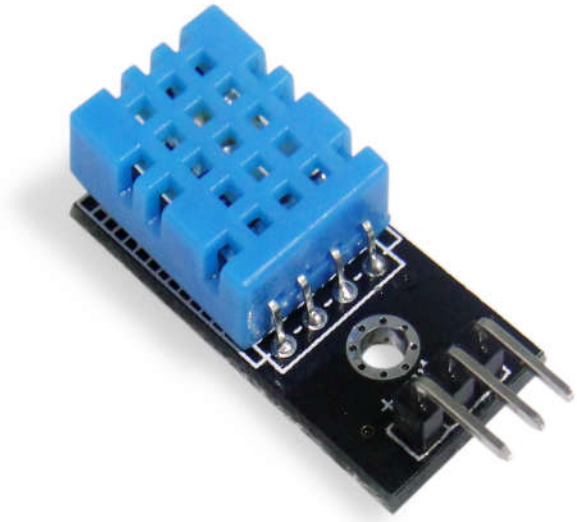
DHT11 Breakoutboard

Technical Manual Rev 1r0



DHT11 Temperature & Humidity sensor breakoutboard are made of two parts, a capacitive humidity sensor and a thermistor. It has very basic chip inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity.Compatible in gizDuino boards.

Features:

- 2.5mA max current use during conversion (while requesting data)
- Indicator LED to check if the board is powered up
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50 degC temperature readings +/- 2 degC accuracy
- Includes Jumperwires, 20cmF/F

General Specifications:

Input Supply Voltage: 3.5V ~ 5VDC
Output signal: Digital
Sampling rate: 1Hz(once every second)
PCB Dimensions: 12mm x 28mm

