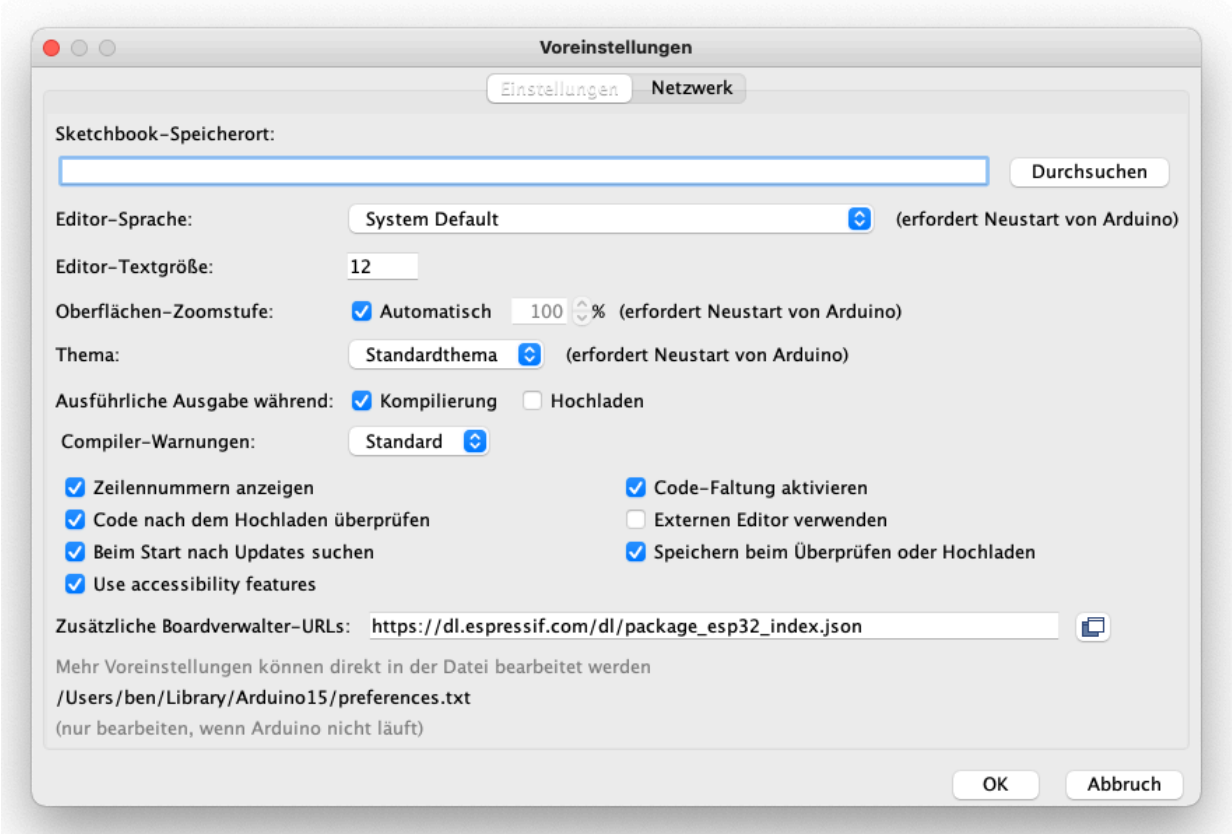
Leitfaden für den Bau eines low-cost pH-Sensors

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 **“Boards”** 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. Lade nun die Blynk App herunter.

(a) iOS (b) Android

11. Öffne die Blynk App und melde dich an. Erstelle über “**+**” ein neues Projekt. Gib dem Projekt einen Namen, wähle als Gerät **“ESP32 Dev Board”** und als Verbindungsart **“BLE”**. Klicke abschließend auf **“Create Project”**.

12. Wische in deiner Projektoberfläche nach links und wähle abhängig davon, wie deine pH-Werte dargestellt werden sollen das entsprechende Widget aus.

