

Projektbericht

Projekt : Smart-Garten mit Hilfe von Temperatur- und Feuchtigkeitsmessgerät

DHT11-Sensor

Deniz Özdemirli-190503029

Dilanur Gider-190503060

31.Mai 2022

Lehrkraft : Faruk Bağcı / Ferit Tiryaki / Ebru Subutay / Onur Akgün

- **Das zu lösende Problem :**

Die Temperatur ist eine wichtige Eigenschaft, die sich erheblich auf die biologischen Vorgänge im Boden auswirkt und die physikalischen und chemischen Prozesse steuert. Die Temperatur im Pflanzenanbaubereich sollte regelmäßig überwacht und bei Bedarf angepasst werden. Der ideale Temperaturwert für die Lichtperiode liegt zwischen 24-26°C. Für die Dunkelperiode liegt der ideale Temperaturwert zwischen 23-24°C. Die Temperatur in der Umgebung der Pflanze sollte nie unter 21°C fallen und nie über 28°C steigen. Die Temperatur in der Anbaufläche kann mit Hilfe von Ventilatoren, Gebläsen und Heizgeräten reguliert werden.

Die Bodenfeuchtigkeit sollte gemessen werden, um die Menge an Wasser zu bestimmen, die den Pflanzen im Boden zur Verfügung steht. Denn die Pflanzen können nicht das gesamte Wasser im Boden nutzen. Mit anderen Worten: Die Pflanzen können das Wasser zwischen der Feldkapazität und dem Verwelkungspunkt nutzen, zwei wichtigen Eigenschaften des Bodens. Untersuchungen zeigen, dass der für das Pflanzenwachstum am besten geeignete Wassergehalt im Boden zwischen 25 % und 75 % liegen sollte. In Anbetracht all dieser Faktoren sollte der Wasserstand im Boden bekannt sein, um zu wissen, ob es Zeit für eine Bewässerung ist oder nicht. Zu diesem Zweck sollten Feuchtigkeitskontrolle durchgeführt werden.

Daher ist die Verfügbarkeit von Daten wie Bodentemperatur und -feuchtigkeit unabdingbar, um die Beziehungen zwischen Pflanze und Boden zu verstehen und Aussagen über die Nutzung des Bodens treffen zu können.

Da Bodentemperatur und -feuchtigkeit die wichtigsten Entwicklungsfaktoren für Pflanzen sind, sollen mit dieser Anwendungsstudie folgende Fragen zum Nutzen des Anwenders beantwortet werden, indem eine Temperatur- und Feuchtigkeitskontrolle des Bodens ermöglicht wird:

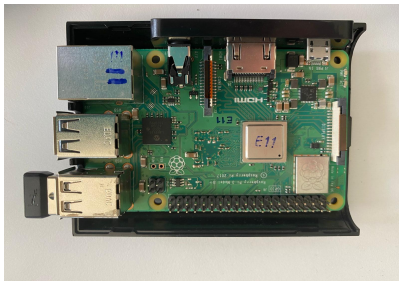
-Ist der Feuchtigkeitsgehalt des Bodens ideal?

- Soll ich den Boden gießen?
- Ist die Temperatur des Bodens ideal?
- Sollte ich eine Heizung verwenden, um die Temperatur des Bodens zu erhöhen?
- Sollte ich einen Ventilator verwenden, um die Bodentemperatur zu senken?

Zusammenfassend lässt sich sagen, dass die Smart Garden-Anwendung darauf abzielt, die aktive Temperatur und den Feuchtigkeitswert Ihres Gartens zu messen und eine E-Mail an den Verbraucher zu senden. Wenn die Feuchtigkeit des Bodens im Garten unter den idealen Feuchtigkeitswert fällt, wird eine E-Mail an den Verbraucher gesendet, die Sie auffordert, den Boden zu bewässern oder zu erwärmen oder zu kühlen, wenn die Temperatur des Bodens einen anderen Wert als den idealen Temperaturbereich annimmt.

- **Literatur Suche**

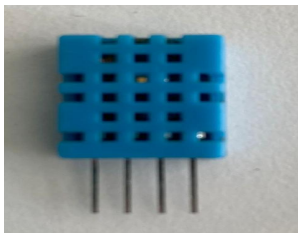
Was ist Raspberry Pi?



Der Raspberry Pi ist ein Open-Source-Einplatinencomputer in der Größe einer Kreditkarte. Pi wurde ursprünglich für Kinder entwickelt, um die Arbeitslogik von Computern zu verstehen und einfache Programmierkenntnisse zu erwerben. Raspberry Pi Monitor, Tastatur und Maus können wie ein normaler Computer angeschlossen und verwendet werden. Der von uns verwendete Raspberry Pi-Typ ist Raspberry Pi 3 Model B+, die aktuellste und leistungsfähigste Version der

Raspberry Pi-Typen. Es verfügt über integrierte Wi-Fi- und Bluetooth-Verbindungen. Es verfügt über einen ARMv8 64-Bit 4-Kern-Prozessor mit einer Frequenz von 1,2 GHz und 1 GB RAM-Speicher.