©2016 by e-Gizmo Mechatronix Central

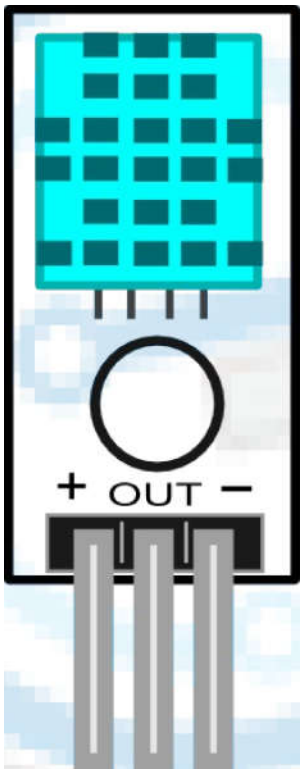


Figure 1. Major parts presentation of DHT11 Temperature and Humidity Sensor

©2016 by e-Gizmo Mechatronix Central

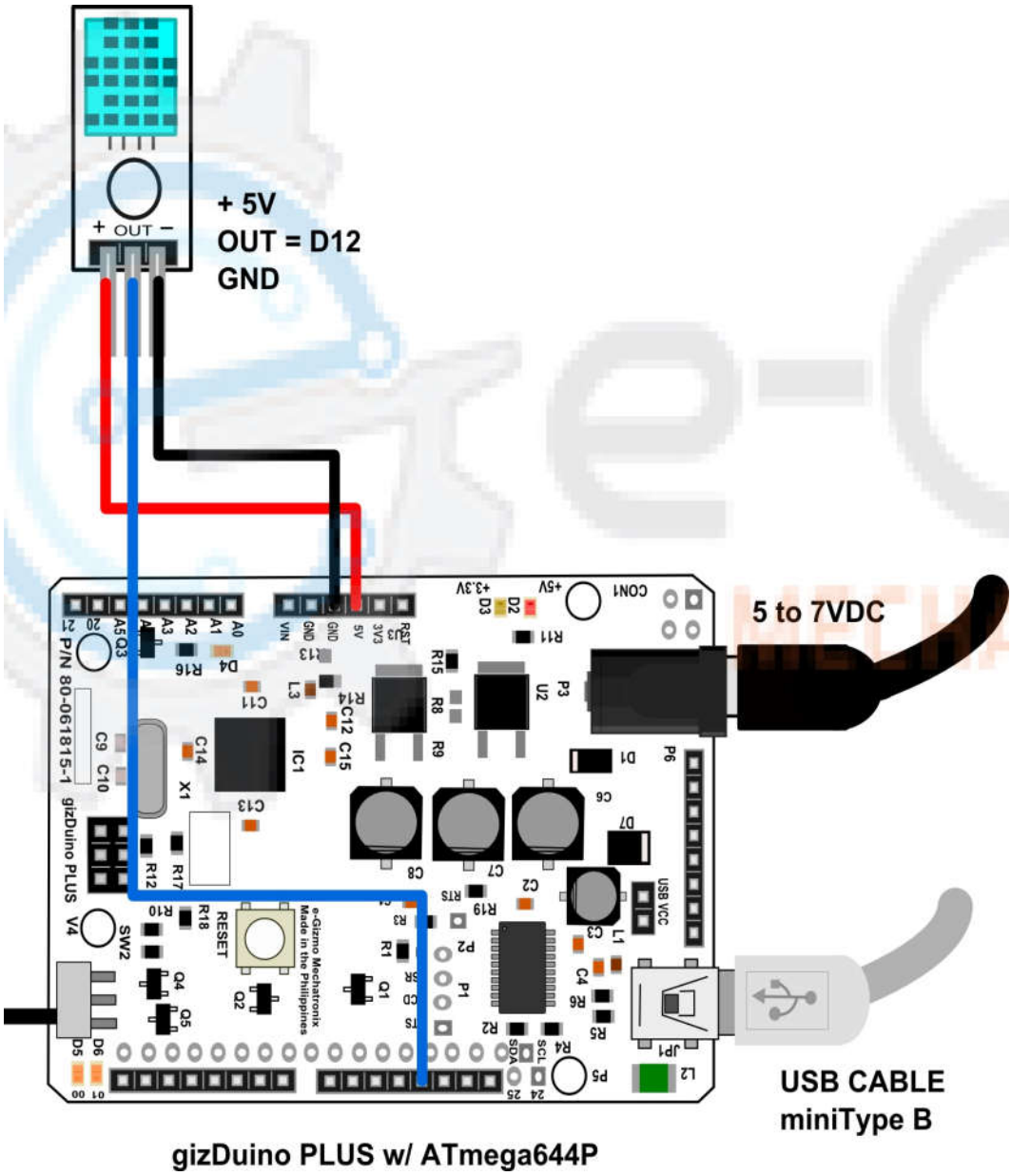


Figure 2. Sample connections

©2016 by e-Gizmo Mechatronix Central

Upload this code to the gizDuino PLUS Microcontroller. then Open the Serial Monitor to see the temperature and humidity data.

```
//*********************************************//
//          DHT 11 Breakoutboard               //
//                                             //
//      This sample sketch is reading the      //
// digital output signal of temperature and    //
// humidity value from DHT 11 module.          //
// Displaying data on the serial monitor.      //
//                                             //
//               Codes by:                     //
//       e-Gizmo Mechatronix Central           //
//        Taft, Manila, Philippines            //
//          http://www.egizmo.com              //
//           October 11,2016                   //
//*********************************************//

#include "DHT.h"

#define DHTPIN 12     // what pin we're connected to

// Uncomment whatever type you're using!
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302)
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);
// NOTE: For working with a faster chip, like an Arduino Due or Teensy, you
// might need to increase the threshold for cycle counts considered a 1 or 0.
// You can do this by passing a 3rd parameter for this threshold.  It's a bit
// of fiddling to find the right value, but in general the faster the CPU the
// higher the value.  The default for a 16mhz AVR is a value of 6.  For an
// Arduino Due that runs at 84mhz a value of 30 works.
// Example to initialize DHT sensor for Arduino Due:
//DHT dht(DHTPIN, DHTTYPE, 30);
```


