The **DHT11** module is a temperature and humidity sensor that is widely used in embedded systems and IoT projects due to its simplicity and affordability. It communicates with microcontrollers like Arduino using a proprietary 1-wire protocol to transmit **temperature and humidity data** in an 8-bit format.

Key Features of DHT11:

1. **Measurement Range:**
    - **Humidity:** 20% to 90% Relative Humidity (RH)
    - **Temperature:** 0°C to 50°C
2. **Accuracy:**
    - **Humidity:** ±5%
    - **Temperature:** ±2°C
3. **Power Supply:** 3.3V to 5.5V
4. **Communication:** Single-wire digital signal
5. **Output Data:** 40 bits (5 bytes)

How DHT11 Works:

1. **Data Transmission:** The sensor uses a single data pin to send 40 bits of data, split into five 8-bit segments:
    - **1st Byte:** Integral part of the humidity value.
    - **2nd Byte:** Decimal part of the humidity value (always 0 for DHT11).
    - **3rd Byte:** Integral part of the temperature value.
    - **4th Byte:** Decimal part of the temperature value (always 0 for DHT11).
    - **5th Byte:** Checksum for error detection.
2. **Timing Protocol:**
    - **Start Signal:** The microcontroller sends a low pulse (at least 18 ms) to the sensor to initiate communication.
    - **Response Signal:** The sensor responds with a low pulse (~80 µs) followed by a high pulse (~80 µs).
    - **Data Transmission:** The sensor sends 40 bits of data (5 bytes) sequentially.
3. **Data Representation:**
    - Each bit is transmitted by varying the duration of the high pulse:
        - **0-bit:** High pulse lasts for ~26–28 µs.
        - **1-bit:** High pulse lasts for ~70 µs.
    - A low pulse (~50 µs) separates each bit.
4. **Checksum:**
    - The checksum byte ensures data integrity.
    - It is calculated as the sum of the first 4 bytes (humidity and temperature) modulo 256.

How DHT11 Uses 8-bit Segments for 40-bit Data

The data is transmitted as a **40-bit sequence** where:

- The first **8 bits** are for the integer part of the humidity.
- The next **8 bits** are for the decimal part of the humidity (always zero in DHT11).
- The third **8 bits** are for the integer part of the temperature.
- The fourth **8 bits** are for the decimal part of the temperature (always zero in DHT11).
- The last **8 bits** are the checksum.

For example:

- If the humidity is **55%** and the temperature is **23°C**, the transmitted data would look like: scss

```
00110111 00000000 00010111 00000000 00110001

(55)     (0)      (23)     (0)      (49 checksum)
```

Advantages of DHT11:

1. Simple to interface with microcontrollers.
2. Requires minimal external components.
3. Provides both temperature and humidity data in a compact package.

Limitations:

1. Limited accuracy compared to more advanced sensors like DHT22.
2. Slower response time (~1 reading per second).
3. Narrower range of measurement (e.g., temperature limited to 0–50°C).

Interface DHT11 Module With Arduino

How to Make an Arduino Weather Station With DHT11 Temperature and Humidity Sensor

How to Use a DHT11 Humidity Sensor on the Arduino - Ultimate Guide to the Arduino #38

DHT11 vs DHT22 with Arduino