Spezifikationen des DHT11-Sensors:



DHT11 Temperatur- und Feuchtigkeitssensorplatine ist ein Modul mit einem DHT11-Sensor darauf, der getrennt und für Steckbretter oder andere Verwendungen einfacher gemacht wurde. Der Temperatur- und Feuchtigkeitssensor DHT11 ist eine fortschrittliche Sensoreinheit, die einen kalibrierten digitalen Signalausgang liefert. Es ist sehr zuverlässig und stabil im Langzeitbetrieb. Es enthält einen 8-Bit-Mikroprozessor, der eine schnelle und qualitativ hochwertige Antwort liefert. Es stellt die

Temperatur im Format Celsius dar. Das Gerät misst die Temperatur zwischen 0 und 50 °C mit einer Fehlerspanne von 2 °C und misst die Luftfeuchtigkeit mit einer Fehlerspanne von 5 % RH zwischen 20 und 90 % RH.

- Lösungsweg:

```

1  #include <wiringPi.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <stdint.h>
5  #include <string.h>
6  #define MAXTIMINGS 85
7  #define DHTPIN 7
8  int dht11_dat[5] = { 0, 0, 0, 0, 0 };
9
10 void read_dht11_dat(){
11
12     uint8_t laststate = HIGH;
13     uint8_t counter = 0;
14     uint8_t j = 0, i;
15     float f;
16     dht11_dat[0] = dht11_dat[1] = dht11_dat[2] = dht11_dat[3] = dht11_dat[4] = 0;
17     pinMode( DHTPIN, OUTPUT );
18     digitalWrite( DHTPIN, LOW );
19     delay( 18 );
20     digitalWrite( DHTPIN, HIGH );
21     delayMicroseconds( 40 );
22     pinMode( DHTPIN, INPUT );
23
24     for ( i = 0; i < MAXTIMINGS; i++ ){
25         counter = 0;
26         while ( digitalRead( DHTPIN ) == laststate ){
27             counter++;
28             delayMicroseconds( 1 );
29             if ( counter == 255 ){
30                 break;
31             }
32         }
33         laststate = digitalRead( DHTPIN );
34         if(counter==255){
35             break;
36         }
37         if ( ( i >= 4) && ( i % 2 == 0) ){
38             dht11_dat[j / 8] <= 1;
39             if ( counter > 50 )
40                 dht11_dat[j / 8] |= 1;
41             j++;
42         }
43     }

