

## Das DHT11-Modul: Funktionsweise und Programmierung

Das DHT11-Modul dient zur Messung von Temperatur- und Luftfeuchtigkeitswerten. Dabei können Temperaturen zwischen 0°C und 50°C und Luftfeuchtigkeitswerte zwischen 20% und 90% gemessen werden. Nach Datenblatt ist die Auflösung dabei 1°C bzw. 1%; die Genauigkeit liegt bei ca. 1°C bzw. 4%. Betrieben wird der Baustein mit einer Spannung zwischen 3,0 V und 5,5 V. Weitere Informationen - insbesondere zur Genauigkeit - findet man unter:

[www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf](http://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf).

Der eigentliche Sensor-Baustein ist häufig auf einer kleinen Platine (KY-015) montiert; auf dieser befindet sich meist eine Beschriftung für die Anschlüsse sowie ein Pullup-Widerstand für die Datenleitung (vgl. Abb. 1). Tatsächlich besitzt das Modul nur eine einzige Datenleitung; über diese werden die gemessenen Werte Bit für Bit übertragen. Dabei wird allerdings nicht das standardisierte 1-Wire-Protokoll benutzt. Das vom DHT11 eingesetzte Protokoll ist aber recht einfach und lässt sich mit wenig Aufwand in ein entsprechendes Mikrocontroller-Programm umsetzen.

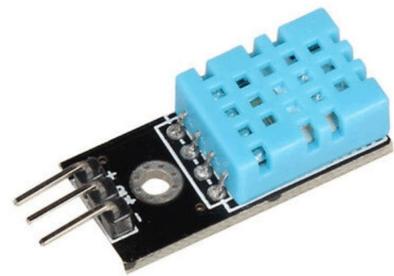


Abb. 1

### Funktionsweise

Im Leerlauf ist der Pegel der Datenleitung auf High. Um Messwerte abzufragen, zieht der Mikrocontroller die Datenleitung für mindestens 18 ms auf Low (Phase A, vgl. Abb. 2). Danach stellt der Mikrocontroller seinen Port vom Ausgabemodus in den Eingangsmodus um; von jetzt an überwacht der Mikrocontroller den Pegelzustand. Nun wird der Pegel vom oben erwähnten Pull-Up-Widerstand bzw. vom DHT11 wieder auf High gezogen. Gemäß Datenblatt zieht der DHT11 innerhalb von 20 - 40 us den Pegel wieder auf Low (Phase B); meine Messung ergab für diese Phase nur eine Dauer von 13 us. Es schließt sich eine Low-Phase mit ca. 80 us und eine High-Phase mit ca. 90 us an (Phasen C und D). Diese beiden Signale kann der Mikrocontroller als Quittierung seiner Anfrage deuten. Mit der Phase D ist die **Startsequenz** beendet.

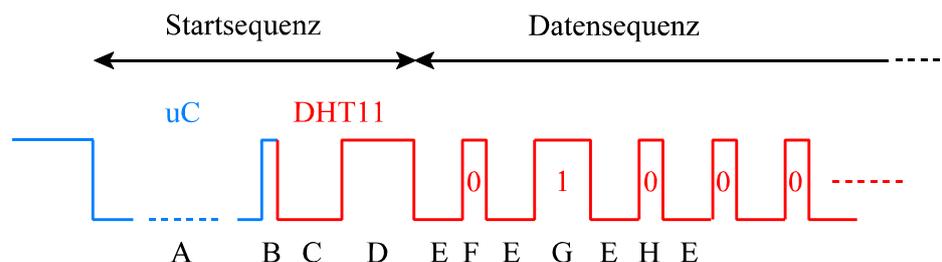


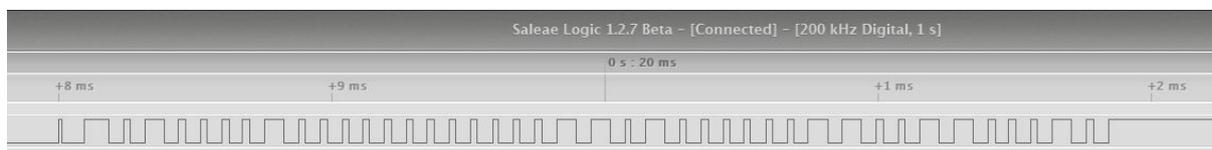
Abb. 2

Nun werden die Daten übertragen, insgesamt 5 Bytes: In dieser **Datensequenz** wird zunächst ein Byte für den ganzzahligen Teil der Luftfeuchtigkeit übertragen, danach ein Byte für die Stellen hinter dem Komma; dieses Byte ist für uns allerdings wegen der Messgenauigkeit irrelevant (s. o.). Es folgen zwei Bytes für die Temperatur, das erste für den ganzzahligen Teil, das zweite für die Nachkommastellen; auch hier ist das zweite Byte wieder irrelevant. Das letzte Byte beinhaltet eine **Prüfsumme**: Alle 5 Bytes werden vom DHT11 addiert; die rechten 8 Bits dieser Summe bilden die Prüfsumme. Diese kann ggf. benutzt werden, um Übertragungsfehler zu entdecken.

