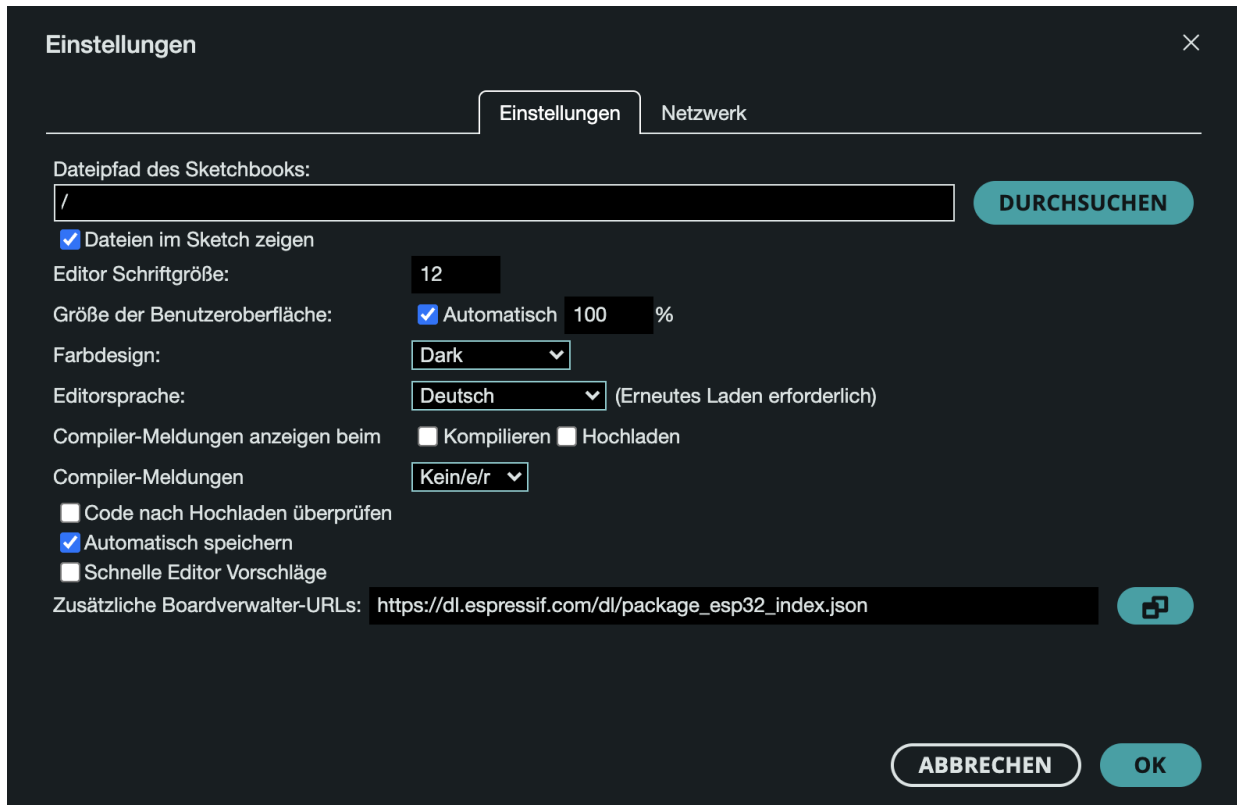


Leitfaden für den Bau eines low-cost pH-Sensors

1. Lade Arduino IDE auf deinen PC und installiere es.
2. Öffne **Arduino IDE** und die Einstellungen.
3. Kopiere den Pfad **https://dl.espressif.com/dl/package_esp32_index.json** zu **Boardverwalter-URLs**. Klicke anschließend auf OK.



4. Gehe über den Reiter **“Werkzeuge”** zu **“Board”** und wähle den **“Boardverwalter”** aus. Suche in dem neuen Fenster nach **“ESP32”** und installiere das angezeigte Board.



5. Gehe erneut über den Reiter **“Werkzeuge”** zu **“Bibliotheken verwalten”** und suche nach der Bibliothek **“Blynk”**, installiere diese anschließend.
6. Füge außerdem über den Reiter **“Sketch”**, **“Bibliothek einbinden”** und **“Bibliothek verwalten”** die .zip Datei ein, die dir auf GRIPS zur Verfügung gestellt wurde.
7. Baue den pH-Sensor nach dem dargestellten Schaltkreis zusammen.