```

```

44
45     if ( ( j >= 40) && (dht11_dat[4] == ( (dht11_dat[0] + dht11_dat[1] + dht11_dat[2] + dht11_dat[3]) & 0xFF) ) ){
46
47         char num1[2];
48         num1[0] = (dht11_dat[2] / 10)+'0';
49         num1[1] = (dht11_dat[2] % 10)+'0';
50
51
52         char num[2];
53         num[0] = (dht11_dat[0] / 10)+'0';
54         num[1] = (dht11_dat[0] % 10)+'0';
55         printf("%s",num);
56
57         char command[500];
58         strcpy(command,"echo -e \"Subject: Test Mail\r\n\r\nDie Feuchtigkeit
59         ist: \"");
60         strncat(command,num,2);
61         strcat(command,"% .\nDie Temperatur: ");
62         strncat(command,num1,2);
63         strcat(command," C.\nDie Feuchtigkeit ist mehr als 50%.
64         Bitte lassen Sie Ihren Garten die Sonne sehen! \" |smtp
65         --debug --from=default -t bahceaklli@gmail.com");
66
67         char command1[500];
68         strcpy(command1,"echo -e \"Subject: Test Mail\r\n\r\nDie
69         Feuchtigkeit ist: \"");
70         strncat(command1,num,2);
71         strcat(command1,"% .\nDie Temperatur: ");
72         strncat(command1,num1,2);
73         strcat(command1," C.\nDie Feuchtigkeit ist weniger als
74         50%. Bitte bewaessern Sie Ihren Garten! \" |smtp --debug
75         --from=default -tbahceaklli@gmail.com");
76
77         char command2[500];
78         strcpy(command2,"echo -e \"Subject: Test Mail\r\n\r\nDie
79         Feuchtigkeit ist: \"");
80         strncat(command2,num,2);
81         strcat(command2,"% .\nDie Temperatur: ");
82         strncat(command2,num1,2);
83         strcat(command2," C.\nTemperatur ist ueber 28 Grad. Bitte
84         kuehlen Sie Ihren Garten! \" |smtp --debug --from=default
85         -t bahceaklli@gmail.com");
86
87         char command3[500];
88         strcpy(command3,"echo -e \"Subject: Test Mail\r\n\r\nDie
89         Feuchtigkeit ist: \"");
90         strncat(command3,num,2);

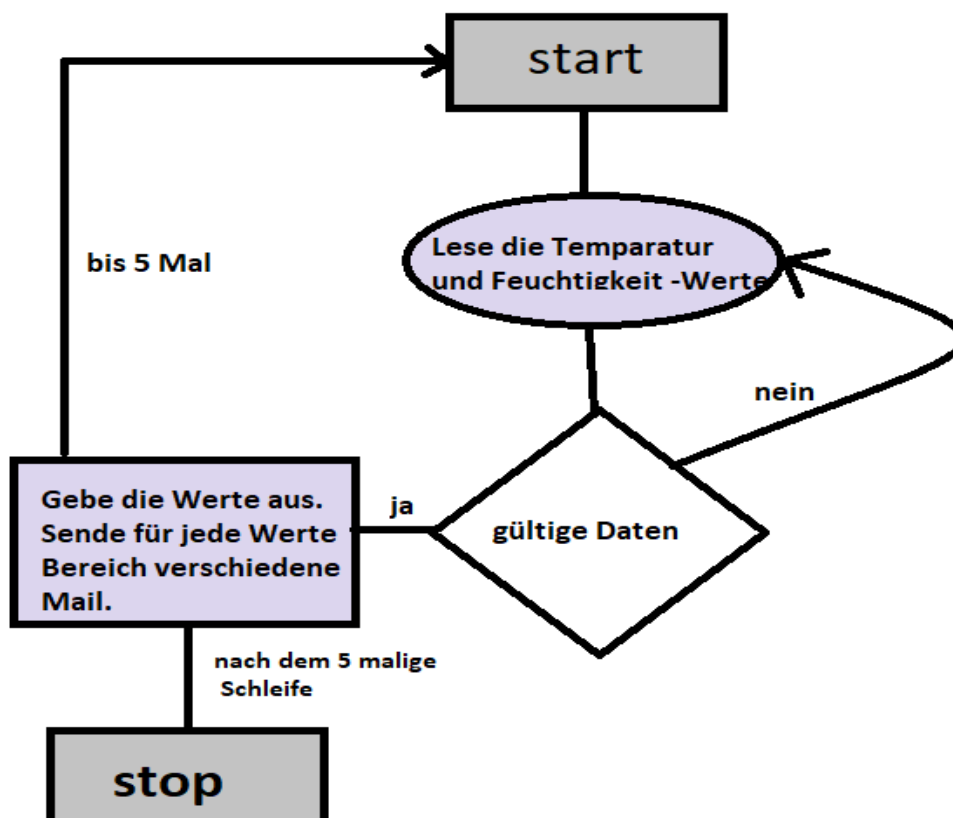
```

```

90     strcat(command3,num,2);
91     strcat(command3,"% .\nDie Temperatur: ");
92     strcat(command3,num1,2);
93     strcat(command3," C.\nTemperatur ist unter 21 Grad. Bitte
94     halten Sie Ihren Garten warm!\n" |msmtp --debug
95     --from=default -t bahceaklli@gmail.com");
96
97     char command4[500];
98     strcpy(command4,"echo -e \"Subject: Test Mail\r\n\r\nDie
99     Feuchtigkeit ist: ");
100    strcat(command4,num,2);
101    strcat(command4,"% .\nDie Temperatur: ");
102    strcat(command4,num1,2);
103    strcat(command4," C.\nDie Werte sind normal. \n" |msmtp
104    --debug --from=default -t bahceaklli@gmail.com");
105
106    f = dht11_dat[2] * 9. / 5. + 32;
107
108    printf( "Humidity = %d.%d %% Temperature = %d.%d C (%.1f
109    F)\n", dht11_dat[0], dht11_dat[1], dht11_dat[2],
110    dht11_dat[3], f );
111
112    if(dht11_dat[0]>75){ system(command);}
113    if(dht11_dat[0]<50){ system(command1);}
114    if(dht11_dat[2]>28) {system(command2);}
115    if(dht11_dat[2]<21) { system(command3);}
116    else{system(command4);}
117
118    }else {printf( "Data not good, skip\n" );}
119
120 }
121 int main( void ){
122
123     printf( "Raspberry Pi wiringPi DHT11 Temperature test program\n" );
124     int counter=0;
125     if ( wiringPiSetup() == -1 ) exit( 1 );
126     while ( 1 ){
127         read_dht11_dat();
128         delay( 2000 );
129         counter++;
130         if(counter==5) return 0;
131     }
132
133     return(0);
134 }
135

```

- Algorithm-Flowchart:



Main:

Geben Sie die Zeichenfolge "Raspberry Pi wiringPi DHT11 Temperature test programme\n" im Terminal aus.

Legen Sie eine neue Variable int counter an und setzen Sie sie auf 0, um zu speichern, wie oft die Messung durchgeführt wird.

Rufen Sie die Funktion WiringPiSetup auf und geben Sie an, dass Sie das eigene Nummerierungssystem des WiringPi verwenden möchten.

If Liegt ein Fehler in der Bedienung des wiringPi vor:

Beenden Sie das Programm.

While(1) :

Rufen Sie die Funktion auf, die den Temperatur- und Luftfeuchtigkeitswert ausliest, d. h.

read_dht11_dat().

Nach Beendigung der Funktion erhöhen Sie counter.

Wenn counter == 5 :

Beenden Sie das Programm.

Beenden Sie das Programm.

read_dht11_dat() :

Pin-Nummer definiert durch den Anschluss.

Das Array dht11_dat[5], in dem die Daten gespeichert werden, wird erstellt, und jedes Element hat den Wert 0.

Die Float-Variable f wurde erstellt, um den Fahrenheit-Wert zu speichern.

