

Das GPS-Modul NEO-6M am Arduino anschließen und verwenden



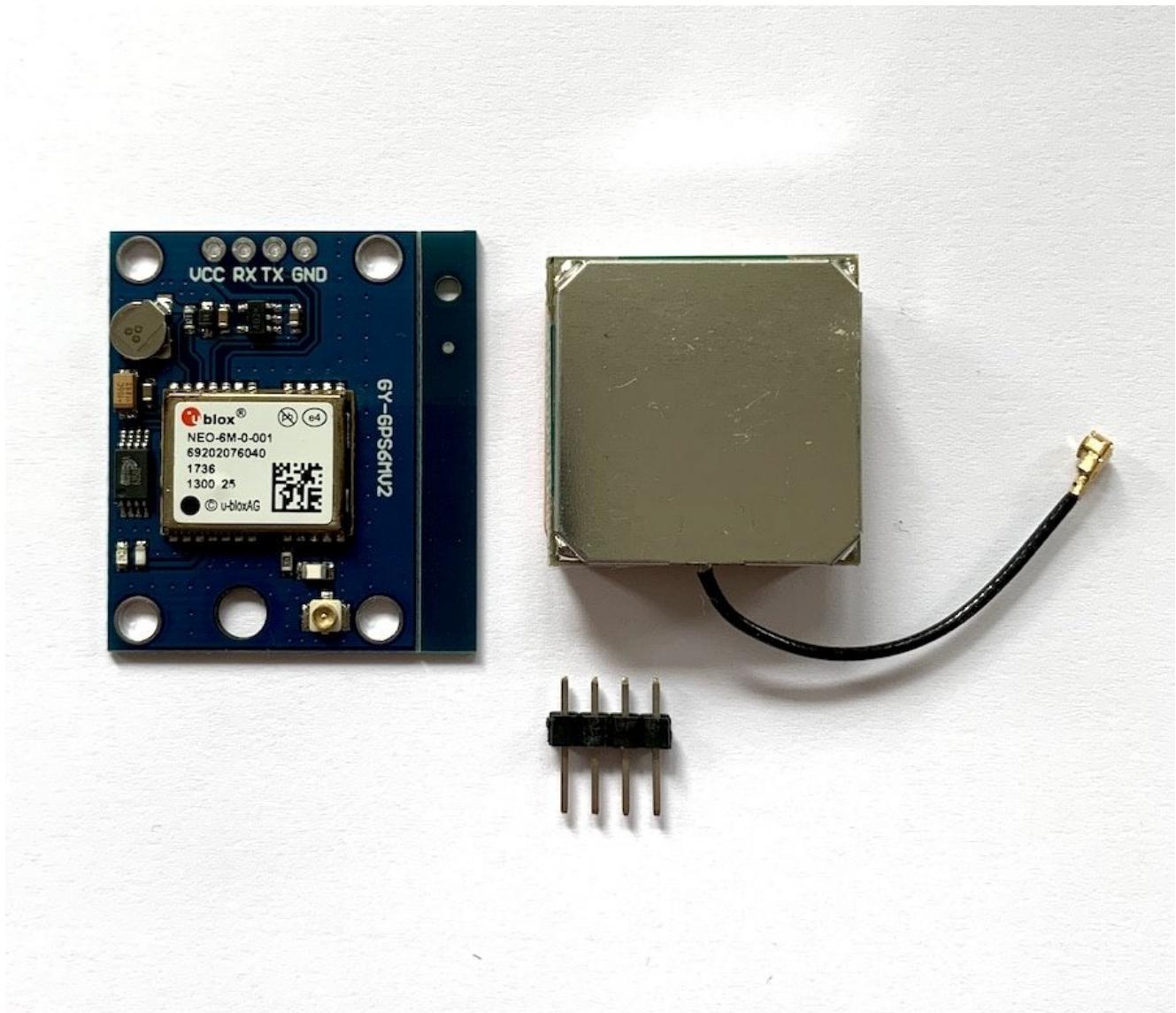
Um die Koordinaten deines Standorts herauszufinden, benötigst du nur einen Arduino, das [GPS Modul NEO-6M*](#) und ein paar Zeilen Code. Erfahre in diesem Tutorial, wie du dieses Modul richtig anschließt und die GPS-Daten ausliest.