Bild 2 zeigt, wie die Datensequenz im Detail aussieht: Die einzelnen Bits eines Bytes werden mit dem höchstwertigen Bit beginnend der Reihe nach übertragen. Dabei wird jedes Bit durch eine Low- und eine anschließende High-Phase dargestellt. Die Low-Phasen (in Abb. 2 mit E gekennzeichnet) dauern immer ca. 50  $\mu$ s lang; die Highphasen sind bei einem Bit 0 ca. 24  $\mu$ s lang (Phasen F und H in Abb. 2) und bei einem Bit 1 ca. 70  $\mu$ s lang (Phase G in Abb. 2). An der Länge der High-Phasen kann der Mikrocontroller also feststellen, welches Bit gerade übertragen wird.

Insgesamt werden auf diese Weise alle  $5 \cdot 8 = 40$  Bits übertragen; an die High-Phase des allerletzten Bits wird noch einmal eine Low-Phase (E) angeschlossen. Danach geht der Baustein wieder in den Leerlauf (s. o.) über.

Für die Abb. 3 wurde eine solche Datenübertragung mit einem Logik-Analysator festgehalten. Von der Phase A der Startsequenz ist allerdings nur ein kleiner Teil zu sehen; mit ihren 18 ms ist sie um ein Vielfaches länger als die Datensequenz.



**Abb. 3:** Luftfeuchtigkeit: 66%; Temperatur: 26°C

## Programm für das Nano-Board

Es folgt ein BASCOM-Programm für den Atmega328p. Nach den ausführlichen Erörterungen aus dem letzten Abschnitt sollte es mit Hilfe der zahlreichen Kommentare gut nachzuvollziehen sein. Die LEDs an PortD.2 - PortD.5 wurden eingesetzt, um (in der Testphase) den Fortschritt von Start- und Datensequenz verfolgen zu können. Sie können bei Bedarf entfernt werden.

Wer möchte, kann das Programm auch durch einen Prüfsummentest ergänzen.

- ' Nano-Board
  - ' Temperatur- und Luftfeuchtigkeit messen
  - ' DHT11 an PortB.4
  - ' Ausgabe der Werte als Byte über die serielle Schnittstelle
  - ' LEDs an PortD.2-PortD.5; sie dienen lediglich zur Kontrolle
- ' Version 1.1, getestet am 03.06.2021



```
    Loop Until Pinb.4 = 1           'Warten bis Signalleitung wieder auf High
    Portd.3 = 1                    'Kontroll-LED (D.3) an
    Timer1 = 0                     'Zeitmesser auf 0; läuft permanent
    Waitus 5                       'Etwas Zeit für Flankenanstieg
    Do                             'Warten bis Signalleitung vom DHT11...
    Loop Until Pinb.4 = 0         '... auf Low gezogen wird
    If Timer1 > 120 Then Portd.4 = 1 'Highzeit > 60 us? Dann zur Kontrolle LED (D.4) an
    End If
End Sub

Sub Empfange_bits
  For I = 1 To 40
    Ddrb.4 = 0
    Do
      Loop Until Pinb.4 = 1       'Warten bis Signalleitung ansteigt
      Timer1 = 0                  'Zeitmesser auf 0; läuft permanent
      Do
        Loop Until Pinb.4 = 0     'Warten bis Signalleitung auf Low geht
        'Dauer des High-Zustands > 50 us -> Bitwert = 1, sonst 0...
        If Timer1 > 100 Then Bitarray(i) = 1 Else Bitarray(i) = 0
      Next I
      Portd.5 = 1                 'zur Kontrolle LED (D.5) an
    End Sub

Sub Feuchtigkeit_ausgabe
  For I = 1 To 8
    Nr = 8 - I
    Hum.nr = Bitarray(i)
  Next I
  Printbin Hum
End Sub

Sub Temperatur_ausgabe
  For I = 17 To 24
    Nr = 24 - I
    Temp.nr = Bitarray(i)
  Next I
  Printbin Temp
End Sub
```

## Der große Bruder DHT22

Der DHT22 hat nahezu dieselben Eigenschaften wie der DHT11. Allerdings liefert er genauere Messwerte: Die Auflösung ist 0,1 °C bzw. 0,1 %, die Genauigkeit liegt bei 0,5 °C bzw. 2 %.