Der folgende Code besagt, dass DHTPIN ein Ausgangspin ist.

```
pinMode( DHTPIN, OUTPUT );
```

Im Raspberry PI wird der Code digitalWrite([pin], [HIGH/LOW]) verwendet, um eine digitale Ausgabe von einem Pin zu erhalten. Der folgende Code wurde so geschrieben, dass die Arbeit mit DHTPIN beendet wurde und das Signal von dort oder das Signal und der Strom zum Pin gestoppt wurden:

```
digitalWrite( DHTPIN, LOW );
```

Der folgende Code wurde geschrieben, um das digitale 1-Signal (ein oder vorhanden) von DHTPIN zu empfangen:

```
digitalWrite( DHTPIN, HIGH );
```

Pin bereit zum Lesen. Wenn INPUT auf diese Weise geschrieben wird, wird dem Raspberry mitgeteilt, dass DHTPIN ein Eingangspin ist.

```
pinMode( DHTPIN, INPUT );
```

Eine for-Schleife wurde geschrieben, um die Änderung zu erkennen und die Daten zu lesen, und der folgende Code wurde geschrieben, um zu sehen, ob in dieser Schleife ein Signal an DHTPIN anliegt, und um diese eingehende Signal zu speichern:

```
laststate = digitalRead( DHTPIN );
```

Bedingung i >= 4 hinzugefügt, damit die ersten 3 Übergänge ignoriert und nicht gelesen werden.

```
if ( ( i >= 4 ) && ( i % 2 == 0 ) ){
```

Dann wurde der folgende Code geschrieben, um jedes Bit den Speicherbytes zuzuordnen:

```
dht11_dat[j / 8] <<= 1;
```

Prüfen Sie, ob die Temperatur und die Luftfeuchtigkeit richtig gespeichert sind:

j > = : Prüfen Sie, ob wir 40 Bits (8 Bits * 5) gelesen haben.

dht11_dat[0] + dht11_dat[1] + dht11_dat[2] + dht11_dat[3]) & 0xFF : Prüfen Sie das letzte Byte mit der Prüfsumme.

```
if ( ( j >= 40 ) && ( dht11_dat[4] == ( ( dht11_dat[0] + dht11_dat[1] + dht11_dat[2] + dht11_dat[3] ) & 0xFF ) ) ) {
```

If die Daten korrekt sind und gelesen werden können:

Konvertieren Sie die Luftfeuchtigkeits- und Temperaturwerte in char, um sie als E-Mail zu versenden.

Dividieren Sie dht11_dat[0] durch 10 für die Luftfeuchtigkeit.

Da der Divisor ein int-Wert ist, fügen Sie den Wert '0' zu int hinzu, um ihn in char zu konvertieren.

Speichern Sie die char-Form des Divisors an Position 0 des soeben erstellten num-Arrays.

Da der Rest des Quotienten ein int-Wert ist, fügen Sie '0' zu int hinzu, um ihn in char zu konvertieren.

Speichern Sie die char-Version des Rests des Quotienten an Position 1 des soeben erstellten num-Arrays.

Führen Sie die gleichen Operationen für die Temperatur mit dht11_dat[2] und num1 durch.

Für jeden fünf Mailtyp wird ein neues char-Array(command[500]) erstellt, das in den Systembefehl geschrieben wird.

Diese Arrays haben alle den gleichen Inhalt, mit Ausnahme der Nachricht, die mit der Mail gesendet wird.

Jedem command-sarray wird mit der Funktion strcpy der erste Wert "Subject: Test Mail\n\nDie Feuchtigkeit ist:" zugewiesen.

Dann wird die Funktion strncat verwendet, um den in char umgewandelten Luftfeuchtigkeitswert in dem command-array hinzuzufügen.

Mit strcat zum Befehl, "% .\nDie Temperatur: " hinzugefügt.

Um den Temperaturwert in die E-Mail-Nachricht aufzunehmen, werden die Zeichen im Array num1 dem Befehl mit der Funktion strncat hinzugefügt.

Der String, der als letztes hinzugefügt wird, hängt vom Inhalt der zu versendenden E-Mail ab, zum Beispiel :

```
strcat(command1," C.\nDie Feuchtigkeit ist weniger als
50%. Bitte bewaessern Sie Ihren Garten! \n" |msmtp --debug
--from=default -tbahceaklli@gmail.com");
```

Statt der Nachricht "Die Feuchtigkeit ist weniger als 50%. Bitte bewaessern Sie Ihren Garten!" wird für jeden if-Befehl eine eigene Nachricht geschrieben. Der Rest des String-Teils ist in jedem Befehlsfeld gleich.

Konvertiert die Temperatur in Fahrenheit.

```
f = dht11_dat[2] * 9. / 5. + 32;
```

Drucken Sie die Temperatur- und Feuchtigkeitswerte auf dem Terminal aus.

If die Luftfeuchtigkeit >75 ist:

Führt den command in der Systemfunktion aus. Als Ergebnis sendet es eine E-Mail an den bahceaklli@gmail.com Account mit Informationen über Temperatur und Luftfeuchtigkeit und was der Benutzer tun sollte, um den Garten produktiver zu machen. "Die Feuchtigkeit ist mehr als 50%. Bitte lassen Sie Ihren Garten die Sonne sehen".