Überblick und Anschluss des GPS-Moduls

Das NEO-6M verwendet GPS (Global Positioning System), um seinen aktuellen Standort zu ermitteln. GPS ist seit dem Jahr 2000 frei verfügbar und mittlerweile allgegenwärtig. Für die Bestimmung der Position dienen spezielle Satelliten, mit denen sich dein GPS-Modul verbindet. Übrigens lässt sich so nicht nur der Standort ermitteln, sondern auch die Geschwindigkeit. [Auf Wikipedia erfährst du mehr über die Funktion von GPS.](#)

Das Bauteil besteht aus zwei Teilen – aus dem Modul selbst und

einer externen Antenne. Außerdem wird es üblicherweise ohne verlötzte Header-Pins geliefert. Du benötigst also einen Lötkolben, um diese selbst anzubringen.



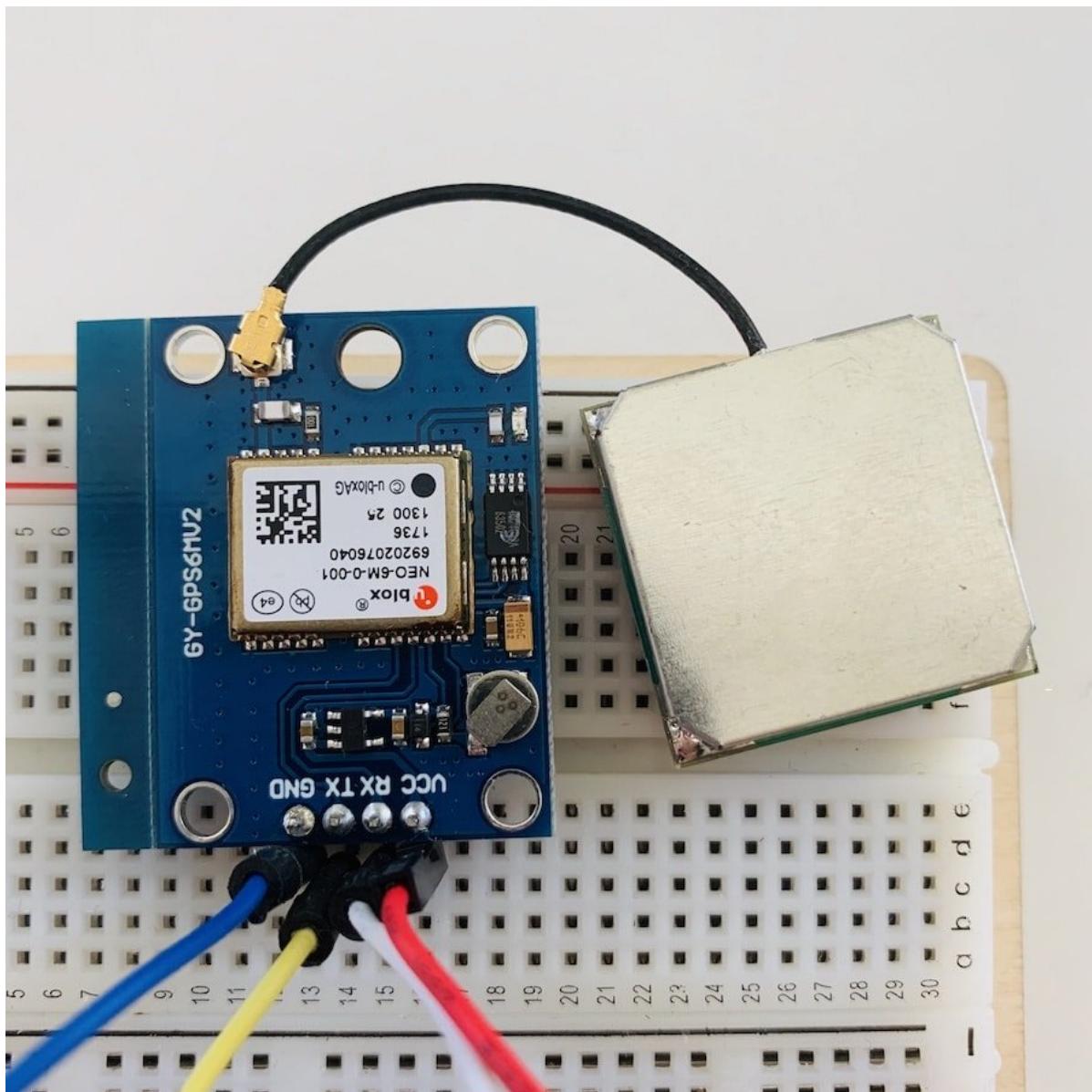
Am GPS-Modul befinden sich vier Anschlüsse, die du wie folgt mit deinem Arduino verbindest:

