# DHT Sensor Documentation - Raspberry Pi 5 Integration

El Mehdi Adnani Kadmiri

July 17, 2025

## 1 Description

The DHT sensor (e.g., DHT11 or DHT22) is used for measuring temperature and humidity. It includes a thermistor and a capacitive humidity sensor to generate calibrated digital output.

## 2 Applications

- Weather stations
- Home automation
- Greenhouse monitoring
- IoT environmental sensing

## 3 Working Principle

The DHT sensor measures temperature and humidity and sends the data as a digital signal via a single data pin. The sensor requires specific timing to read the data signal, which is parsed into two values: temperature and humidity.

## 4 Wiring Diagram

| DHT Pin | Raspberry Pi Pin | Function |
|---|---|---|
| VCC | 3.3V or 5V | Power |
| GND | GND | Ground |
| DATA | GPIO4 | Data Signal |

## 5 Libraries Used

### Python: Adafruit_DHT

This library simplifies interaction with DHT sensors.

- Install via pip: `pip install Adafruit-DHT`
- Usage: `humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)`

### C: WiringPi and Custom Protocol

There is no official WiringPi support; bit-banging and timing logic are used in custom C implementations.

- GPIO access: `wiringPiSetupGpio()`
- Signal parsing: Requires microsecond timing

## 6 Python Code Example

```
import Adafruit_DHT

sensor = Adafruit_DHT.DHT11
pin = 4

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

if humidity is not None and temperature is not None:
print(f"Temp={temperature:.1f}C  Humidity={humidity:.1f}%")
else:
print("Sensor failure. Check wiring.")
```

## 7 C Code Example

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

#define MAX_TIMINGS 85
#define DHT_PIN 4

int data[5] = {0,0,0,0,0};

void read_dht_data() {
        uint8_t laststate = HIGH;
        uint8_t counter = 0;
        uint8_t j = 0, i;

        data[0] = data[1] = data[2] = data[3] = data[4] = 0;

        pinMode(DHT_PIN, OUTPUT);
        digitalWrite(DHT_PIN, LOW);
        delay(18);
        digitalWrite(DHT_PIN, HIGH);
        delayMicroseconds(40);
        pinMode(DHT_PIN, INPUT);

        for (i = 0; i < MAX_TIMINGS; i++) {
                counter = 0;
                while (digitalRead(DHT_PIN) == laststate) {
                        counter++;
                        delayMicroseconds(1);
                        if (counter == 255)
                        break;
                }
                laststate = digitalRead(DHT_PIN);
                if (counter == 255) break;
                if ((i >= 4) && (i % 2 == 0)) {
                        data[j / 8] <<= 1;
```

```
                        if (counter > 16) data[j / 8] |= 1;
                        j++;
                }
        }

        if ((j >= 40) && (data[4] == ((data[0] + data[1] + data[2] + data[3
                float humidity = data[0];
                float temperature = data[2];
                printf("Humidity = %.1f %% Temperature = %.1f *C\n", humid
        } else {
                printf("Data not good, skip\n");
        }
}

int main(void) {
        if (wiringPiSetupGpio() == -1)
        return 1;
        while (1) {
                read_dht_data();
                delay(2000);
        }
        return 0;
}
```

# 8 Conclusion

The DHT sensor offers a compact and easy-to-use solution for measuring environmental temperature and humidity, making it perfect for Raspberry Pi-based monitoring systems.