If die Luftfeuchtigkeit unter 50 liegt:

Führt den command1 in der Systemfunktion aus. Bei diesem if-Befehl wird die Meldung "Die Feuchtigkeit ist weniger als 50%. Bitte bewundern Sie Ihren Garten!".

If die Temperatur > 28 :

Führt den command2 in der Systemfunktion aus. In diesem if-Befehl wird die Meldung "Die Reparatur ist über 28 Grad. Bitte kühlen Sie Ihren Garten!" Nachricht.

If die Temperatur <21 :

Führt den command3 in der Systemfunktion aus. Dieser if-Befehl sendet die Meldung "Temperatur ist unter 21 Grad. Bitte halten Sie Ihren Garten warm!".

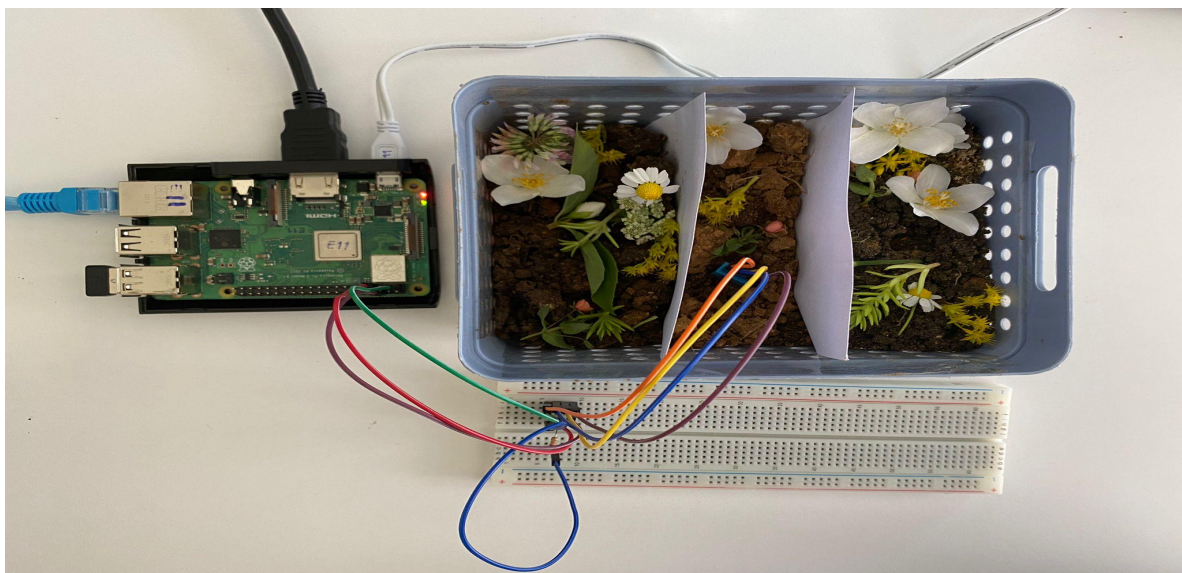
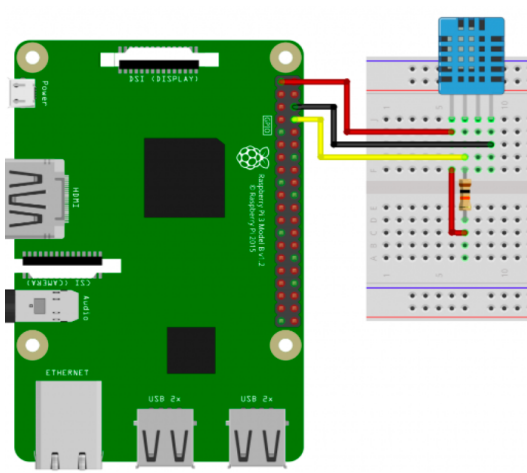
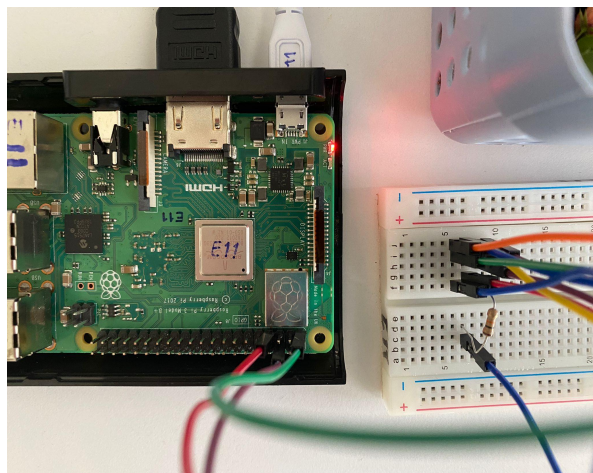
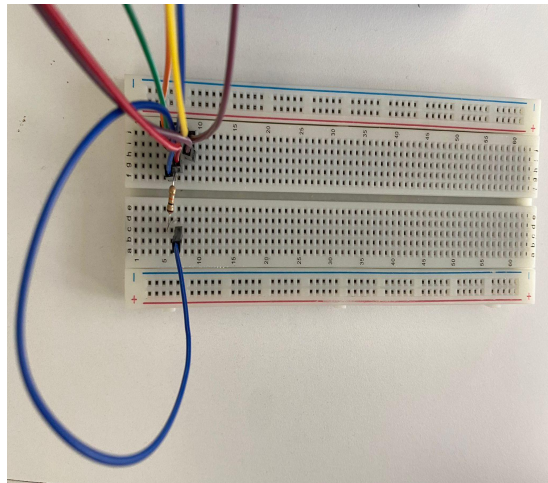
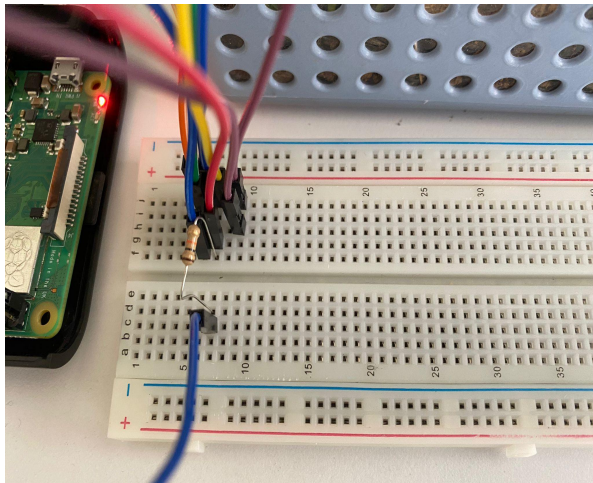
Else :