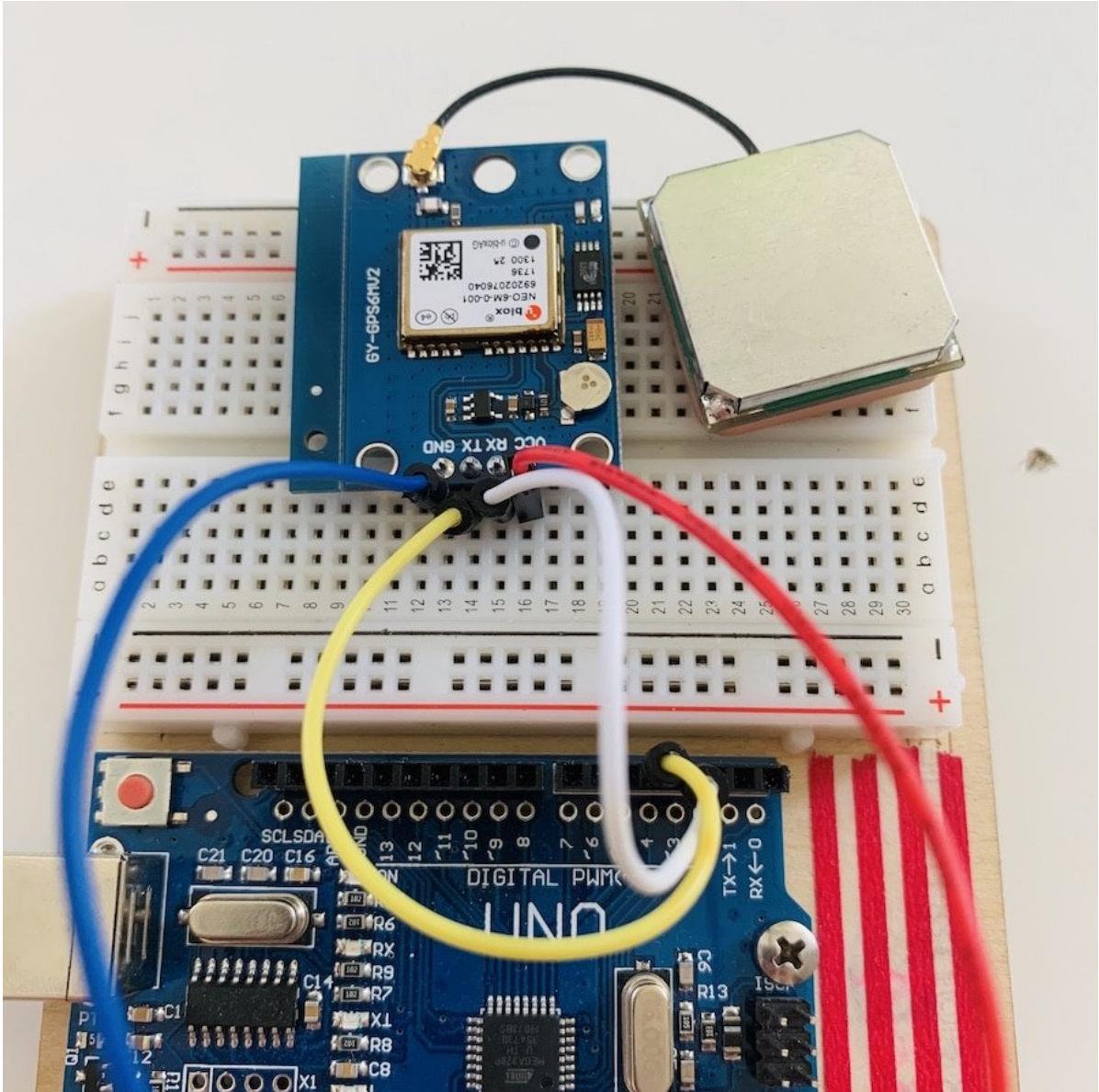
| | |
|-----|------------------------|
| VCC | 5V |
| RX | Digitalpin deiner Wahl |
| TX | Digitalpin deiner Wahl |
| GND | GND |

Stecke zunächst die externe Antenne, wie auf den folgenden Bildern zu sehen, in ihren Anschluss auf der Platine. Danach verbindest du die Pins **VCC** und **GND** mit einem **5V-Pin** und **GND** an

deinem Arduino.