©2016 by e-Gizmo Mechatronix Central

```
void setup() {
  Serial.begin(9600);
  Serial.println("DHTXX TEST!");

  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
   // Sensor readings may also be up to 2 seconds 'old' (its a very
slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("FAILED TO READ FROM DHT SENSOR!");
    return;
  }

  Serial.print("HUMIDITY: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("TEMPERATURE: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.println(" *F\t");

}
```


©2016 by e-Gizmo Mechatronix Central

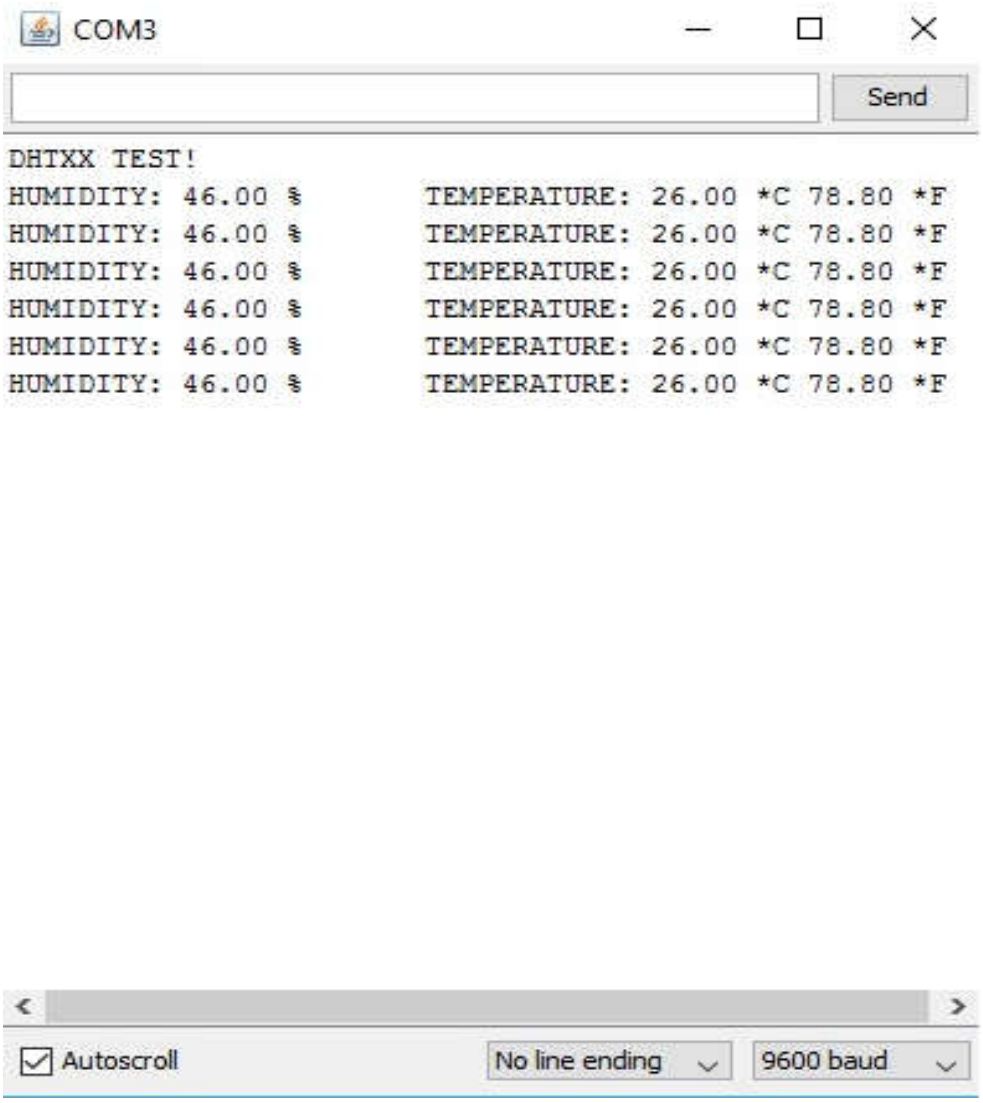


Figure 3. Serial Monitor

©2016 by e-Gizmo Mechatronix Central