Führt den Befehl4 in der Systemfunktion aus, der die Meldung "Die Werte sind normal" sendet.

Else die Daten sind fehlerhaft und können nicht gelesen werden:

Die Zeichenkette "Data not good,skip" wird auf dem Terminal ausgegeben.

Beenden Sie das Programm.



- **Terminal-Ausgänge:**

```
File Edit Tabs Help
pi@raspberrypi:~ $ gcc -Wall -o t2 temp2.c -lwiringPi
^[[Api@raspberrypi:~/t2
Raspberry Pi wiringPi DHT11 Temperature test program
Data not good, skip
Data not good, skip
Humidity = 89.0 % Temperature = 26.5 C (78.8 F)
Data not good, skip
Humidity = 95.0 % Temperature = 27.5 C (80.6 F)
Humidity = 95.0 % Temperature = 28.1 C (82.4 F)
Data not good, skip
Humidity = 95.0 % Temperature = 28.7 C (82.4 F)
Humidity = 95.0 % Temperature = 28.8 C (82.4 F)
Data not good, skip
Data not good, skip
Data not good, skip
Data not good, skip
Data not good, skip
Humidity = 85.0 % Temperature = 28.8 C (82.4 F)
Humidity = 86.0 % Temperature = 28.9 C (82.4 F)
Humidity = 89.0 % Temperature = 29.2 C (84.2 F)
Humidity = 90.0 % Temperature = 29.4 C (84.2 F)
Humidity = 95.0 % Temperature = 29.6 C (84.2 F)
Data not good, skip
Humidity = 88.0 % Temperature = 29.6 C (84.2 F)
Humidity = 95.0 % Temperature = 29.8 C (84.2 F)
Humidity = 95.0 % Temperature = 30.1 C (86.0 F)
Humidity = 95.0 % Temperature = 30.2 C (86.0 F)
Humidity = 95.0 % Temperature = 30.6 C (86.0 F)
Data not good, skip
Humidity = 95.0 % Temperature = 30.7 C (86.0 F)
Humidity = 94.0 % Temperature = 30.8 C (86.0 F)
Humidity = 95.0 % Temperature = 31.0 C (87.8 F)
Data not good, skip
Data not good, skip
Humidity = 95.0 % Temperature = 32.5 C (89.6 F)
Humidity = 95.0 % Temperature = 31.5 C (87.8 F)
Humidity = 95.0 % Temperature = 31.3 C (87.8 F)
Humidity = 95.0 % Temperature = 31.2 C (87.8 F)
Humidity = 95.0 % Temperature = 31.2 C (87.8 F)
Humidity = 95.0 % Temperature = 31.1 C (87.8 F)
Humidity = 95.0 % Temperature = 31.0 C (87.8 F)
Humidity = 95.0 % Temperature = 31.3 C (87.8 F)
Humidity = 95.0 % Temperature = 31.8 C (87.8 F)
pi@raspberrypi:~ $
```

[illegible]

