README

NAMES:

N. Hruthik Nitchal Rao	SE20UARI108
SIDDARTHA RAHUL K	SE20UARI084
LAVANYA DEEPAK GUNDA	SE20UARI087

DHT22 Sensor(also known as the AM2302)

- The DHT22 Digital Temperature and Humidity Sensor Module AM2302 is a basic, low-cost digital temperature and humidity sensor.
- It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin.
- It requires careful timing to grab data.
- you can only get new data from it once every 2 seconds, so when using A library, sensor readings can be up to 2 seconds old.
- It uses a polymer capacitor to sense the temperature and humidity, measuring the temperature of the air between –40 and 80 degrees Centigrade (which Arduino can convert to Fahrenheit), and the relative humidity between 0 and 100%.
- the leftmost pin is for voltage to power the sensor (anywhere from 3.3 to 6 volts; we'll use the 3.3 volt pin on Arduino); the second pin outputs data from the sensor to the Arduino; the third pin is null (not connected to anything); and the rightmost pin is GND.
- relatively inexpensive and easy to use for Home Automation.
- The capacitive humidity sensor works by detecting the change in capacitance between two electrodes as the humidity of the air changes.
- The thermistor is a temperature-sensitive resistor that changes its resistance as the temperature changes.
- It has a built-in microcontroller that converts the analogue signals from the capacitive humidity sensor and the thermistor into digital signals.
- The digital signals are then transmitted to a host device, such as a microcontroller or a computer, over a single data line.
- To use the DHT22, you will need to connect it to a microcontroller or a computer. It has four pins: VCC, GND, DATA, and NC. The VCC pin should be connected to a 3.3V or 5V power supply. The GND pin should be connected to ground. The DATA pin

is the data line that should be connected to a digital input pin on the microcontroller or computer. The NC pin is not connected.

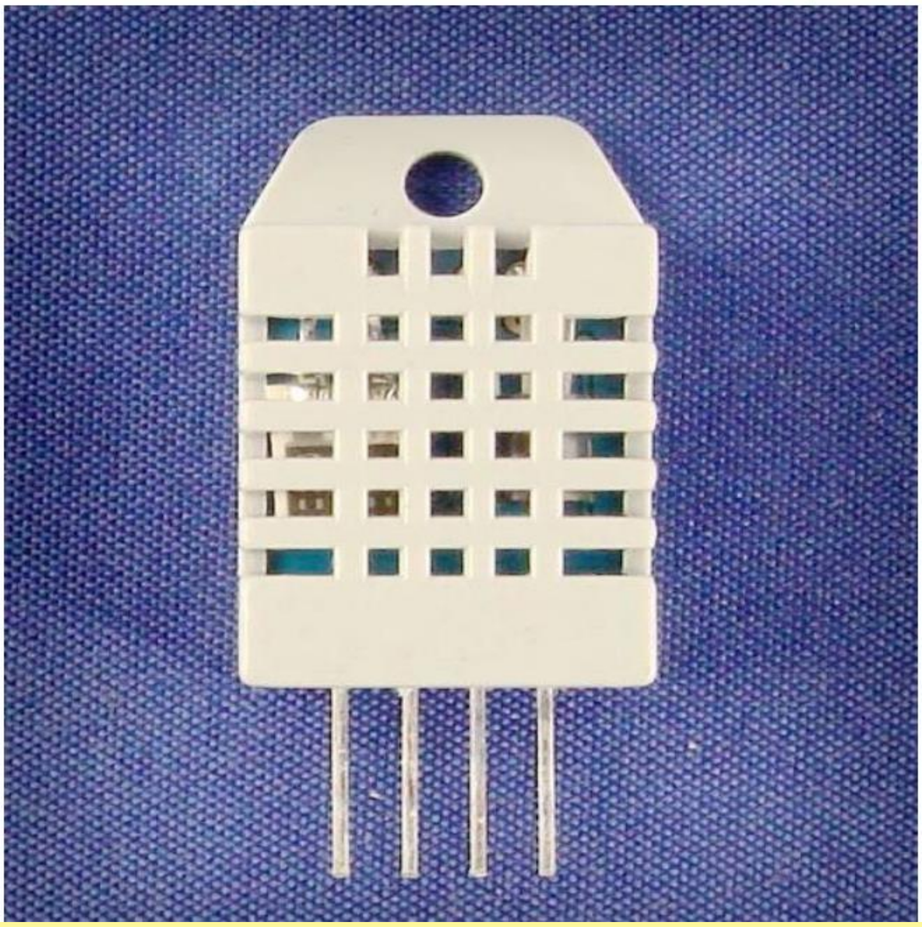
- Once the DHT22 is connected, you can use a software library to read the temperature and humidity data. There are software libraries available for a variety of programming languages, including Arduino, Python, and C++.