8. Überprüfe den Schaltkreis bevor du den Mikrocontroller mit dem PC verbindest, um eventuelle Schäden zu vermeiden.
9. Wähle nun über **“Werkzeuge”** und **“Board”** unter **“ESP32”** das Board **“ESP32 Dev Module”** aus. Außerdem solltest du unter **“Werkzeuge”** den Uploadspeed auf **“921600”** stellen und den entsprechend **“Port”** also den USB-Anschluss an deinem PC auswählen.
10. Gehe auf die Website **Blynk** und erstelle ein Konto. Melde dich anschließend an und folge der Einführung.
11. Wähle nun dein Gerät aus und klicke neben dem Namen deines Gerätes auf die drei Punkte. Gehe zu **“Dashboard bearbeiten”**.
12. Wähle den Reiter **“Datenströme”** aus, lösche alle bestehenden und erstelle einen neuen.

Virtueller Pin-Datenstrom

NAME

PSEUDONYM

PIN

V4
▼

DATENTYP

Doppelt
▼

EINHEITEN

Keine
▼

MIN

0
▼

MAX

14
▼

DEZIMALSTELLEN

#.#
▼

STANDARDWERT

Standardwert
▼

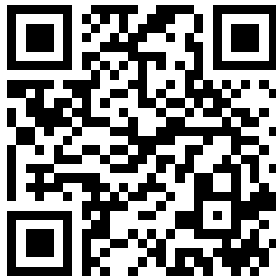
ERWEITERTE EINSTELLUNGEN

Abbrechen

Erstellen

13. Gehe zum Reiter **“Web-Dashboard”**.
14. Lösche zunächst alle vorgeschlagenen Widgets und wähle stattdessen das Diagramm-Widget aus. Dieses kannst Du anpassen, indem Du auf das Zahnrad des Widgets klickst. Wähle als Datenstrom **“V4”** aus und bestätige mit speichern.

15. Lade nun die Blynk App herunter.



(a) iOS



(b) Android

16. Öffne die Blynk App und melde dich an.

17. Verbinde deinen Mikrocontroller mit dem PC und öffne den Code für den pH-Sensor mit Arduino IDE, der dir auf **Github** zur Verfügung gestellt werden.

18. Füge in **Zeile 2 - 4** die Informationen deines Gerätes ein.

19. Ergänze außerdem die Zugangsdaten deines drahtlosen Netzwerks in **Zeile 18 / 19**.

20. Wenn du deinen pH-Wert nicht in einem Graphen darstellen lässt, kannst du den "delay"-Befehl in **Zeile 80** auskommentieren, indem du direkt davor `///
"` schreibst. Somit wird der Befehl nicht ausgeführt. Der "delay"-Befehl ist dafür da, um die Frequenz der Datenausgabe zu steuern, wenn du deine Werte aber nicht grafisch darstellen willst, kannst du die Frequenz über die Blynk-App steuern.

21. Wenn du fertig bist, kannst du den Code über den Haken oben links kompilieren (überprüfen) und anschließend auf den Mikrocontroller übertragen, in dem du den Pfeil rechts neben dem Haken anklickst.

22. Öffne anschließend den seriellen Monitor, welcher unter Werkzeuge zu finden ist.

23. Stellt im seriellen Monitor die Ausgabe auf **115200 Baud**, wie es auch im Code in **Zeile 50** steht. So stellt ihr sicher, dass verständliche Werte ausgegeben wurden.

24. Nimm nun den pH-Sensor, öffne die Schutzkappe und reinige die pH-Elektrode mit destilliertem Wasser.

25. Bevor der pH-Sensor verwendet werden kann, muss dieser kalibriert werden. Stellt ihn dazu in eine der Kalibrierlösungen (pH 4 / pH 7).