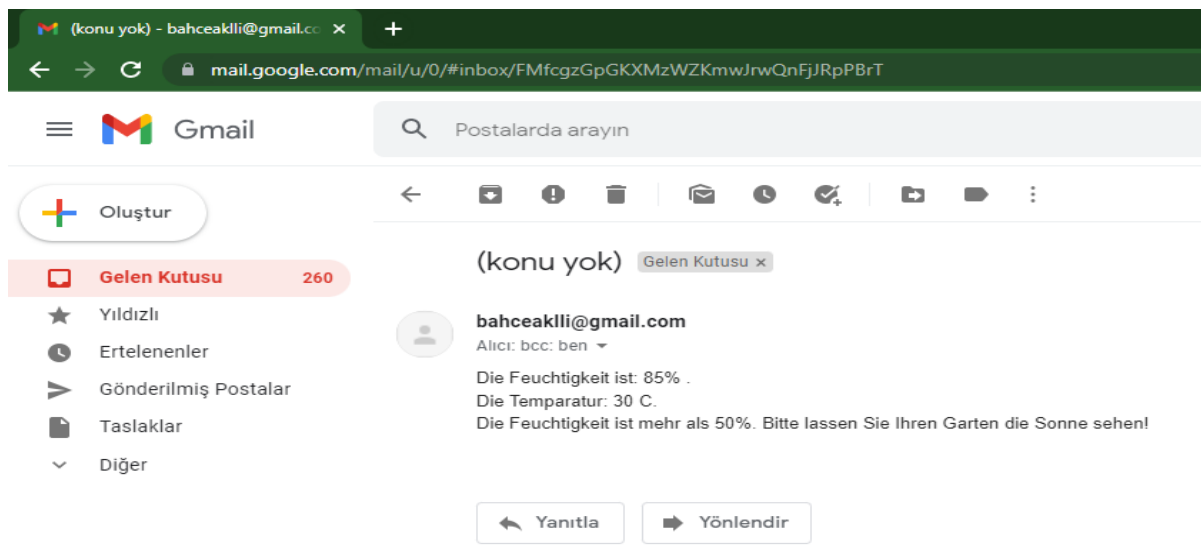
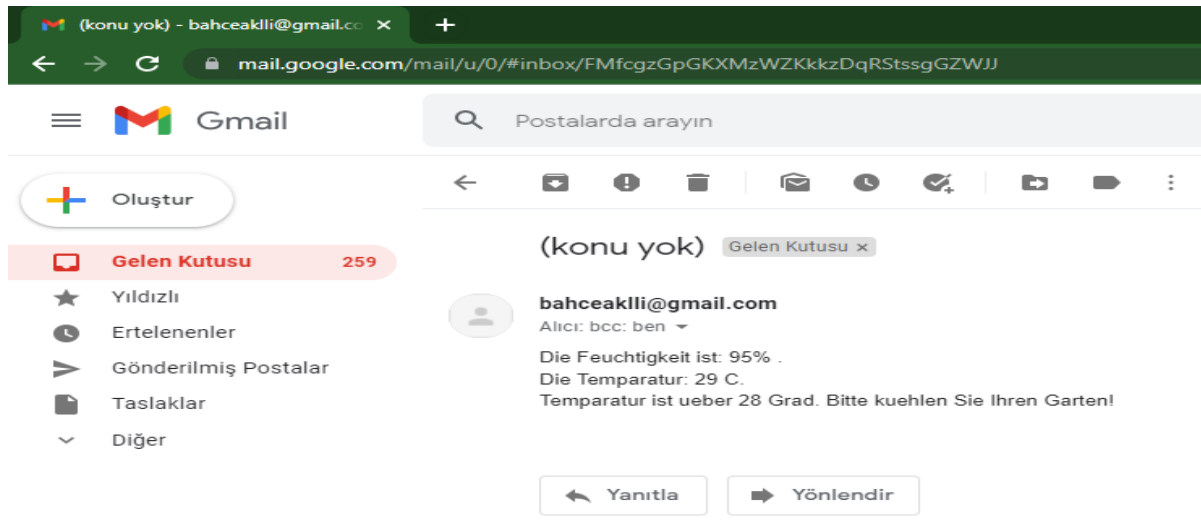
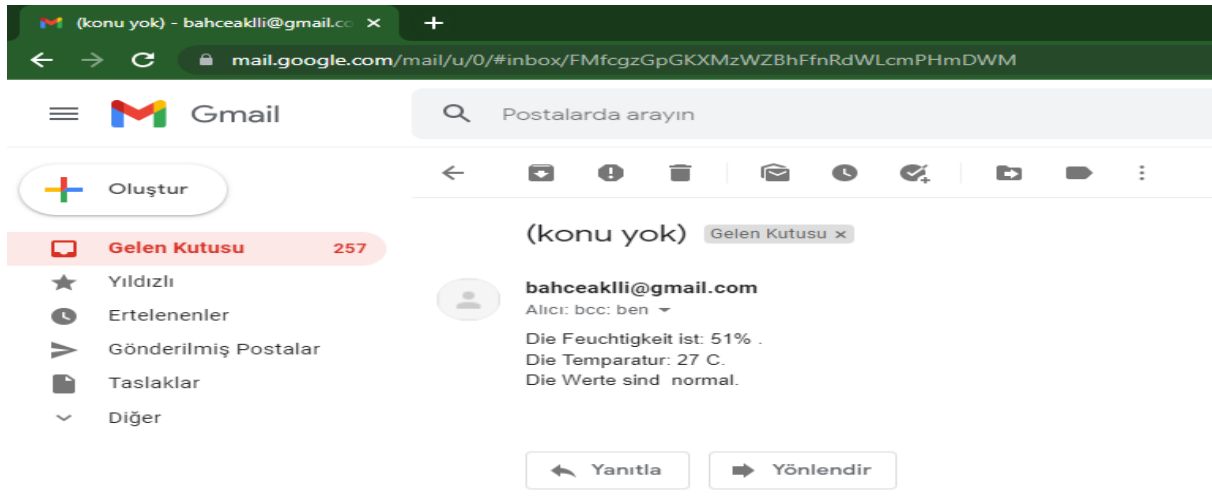
```
File Edit Tabs Help
pi@raspberrypi:~ $ gcc -Wall -o t2 temp2.c -lwiringPi
pi@raspberrypi:~ $ ./t2
Raspberry Pi wiringPi DHT11 Temperature test program
Data not good, skip
Data not good, skip
Humidity = 72.0 % Temperature = 28.3 C (82.4 F)
Data not good, skip
Humidity = 77.0 % Temperature = 29.0 C (84.2 F)
Humidity = 95.0 % Temperature = 29.3 C (84.2 F)
Data not good, skip
Humidity = 86.0 % Temperature = 29.8 C (84.2 F)
Humidity = 90.0 % Temperature = 29.8 C (84.2 F)
Humidity = 93.0 % Temperature = 30.6 C (86.0 F)
Data not good, skip
Humidity = 90.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Humidity = 86.0 % Temperature = 31.0 C (87.8 F)
Humidity = 78.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Data not good, skip
Humidity = 77.0 % Temperature = 30.8 C (86.0 F)
Humidity = 73.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Humidity = 72.0 % Temperature = 30.8 C (86.0 F)
Humidity = 74.0 % Temperature = 30.8 C (86.0 F)
Humidity = 74.0 % Temperature = 30.8 C (86.0 F)
Humidity = 72.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Humidity = 76.0 % Temperature = 30.9 C (86.0 F)
Humidity = 83.0 % Temperature = 30.9 C (86.0 F)
Data not good, skip
Humidity = 87.0 % Temperature = 31.1 C (87.8 F)
Data not good, skip
Humidity = 78.0 % Temperature = 30.8 C (86.0 F)
Humidity = 76.0 % Temperature = 30.7 C (86.0 F)
Humidity = 74.0 % Temperature = 30.8 C (86.0 F)
Humidity = 73.0 % Temperature = 30.6 C (86.0 F)
Humidity = 72.0 % Temperature = 30.6 C (86.0 F)
Humidity = 73.0 % Temperature = 30.5 C (86.0 F)
Data not good, skip
Humidity = 72.0 % Temperature = 30.5 C (86.0 F)
Humidity = 74.0 % Temperature = 30.5 C (86.0 F)
Humidity = 73.0 % Temperature = 30.4 C (86.0 F)
pi@raspberrypi:~ $
```

```

File Edit Tabs Help
pi@raspberrypi:~$ gcc -Wall -o t2 temp2.c -lwiringPi
pi@raspberrypi:~$ ./t2
Raspberry Pi wiringPi DHT11 Temperature test program