Technical Details

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")
- 4 pins, 0.1" spacing

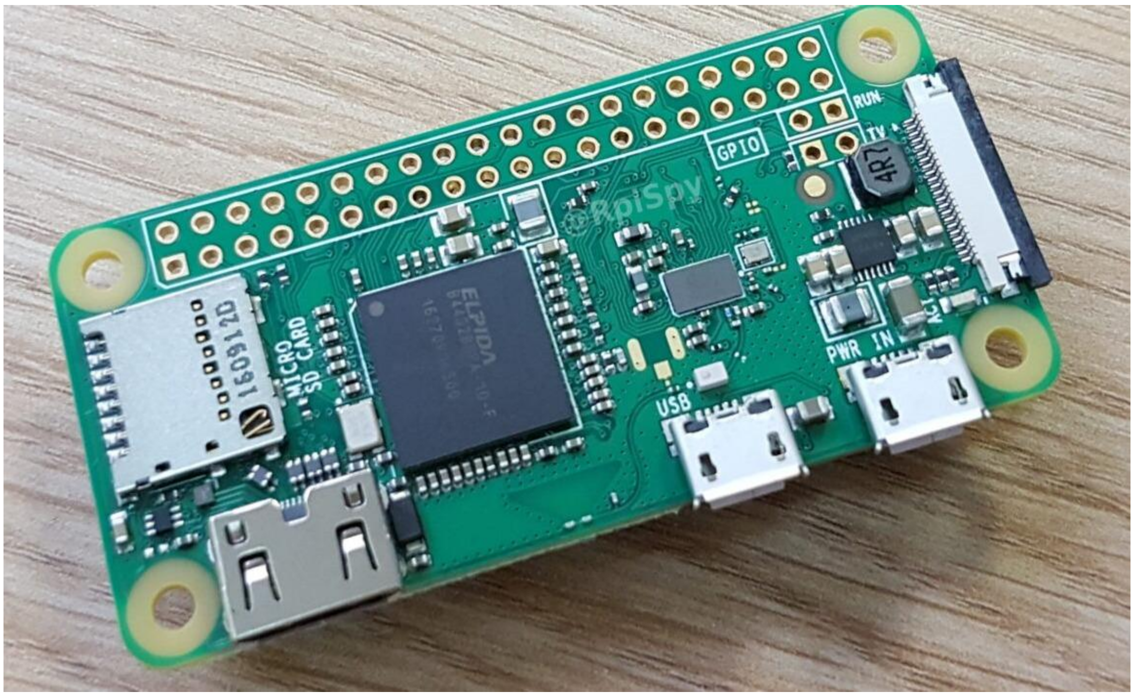
Digital-output relative humidity & temperature sensor/module

DHT22 (DHT22 also named as AM2302)



Raspberry Pi:

- The Raspberry Pi Zero W is a small, affordable, and versatile single-board computer that is part of the Raspberry Pi family.
- The "W" in its name stands for "Wireless" because it includes built-in wireless connectivity.
- It is powered by a 1GHz single-core CPU and has 512MB of RAM.
- It runs a version of the Linux operating system called Raspbian. Raspbian comes with a variety of pre-installed software, a web browser, a media player, and a programming environment.
- To use the Raspberry Pi Zero W, you will need a micro HDMI cable, a micro USB power supply, and a microSD card along with a power source, a display, and input devices such as a keyboard and mouse.
- Its small size and low power consumption make it ideal for portable and battery-powered projects.