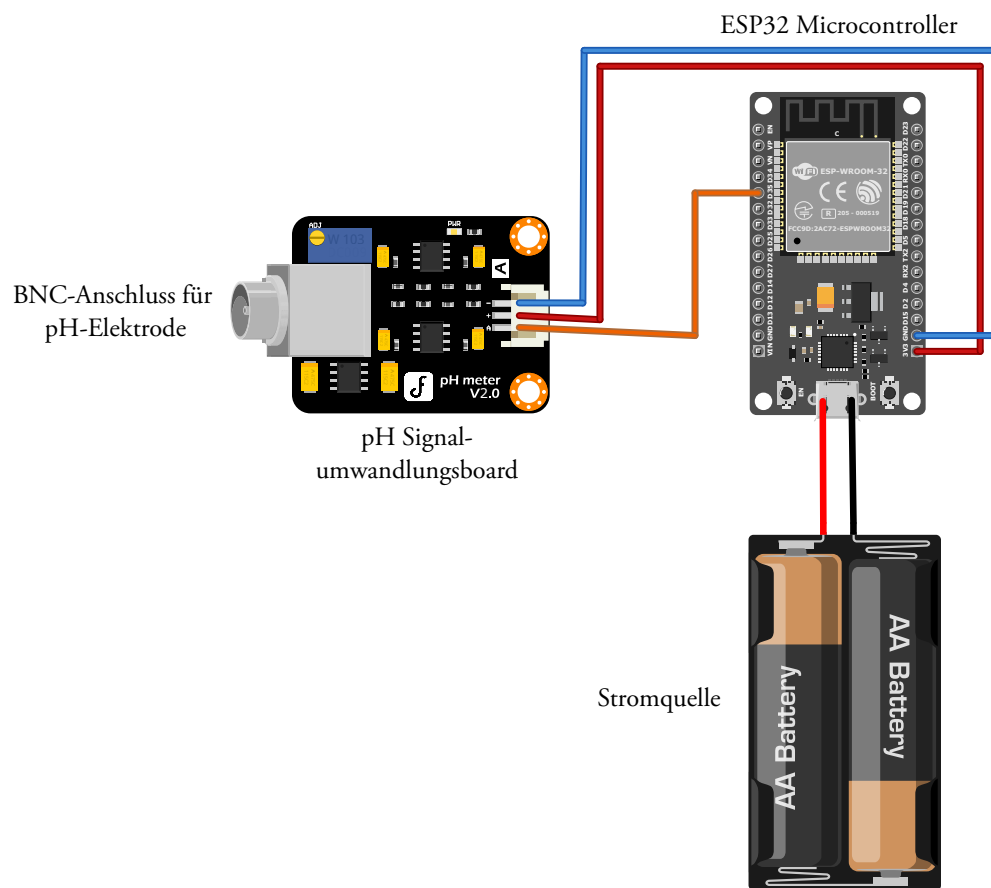
26. Beginnt den Kalibrierprozess in dem ihr in die Kommandozeile des seriellen Monitors den Befehl **"enterph"** eintippt und mit Enter bestätigt. Nun sollte der Kalibrierprozess aktiviert sein, kalibriert nun auf pH 7 oder pH 4 in dem ihr in die Kommandozeile den Befehl **"calph"** eingibt und mit Enter bestätigt. Ob es sich um die Lösung mit pH 4 oder pH 7 handelt, wird automatisch erkannt. Sobald im seriellen Monitor stabile Werte dargestellt werden, kann der erste Durchlauf mit dem Befehl **"exitph"** beendet werden.

