humidity (in %RH) and temperature (in Celsius) stored as fixed point numbers (see TFix16 type description)

Example

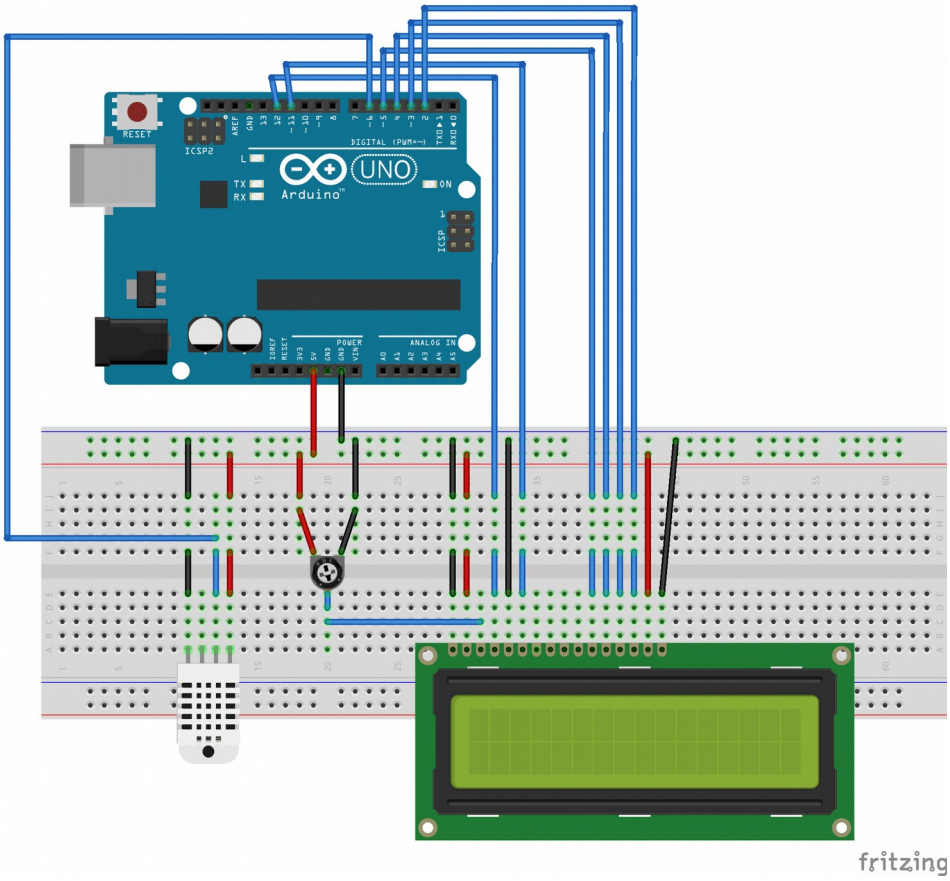
This example demonstrate a sensor-to-display interface program. Its main purpose is to read temperature and humidity data from a DHT11 sensor and display the results on a 16x2 Liquid Crystal Display (LCD, based on Hitachi HD44780 controller chipset). The code loads the necessary units: fix16 (for fixed-point math), timer (for delays), liquidcrystal (for the LCD), and dht (for the sensor).

To initialize the Liquid Crystal Display (LC) it uses a initialization pattern *LC.Init*(12, 11, 5, 4, 3, 2), which sets the Arduino pins connected to the LCD control and data lines (RS, Enable, D4, D5, D6, D7). Finally, *LC._begin*(16, 2) configures the display size to 16 columns and 2 rows. To initialize the DHT object (_DHT) it uses *_DHT.Init*(DHT11Pin, DHT11), which specifies that the sensor is connected to Pin 6 and identifies the sensor model as DHT11.

The program enters an infinite loop (while True do), where clears the LCD screen (LC.Clear) and attempts to read data from the DHT sensor (*_DHT.Read*). If the reading was successful (*DHT_OK*) :

- displays the label ‘T:’ and converts the fixed-point temperature value (*_DHT.Temperature*) to a string, showing 1 decimal place (*Fix16ToStr*), and writes it to the LCD.

- displays the label 'H:' and converts the fixed-point humidity value (*_DHT.Humidity*) to a string, showing 1 decimal place, and writes it to the LCD.



Component connections:

Component	Signal	Arduino Pin
LCD	RS	12
	Enable (E)	11
	Data 4 (D4)	5
	Data 5 (D5)	4
	Data 6 (D6)	3
	Data 7 (D7)	2
DHT11 Sensor	Data Pin	6
Common	Power (VCC)	+5V
	Ground (GND)	GND

LCD power supply and contrast setup:

LCD Pin	Connection	Purpose
VCC (Pin 2)	+5V on the Arduino	Powers the display module.
GND (Pin 1, 5, 16)	GND on the Arduino	Connects to ground (includes Read/Write control).
V0 (Pin 3 - Contrast)	Center pin of the 10kΩ potentiometer	Adjusts screen contrast. The two outer pins of the potentiometer connect to +5V and GND.
LED+ (Pin 15)	+5V on the Arduino	Powers the backlight LED (some displays include a built-in current-limiting resistor).
LED- (Pin 16)	GND on the Arduino	Ground for the backlight.

LCD control and data pin connections:

LCD Pin	Code Name	Arduino Pin	Purpose
RS (Register Select)	rs (1st parameter)	Digital Pin 12	Switches between command and data mode.
E (Enable)	enable (2nd parameter)	Digital Pin 11	Strobe signal to execute commands.
D4 (Data Pin 4)	d4 (3rd parameter)	Digital Pin 5	Data line for the 4-bit communication.
D5 (Data Pin 5)	d5 (4th parameter)	Digital Pin 4	Data line for the 4-bit communication.
D6 (Data Pin 6)	d6 (5th parameter)	Digital Pin 3	Data line for the 4-bit

			communication.
D7 (Data Pin 7)	d7 (6th parameter)	Digital Pin 2	Data line for the 4-bit communication.

```
program TestDHT11;

{$IF NOT (DEFINED(atmega328p) or DEFINED(arduinouno) or
DEFINED(arduinonano) or DEFINED(fpc_mcu_atmega328p) or
DEFINED(fpc_mcu_arduinouno) or DEFINED(fpc_mcu_arduinonano))}
 {$Fatal Invalid controller type, expected: atmega328p, arduinouno, or
arduinonano}
{$ENDIF}

{$mode objfpc}

uses
  fix16, timer, liquidcrystal, dht;

const
  DHT11Pin = 6;

var
  s: shortstring;
begin
  LC.Init(12, 11, 5, 4, 3, 2);
  LC._begin(16, 2);

  _DHT.Init(DHT11Pin, DHT11);
  s:='';

  while True do
  begin
    LC.Clear;
    if _DHT.Read=DHT_OK then
    begin
      LC.Write('T:');
      Fix16ToStr(_DHT.Temperature, 1, s);
      LC.Write(s);
      LC.Write(' H:');
      Fix16ToStr(_DHT.Humidity, 1, s);
      LC.Write(s);
    end
    else
      LC.Write('ERR');

    Delay(2000);
  end;
end.
```