Specifications:

- 802.11 b/g/n wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)

- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI® port and micro USB On-The-Go (OTG) port
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- CSI camera connector

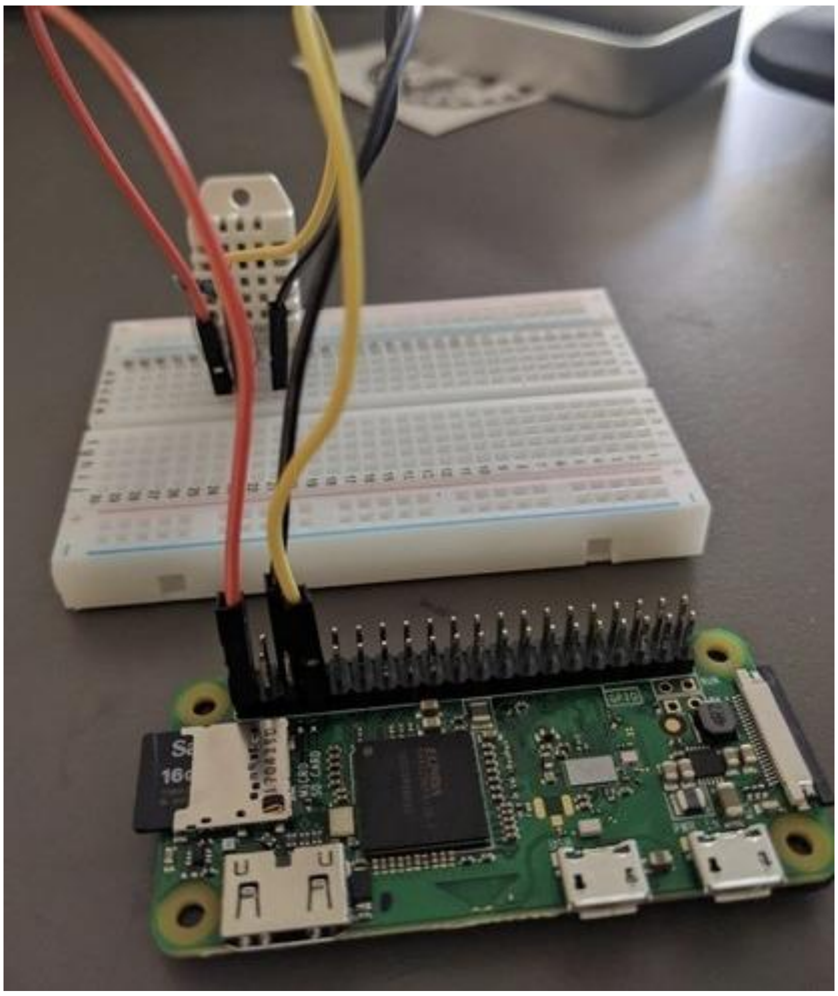
Working Code:

```
import Adafruit_DHT

sensor = Adafruit_DHT.DHT22

pin = 4

try:
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

  if humidity is not None and temperature is not None:
    print(f'Temperature: {temperature:.2f}°C')
    print(f'Humidity: {humidity:.2f}%')
  else:
    print('Failed to retrieve data from the DHT22 sensor.')

except KeyboardInterrupt:
     print('Measurement stopped by user')

except Exception as e:
     print(f'Error: {e}')

finally:
    GPIO.cleanup()
```

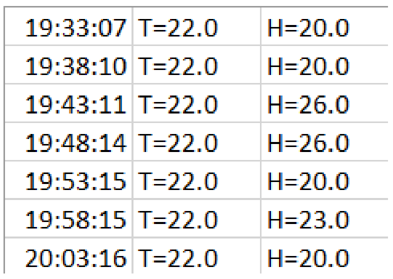
How the code works:

- First , we import the Adafruit DHT library, which provides functions to interact with
  the DHT22 sensor.

- Second, the sensor is set to Adafruit_DHT.DHT22, specifying that we are using a DHT22 sensor.
- pin is set to the GPIO pin number of whose we have connected the sensor.
- Inside the try-except block, data is read from the DHT22 sensor using Adafruit_DHT.read_retry(sensor, pin).
- If the data retrieval is successful (humidity and temperature values are not None), it prints the temperature and humidity values with two decimal places.
- If there's an error during data retrieval, it prints an error message. This helps us identify any issues with the sensor or the GPIO pin.
- this exception handler is used to catch a keyboard interrupt if the user wants to stop the measurements manually. It prints a message indicating that the measurement was stopped.
- Finally, in the last step, cleaning GPIO Resources ensures that GPIO pins are properly released, which is essential for its smooth functioning.



Results:



19:33:07	T=22.0	H=20.0
19:38:10	T=22.0	H=20.0
19:43:11	T=22.0	H=26.0
19:48:14	T=22.0	H=26.0
19:53:15	T=22.0	H=20.0
19:58:15	T=22.0	H=23.0
20:03:16	T=22.0	H=20.0