Data not good, skip
Data not good, skip
Humidity = 72.0 % Temperature = 28.3 C (82.4 F)
Data not good, skip
Humidity = 77.0 % Temperature = 29.0 C (84.2 F)
Humidity = 95.0 % Temperature = 29.3 C (84.2 F)
Data not good, skip
Humidity = 86.0 % Temperature = 29.8 C (84.2 F)
Humidity = 90.0 % Temperature = 29.8 C (84.2 F)
Humidity = 93.0 % Temperature = 30.6 C (86.0 F)
Data not good, skip
Humidity = 90.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Humidity = 86.0 % Temperature = 31.0 C (87.8 F)
Humidity = 78.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Data not good, skip
Humidity = 77.0 % Temperature = 30.8 C (86.0 F)
Humidity = 73.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Humidity = 72.0 % Temperature = 30.8 C (86.0 F)
Humidity = 74.0 % Temperature = 30.8 C (86.0 F)
Humidity = 74.0 % Temperature = 30.8 C (86.0 F)
Humidity = 72.0 % Temperature = 30.8 C (86.0 F)
Data not good, skip
Humidity = 76.0 % Temperature = 30.9 C (86.0 F)
Humidity = 83.0 % Temperature = 30.9 C (86.0 F)
Data not good, skip
Humidity = 87.0 % Temperature = 31.1 C (87.8 F)
Data not good, skip
Humidity = 78.0 % Temperature = 30.8 C (86.0 F)
Humidity = 76.0 % Temperature = 30.7 C (86.0 F)
Humidity = 74.0 % Temperature = 30.8 C (86.0 F)
Humidity = 73.0 % Temperature = 30.6 C (86.0 F)
Humidity = 72.0 % Temperature = 30.6 C (86.0 F)
Humidity = 73.0 % Temperature = 30.5 C (86.0 F)
Data not good, skip
Humidity = 72.0 % Temperature = 30.5 C (86.0 F)
Humidity = 74.0 % Temperature = 30.5 C (86.0 F)
Humidity = 73.0 % Temperature = 30.4 C (86.0 F)
pi@raspberrypi:~$ █

```


E-Mails, die von Raspberry Pi gesendet werden:



Quelle:

<https://forums.raspberrypi.com/viewtopic.php?t=168301>

<https://raspberrytips.com/mail-server-raspberry-pi/>

<https://github.com/nkundu/wiringpi-examples/blob/master/dht11.c>

<https://www.uugear.com/portfolio/read-dht1122-temperature-humidity-sensor-from-raspberry-pi/>

<https://www.delftstack.com/howto/c/how-to-convert-an-integer-to-a-string-in-c/>

<https://www.circuitbasics.com/>