Capacitive Humidity Sensors
DHTxx family Humidity Sensors

(Models DHT11/DHT21/DHT21 and also known as AM2303)

Figure 1. Typical DHT humidity sensor. [1]

Introduction
The DHT family of humidity sensors consist of a capacitive humidity sensor and a thermistor. It is available in a few package types, most commonly in a 15mm x 25mm plastic box with four leads. The different models work with the same principles, but they have slightly different specifications and accuracy.

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1” spacing

**Relative Humidity**

‘Relative humidity is the ratio of the current absolute humidity to the highest possible absolute humidity (which depends on the current air temperature). A reading of 100 percent relative humidity means that the air is totally saturated with water vapor and cannot hold any more, creating the possibility of rain. This doesn't mean that the relative humidity must be 100 percent in order for it to rain -- it must be 100 percent where the clouds are forming, but the relative humidity near the ground could be much less.' [2]

‘The relative humidity () of an air-water mixture is defined as the ratio of the partial pressure of water vapor (H₂O) (e) in the mixture to the equilibrium vapor pressure of water (e*) at a given temperature. Relative humidity is normally expressed as a percentage and is calculated by using the following equation:

\[
\text{RH} = \frac{e}{e^*}
\]

The higher the percentage, the more humid. The higher the relative humidity, the less drying takes place. This affects the human bodies’ perception of heat, as explained next.’ [3]
Figure 2. [7]
Capacitive Humidity Sensor and Physical principles models

A capacitive humidity sensor is used to measure relative humidity with a usual range of 5 to 95 % relHum. The sensor is a capacitor, made from two metal electrodes with a porous dielectric substance between them. Water vapour is able to penetrate this layer, changing the total capacitance. Typical capacitance at 50% relHum is 100 to 500pF, with 0.2 to 0.5pF change per 1% relHum.

As the sensor uses water vapour in the air, placing the sensor in the path of moving air allows it to respond faster. Relative humidity is directly dependant on the temperature and the sensor has a thermistor used for calibration.

<table>
<thead>
<tr>
<th></th>
<th>AM2302 / DHT22</th>
<th>DHT11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0-100%</td>
<td>20-90%</td>
</tr>
<tr>
<td>Absolute accuracy</td>
<td>±2%</td>
<td>±5%</td>
</tr>
<tr>
<td>Repeatability</td>
<td>±1%</td>
<td>±1%</td>
</tr>
<tr>
<td>Long term stability</td>
<td>±0.5% per year</td>
<td>±1% per year</td>
</tr>
<tr>
<td>Typical street price</td>
<td>US$ 4-10</td>
<td>US$ 1-5</td>
</tr>
</tbody>
</table>

### Raw usage

Most humidity sensors available are sold as units that include the capacitive humidity sensor, a temperature sensor and a microcontroller that reads, calculates and transmits this data.

If interface to a raw sensor is needed, the circuitry is identical to basic capacitance measurement. This is done by charging the sensor as a capacitor through a resistor until the voltage equals exactly 63.2% of the supply voltage, and make note of the time it took. Dividing the time it took with the charging series resistor value, (seconds / ohms), gets the capacitance.

This is then compared to the humidity to capacitance value or equation of the sensor, found in the datasheet.

The capacitor is then drained through a parallel resistor, and after making sure the capacitor is fully drained, the measuring can be repeated.

To actually get any usable humidity values, a relatively accurate temperature measurement is also needed, as relative humidity is very dependant on temperature.
Figure 4. Capacitive Humidity Sensor Schematic.  

Communication

DHT sensor family uses single wire serial communication.

You send a start pulse to the sensor, and it responds with the data using a 40-bit packet that includes the humidity and temperature. The polling rate is very slow at 0.5Hz (once every 2 seconds)

Data bits:

00-07  HB Relative Humidity
08-15  LB Relative Humidity
16-23  HB Temperature
24-31  LB Temperature
32-39  CHECKSUM
Checksum is calculated as Relative Humidity HB + Relative Humidity LB + Temp HB + Temp LB.

Example data [00000111 00000101 00000010 00000011 00010001]  
RH 75, Temp 23, CHK 17

Polling the sensor:
Send the start pulse:
  Keep data line high for at least 18ms
Set the line low for at least 500µs, then pull the line high for 20 to 40µs.
Response:
The sensor pulls the line low for 80µs, then high for 80µs.
Transmission:
  Sensor proceeds to send the data, each bit starts with 50µs low, then the bit data itself; 26 to 28µs for ‘0’, 70µs for ‘1’

![Timing Diagram](image)

Figure 5. Timing Diagram. [5]

**Applications: Arduino + Humidity Sensor + Temperature Sensor**

Sensor details: [https://www.youtube.com/watch?v=C7uS1OJccKI](https://www.youtube.com/watch?v=C7uS1OJccKI)
Setting up sensors video: https://www.youtube.com/watch?v=Qj69qfneNzI

Setting up sensors video2: https://www.youtube.com/watch?v=Rppuxjc_Rzc


Arduino code: http://playground.arduino.cc/Main/DHTLib

**DHT11 power consumption**

<table>
<thead>
<tr>
<th>VDD=5V, T = 25°C (unless otherwise stated)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conditions</strong></td>
</tr>
<tr>
<td>---------------------------------</td>
</tr>
<tr>
<td><strong>Power Supply</strong></td>
</tr>
<tr>
<td><strong>Current Supply</strong></td>
</tr>
<tr>
<td><strong>Average</strong></td>
</tr>
<tr>
<td><strong>Standby</strong></td>
</tr>
<tr>
<td><strong>Sampling period</strong></td>
</tr>
</tbody>
</table>

Figure 6. Power consumption sheet.

Humidity Sensor DHT11 component can be used with batteries. For example AA battery 400-900 mAh.

Battery Life = Battery Capacity in Milli amps per hour / Load Current in Milli amps * 0.7

Battery Life = 750~ mAh / 0.2 mA * 0.7 = 2600~ Estimated Hours.

Arduino requires about 46.5 mA current in order to operate properly and sleeping power consumption 34.4 mA.

**DHT11 material (used in component)**

‘This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a highperformance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.’ [8]
So basically humidity sensor is made out of regular material such as silicon microchip, ceramic body and plastic casing?

**Summary**

‘Humidity sensors are sensors that convert the moisture content in air, gases, bulk materials or soil into an electric output signal.

The demand for humidity sensors with higher accuracies, higher repeatability and long-term stability is growing. Wireless humidity sensors are on the rise for monitoring humidity in greenhouses, industrial applications and even in the residential sector.

Major amounts of humidity sensors are required by the various process industries such as the food industry, the mining sector, the cement industry, the chemical industry, the pharmaceutical industry and many more. They are used for product quality control, in-process quality testing and for research and development. In the food industry, humidity sensors help to guarantee the quality level of flour and potato chips. Moisture sensors are also of importance in the pharmaceutical industry; here they help to avoid excessive humidity, which could alter the quality features of the final products.’ [6]

**Reference**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 1 | Figure 1. Picture -
https://learn.adafruit.com/system/assets/assets/000/000/576/medium800/weather_dhtsensors.jpg?1396764183 |
| 2 | http://science.howstuffworks.com/dictionary/meteorological-terms/question651.htm |
| 5 | Timing Diagram -
| 7 | Datasheet -
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>