Arduino Code:

* /\*
* digital humidity.ino
* copyright (c) 2016 avinash.pudale@gmail.com
* All rights reserved.
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* \* Redistributions of source code must retain the above copyright notice, this list
* of conditions and the following disclaimer.
*
* \* Redistributions in binary form must reproduce the above copyright notice, this
* list of conditions and the following disclaimer in the documentation and/or other
* materials provided with the distribution.
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
* CONTRIBUTORS "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. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
* CONTRIBUTORS 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 THIS SOFTWARE, EVEN IF
* ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
* \*/
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* // First we include the libraries
* #include <OneWire.h>
* #include <DallasTemperature.h>
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* // Data wire is plugged into pin 2 on the Arduino
* #define ONE\_WIRE\_BUS 2
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* // Setup a oneWire instance to communicate with any OneWire devices
* // (not just Maxim/Dallas temperature ICs)
* OneWire oneWire(ONE\_WIRE\_BUS);
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* // Pass our oneWire reference to Dallas Temperature.
* DallasTemperature sensors(&oneWire);
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* //LCD
* #include <LiquidCrystal.h>// include the library code:
* const int rs = 10, en = 9, d4 = 8, d5 = 7, d6 = 6, d7 = 5;// with the arduino pin number it is connected to
* LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // initialize the library by associating any needed LCD interface pin
* // these constants won't change. But you can change the size of
* // your LCD using them:
* const int numRows = 4;
* const int numCols = 20;
* const int Vccpin1 = 3;
* const int Vccpin2 = 11;
* const int Gndpin2 = 12;
* void setup()
* {
* pinMode(Vccpin1, OUTPUT);
* pinMode(Vccpin2, OUTPUT);
* pinMode(Gndpin2, OUTPUT);
*
* // start serial port
* Serial.begin(9600);
* Serial.println("Dallas Temperature IC Control Library Demo");
* // Start up the library
* lcd.begin(numCols, numRows);
* sensors.begin();
* }
* void loop()
* {
*
* digitalWrite(Vccpin1, HIGH);
*
* digitalWrite(Vccpin2, HIGH);
*
* digitalWrite(Gndpin2, LOW);
* // call sensors.requestTemperatures() to issue a global temperature
* // request to all devices on the bus
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* sensors.requestTemperatures(); // Send the command to get temperature readings
* float t = sensors.getTempCByIndex(0);
* Serial.print("Temperature");
* Serial.println(t); // Why "byIndex"?
* lcd.clear();
* lcd.setCursor(0,0);//cursur position
* lcd.print("Temp// Print a message to the LCD.
* lcd.setCursor(7,0);//cursur position
* lcd.print(t);// Print a message to the LCD.
* // Serial.println(sensors.getTempCByIndex(0));
* // You can have more than one DS18B20 on the same bus.
* // 0 refers to the first IC on the wire
* //delay(4000);
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* float u = sensors.getTempCByIndex(1);
* float d = ((17.502\*t)/(240.97+t)); // Equation for Humidity
* float e = 2.7183;
* float g = 6.112\*(pow(e,d));
* float j = ((17.502\*u)/(240.97+u));
* float k = 2.7183;
* float m = 6.112\*(pow(k,j));
* float n = (m-(0.6687\*(1+0.00115\*u)\*(t-u)))/g\*100; //Humidity upto 2 digit
*
* Serial.print("Humidity");
* Serial.println(n); // Print a message to serial
* lcd.setCursor(0,1);//cursur position
* lcd.print("Humidity/ Print a message to the LCD.
* lcd.setCursor(9,1);//cursur position
* lcd.print(n);// Print a message to the LCD.
* delay(10000);
* lcd.clear();
* }