13. Füge unbedingt das BLE-Widget hinzu, um dein Smartphone mit dem Mikrocontroller verbinden zu können.

14. Tippe auf das von dir ausgewählte Widget zur Darstellung deiner pH-Werte und wähle als Inputquelle den **Virtuellen PIN 1** aus. Alle weiteren Darstellungsoptionen kannst du nach deinen Vorlieben auswählen.

15. Verbinde deinen Mikrocontroller mit dem PC und öffne den Code für den pH-Sensor mit Arduino IDE, der dir auf GRIPS zur Verfügung gestellt wurden.

16. Füge bei “char auth []”, in **Zeile 10**, deinen **Auth Token** ein, den du per E-Mail erhalten hast.

17. Wenn du deinen pH-Wert nicht in einem Graphen darstellen lässt, kannst du den “delay”-Befehl in **Zeile 50** 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.

18. 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.

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

20. Stellt im seriellen Monitor die Ausgabe auf **9600 Baud**, wie es auch im Code in **Zeile 19** steht. So stellt ihr sicher, dass verständliche Werte ausgegeben wurden.

21. Nehmt nun den pH-Sensor, öffnet die Schutzkappe und reinigt die pH-Elektrode mit destilliertem Wasser.

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

23. Beginnt den Kalibrierprozess in dem ihr in die Kommandoleiste 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”** eingebt 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 darstellt werden, kann der erste Durchlauf mit dem Befehl **“exitph”** beendet werden.

24. Reinige zum Abschluss die Elektrode mit destilliertem Wasser.

25. Wiederhole nun den Kalibrierprozess mit der anderen Kalibrierlösung.

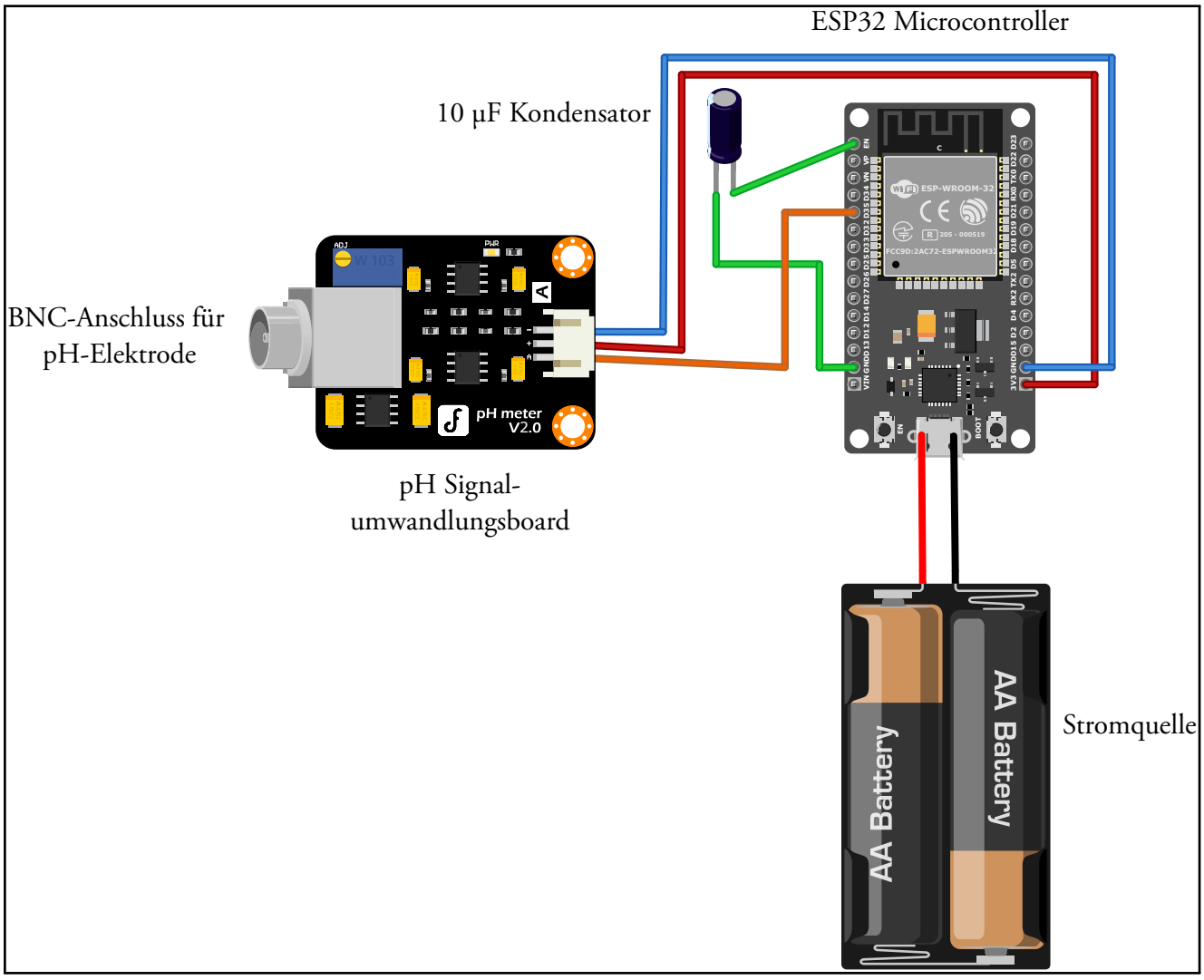
26. Du kannst nun den Mikrocontroller von deinem PC trennen und die Stromversorgung mit der beigelegten Powerbank sicherstellen.