27. Reinige zum Abschluss die Elektrode mit destilliertem Wasser.

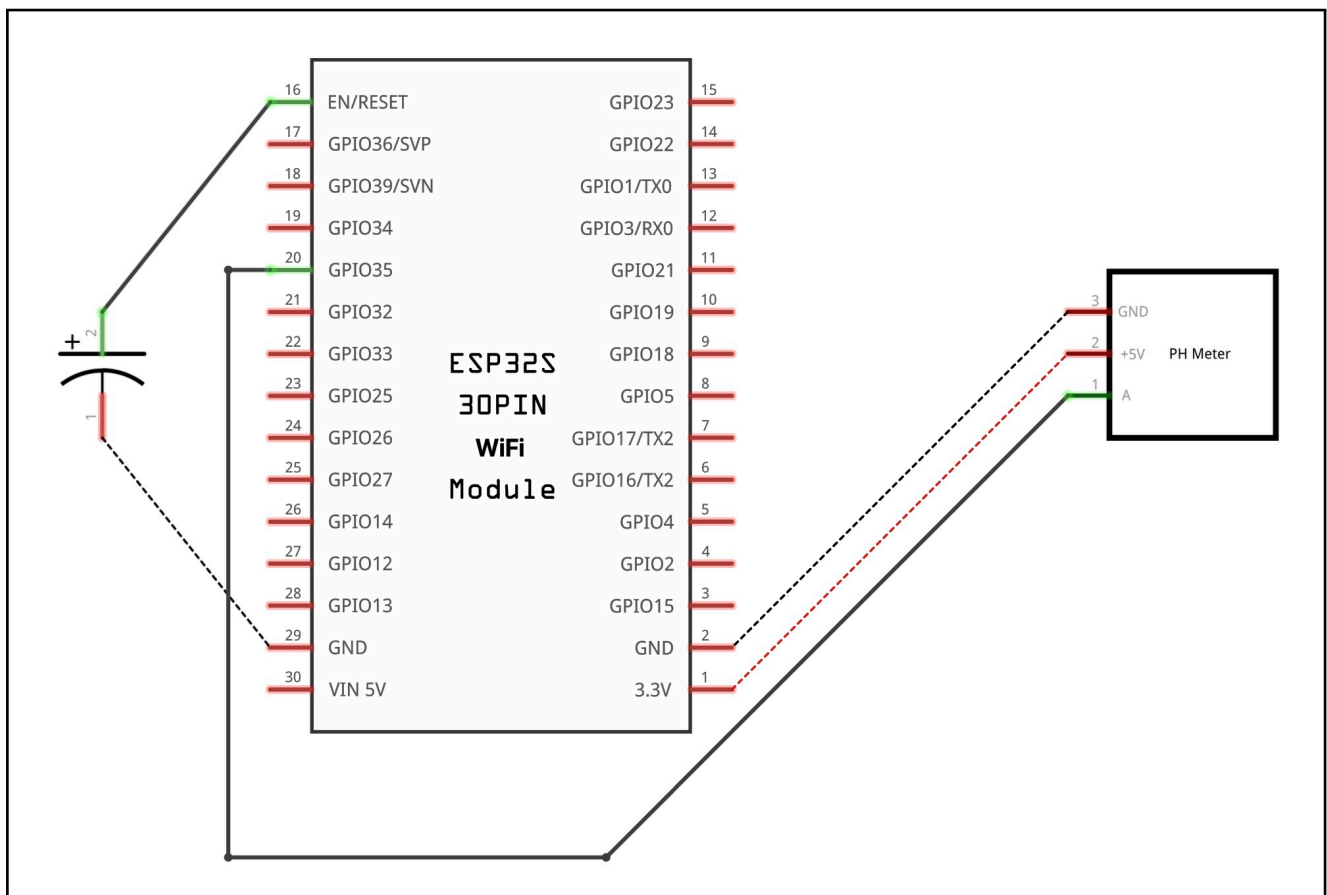
28. Wiederhole nun den Kalibrierprozess mit der anderen Kalibrierlösung.
29. Du kannst nun den Mikrocontroller von deinem PC trennen und die Stromversorgung mit der beigelegten Powerbank sicherstellen.
30. Nun solltest du die Werte des pH-Sensors auf deinem Smartphone oder auf der Website ablesen können.

```
1 /* Fill in information from Blynk Device Info here */
2 #define BLYNK_TEMPLATE_ID   ""
3 #define BLYNK_TEMPLATE_NAME ""
4 #define BLYNK_AUTH_TOKEN    ""
5
6
7 #include "DFRobot_ESP_PH.h"
8 #include <EEPROM.h>
9 DFRobot_ESP_PH ph;
10 #define BLYNK_PRINT Serial
11 #include <WiFi.h>
12 #include <WiFiClient.h>
13 #include <BlynkSimpleEsp32.h>
14
15
16 // Your WiFi credentials.
17 // Set password to "" for open networks.
18 char ssid[] = "";
19 char pass[] = "";
20
21 #define ESPADC 4096.0 //ESP Analog Digital Konvertierungswert
22 #define ESPVOLTAGE 3300 //ESP Spannungsversorgung
23 #define PH_PIN 35 //ESP Pin Nummer
24 float voltage, pHValue, temperature = 25;
25
26 BlynkTimer timer;
27
28 // This function is called every time the Virtual Pin 0 state
   changes
29 BLYNK_WRITE(V4)
30 {
31   // Set incoming value from pin V0 to a variable
32   int value = param.asInt();
33
34   // Update state
35   Blynk.virtualWrite(V1, value);
36 }
37
38 // This function is called every time the device is connected to
   the Blynk.Cloud
39 BLYNK_CONNECTED()
40 {
41   // Change Web Link Button message to "Congratulations!"
42   Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3
       .cdn.digitaloceanspaces.com/general/fte/congratulations.png")
       ;
43   Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3
```