Die Pins RX und TX sind für die Kommunikation mit deinem Arduino zuständig. Mit welchen Digitalpins du sie verbindest, steht dir frei. **Damit den den folgenden Sketch gleich verwenden kannst, verbinde RX mit Pin 3 und TX Pin 4.**





Hinweis: Je nachdem wie gut oder schlecht (z.B. in geschlossenen Räumen) der Empfang ist, kann es etwas dauern, bis die Verbindung deines Moduls mit den GPS-Satelliten ist. **Sobald die Verbindung steht blinkt eine LED auf dem Modul.** Sollte sie auch nach mehreren Minuten nicht anfangen zu blinken, wechsle am besten deinen Standort.

GPS-Daten empfangen und im Seriellen Monitor ausgeben

Im Prinzip benötigst du keine besondere Bibliothek, um die

Daten des GPS-Moduls zu erhalten. Da es sich hierbei jedoch um recht schwer lesbare Rohdaten handelt, empfiehlt sich die die Bibliothek **TinyGPSPlus**, die dein Vorhaben erheblich vereinfacht.

Diese Bibliohek musst du allerdings selbst herunterladen und bereitstellen (Stand Februar 2020). Gehe hierfür wie folgt vor:

1. [Lade dir hier TinyGPSPlus von Github herunter](#), indem du auf den grünen Button **Clone or download** und dann auf **Download ZIP** klickst.
2. Entpacke die ZIP-Datei
3. Benenne den entpackten Ordner in **TinyGPSPlus** um
4. Verschiebe den Ordner in den Unterordner **libraries** deiner Arduino-Installation

Jetzt kannst du die Bibliothek in deinen Sketch einbinden:

```
#include <TinyGPS++.h>
```

Du benötigst noch eine weitere Bibliothek: **SoftwareSerial.h** – diese steuert die Kommunikation zwischen deinem Arduino und dem GPS-Modul:

```
#include <SoftwareSerial.h>
```

Anschließend erstellst du die Instanz **gps** von **TinyGPSPlus** und hinterlegst die Pins, an denen du das Modul angeschlossen hast. **In unserem Fall 4 (TX) und 3 (RX)**.

```
TinyGPSPlus gps;  
SoftwareSerial ss(4, 3);
```

In der Funktion **void setup()** legst du die Baud-Raten für den Seriellen Monitor und die Kommunikation mit dem GPS-Modul fest:

```
void setup() {
```

```
Serial.begin(115200);
ss.begin(9600);
}
```

Nun folgt der Teil, in dem die Action passiert, der Loop. Hier wird zunächst geprüft, ob das Modul angeschlossen ist und es Daten von Satelliten empfängt. Ist das der Fall, besorgst du dir diese Daten lesbar als Koordinaten über die Funktionen **gps.location.lat()** und **gps.location.lng()** und gibst sie im Seriellen Monitor aus.

```
void loop() {
    while (ss.available() > 0) {
        gps.encode(ss.read());

        if (gps.location.isUpdated()) {

            // Breitengrad mit 3 Nachkommastellen
            Serial.print("Breitengrad= ");
            Serial.print(gps.location.lat(), 3);

            // Längengrad mit 3 Nachkommastellen
            Serial.print(" Längengrad= ");
            Serial.println(gps.location.lng(), 3);
        }
    }
}
```

Und das war es auch schon. In deinem Seriellen Monitor erscheinen nun die GPS-Koordinaten deines Standorts. ☐

Hier der gesamte Sketch:

Wie geht es weiter?

Damit die Koordinaten nicht nur im Seriellen Monitor

erscheinen, schließe ein Display an deinen Arduino an. So schließt du ein [TFT-Display](#) oder ein [LCD-Display](#) an.

**Amazon Affiliate Link: Bei einer Bestellung ändert sich für dich nichts – wir erhalten jedoch eine kleine Provision.*