27. Tippe in der Blynk App auf deiner Projektoberfläche auf das Bluetooth Widget und auf den Button **“Connect BLE Device”** wähle dort das Gerät **“Blynk”** aus. Alternativ kannst du dir einen eigenen Namen überlegen und diesen im Code in der **Zeile 21** festlegen. Anschließend muss der Code neu kompiliert und hochgeladen werden.

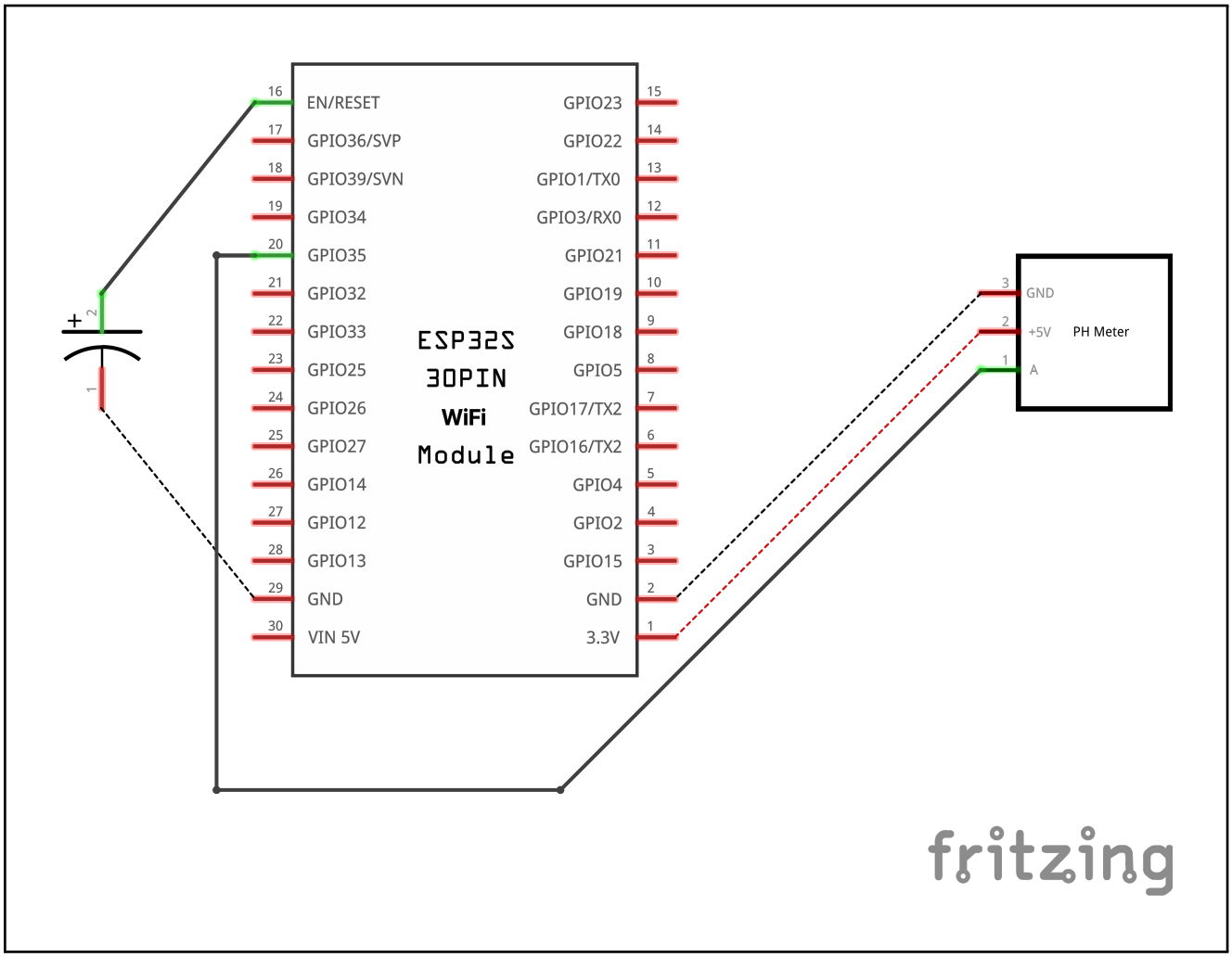
28. Nun solltest du die Werte des pH-Sensors auf deinem Smartphone ablesen können.

```
#include "DFRobot_ESP_PH.h"
#include <EEPROM.h>
DFRobot_ESP_PH ph;
#define BLYNK_USE_DIRECT_CONNECT
#include <BlynkSimpleEsp32_BLE.h>
#include <BLEDevice.h>
#include <BLEServer.h>
//#define BLYNK_PRINT Serial
char auth[] = "SmOrWBah1Me7DfokzXAj8ehBQNxoP3HE";
#define ESPADC 4096.0   //ESP Analog Digital Konvertierungswert
#define ESPVOLTAGE 3300 //ESP Spannungsversorgung
#define PH_PIN 35    //ESP Pin Nummer
float voltage, phValue, temperature = 25;

void setup()
{
  //Debug console
  Serial.begin(115200);
  Serial.println("Verbindungsaufbau...");
  Blynk.setDeviceName("Blynk");
  Blynk.begin(auth);
  EEPROM.begin(32);//Speichererlaubnis der Kalibrierung
  ph.begin();
}
void loop()
{
  Blynk.run();
   static unsigned long timepoint = millis();
  if (millis() - timepoint > 1000U) //Zeitintervall: 1s
  {
    timepoint = millis();
    voltage = analogRead(PH_PIN) / ESPADC * ESPVOLTAGE; //
       Ausgabe der Spannung
    phValue = ph.readPH(voltage, temperature); // Berechnung des
       pH-Werts anhand
    //der Spannung mit Temperaturkompensation (insofern möglich)
    Blynk.virtualWrite(V1, phValue);
    Serial.print("Spannung:");
    Serial.print(voltage, 2);
    Serial.println("mV");
    Serial.print("Temperatur:");
    Serial.print(temperature, 1);
    Serial.println("°C");
    Serial.print("pH:");
    Serial.println(phValue, 1);
    Serial.println();