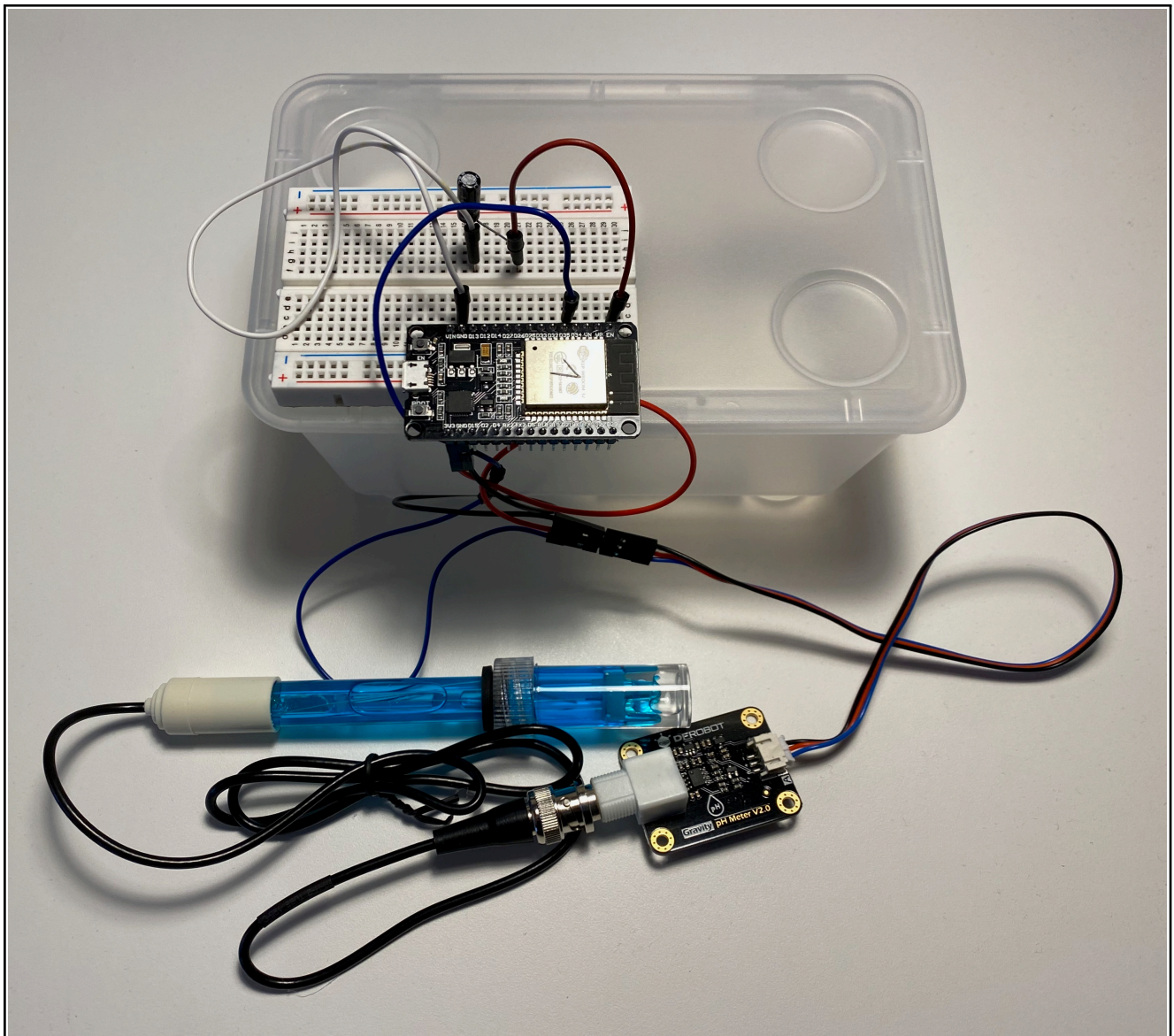
```
.cdn.digitaloceanspaces.com/general/fte/
congratulations_pressed.png");
44 Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-
started/what-do-i-need-to-blynk/how-quickstart-device-was-
made");
45 }
46
47 void setup()
48 {
49     //Debug console
50     Serial.begin(115200);
51     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
52     EEPROM.begin(32); //Speichererlaubnis der Kalibrierung
53     ph.begin();
54 }
55
56 void loop()
57 {
58     Blynk.run();
59     static unsigned long timepoint = millis();
60     if (millis() - timepoint > 1000U) //Zeitintervall: 1s
61     {
62         timepoint = millis();
63         voltage = analogRead(PH_PIN) / ESPADC * ESPVOLTAGE; //
        Ausgabe der Spannung
64         pHValue = ph.readPH(voltage, temperature); // Berechnung des
        pH-Werts anhand
65         //der Spannung mit Temperaturkompensation (insofern möglich)
66         Blynk.virtualWrite(V4, pHValue);
67
68         Serial.print("Spannung: ");
69         Serial.print(voltage, 2);
70         Serial.println(" mV");
71
72         // Serial.print("Temperatur:");
73         // Serial.print(temperature, 1);
74         // Serial.println(" °C");
75
76         Serial.print("pH: ");
77         Serial.println(pHValue, 1);
78         Serial.println();
79
80         delay(5000);
81     }
82     ph.calibration(voltage, temperature); // Kalibrierung via
        Seriellen Monitor
83     }
```



Verbindungsplan



Schaltplan



Exemplarischer Aufbau