```

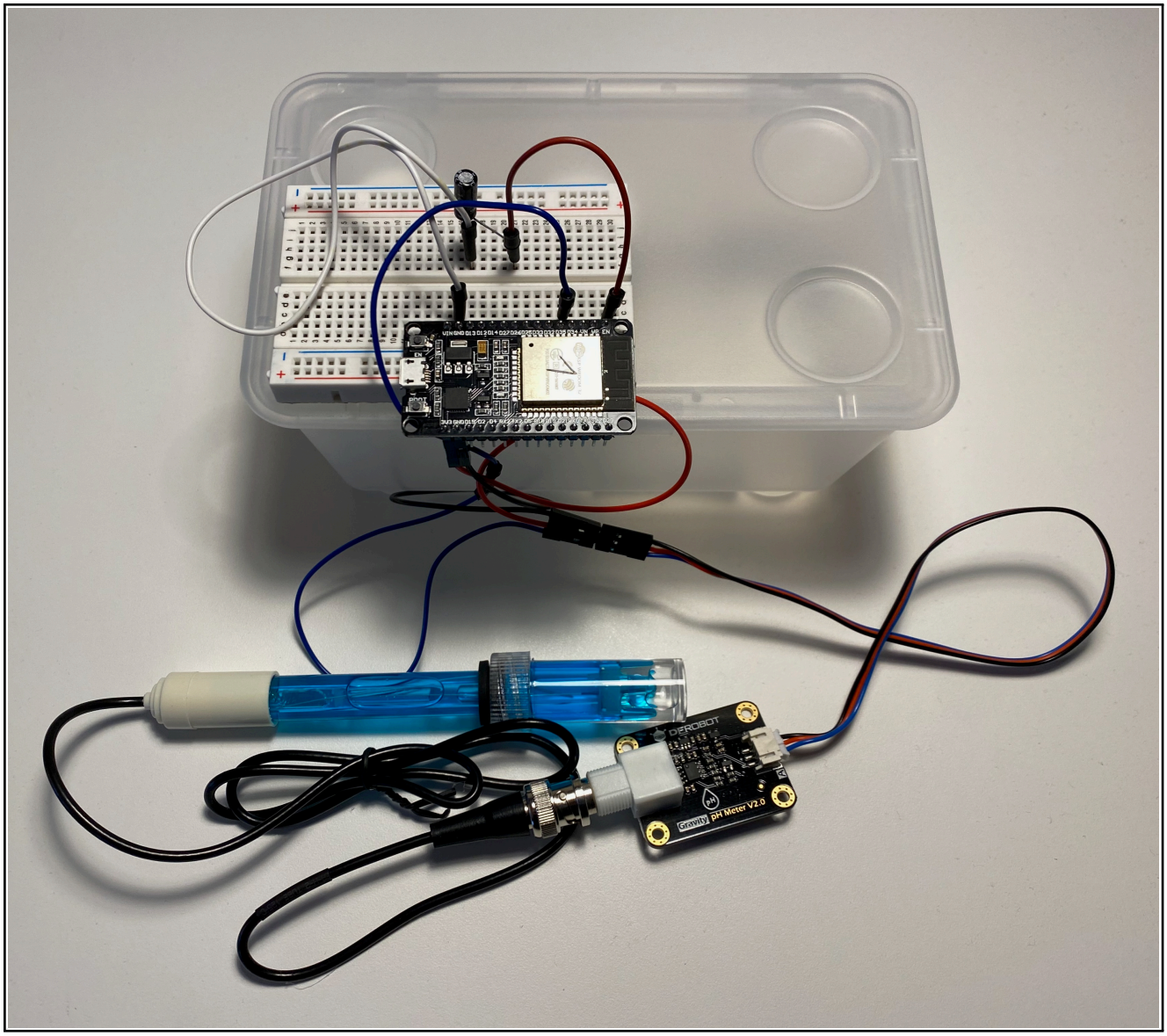
```
    delay(1000);
}
 ph.calibration(voltage, temperature); // Kalibrierung via
    Seriellen Monitor
        }
```



Verbindungsplan


ESP32S
30PIN
WiFi
Module
EN/RESET
GPIO36/SVP
GPIO39/SVN
GPIO34
GPIO35
GPIO32
GPIO33
GPIO25
GPIO26
GPIO27
GPIO14
GPIO12
GPIO13
GND
VIN 5V
GPIO23
GPIO22
GPIO1/TX0
GPIO3/RX0
GPIO21
GPIO19
GPIO18
GPIO5
GPIO17/TX2
GPIO16/TX2
GPIO4
GPIO2
GPIO15
GND
3.3V
GND
+5V
A
PH Meter
fritzing

Schaltplan



Exemplarischer Aufbau